Contents

Preface ........................................................................................................................................................................ vi

The Oracle Health Sciences Safety Suite .................................................................................................................. vi

Audience ........................................................................................................................................ vii

Documentation Accessibility ........................................................................................................................ viii

Related Documents ....................................................................................................................................... viii

Finding Information and Patches ...................................................................................................................... viii

Finding Oracle Documentation ......................................................................................................................... x

Conventions ......................................................................................................................................................... xi

1 Introduction

2 Overview

Out-of-the-box Periodic Reports ........................................................................................................................ 2-1

Supported Output Formats ................................................................................................................................. 2-1

Security Limitations ........................................................................................................................................ 2-1

Authorization ...................................................................................................................................................... 2-1

Authentication .................................................................................................................................................... 2-2

Scheduler Support .......................................................................................................................................... 2-2

Template Design ............................................................................................................................................... 2-2

Database Code Wrapping ................................................................................................................................ 2-2

3 Generic Architecture

Generic Architecture ........................................................................................................................................... 3-1

4 Database Architecture

Database Architecture .......................................................................................................................................... 4-1

BI Publisher Periodic Reporting Owner Schema ........................................................................................... 4-1

Tables .............................................................................................................................................................. 4-2

Views ............................................................................................................................................................... 4-3

Database Link .................................................................................................................................................. 4-4

Java Objects and Function Call ....................................................................................................................... 4-4

Packages .......................................................................................................................................................... 4-4

BI Publisher Owner Schema Database Jobs ..................................................................................................... 4-7

Argus Safety Database Updates ...................................................................................................................... 4-7
5 Design Structure

BI Publisher Periodic Reports Design Structure ................................................. 5-1
    BI Publisher Data Model ..................................................................................... 5-1
        Data Sets ........................................................................................................... 5-1
        Event Triggers .................................................................................................. 5-2
        Parameters ...................................................................................................... 5-2
    BI Publisher Report Layout .................................................................................. 5-4

Report Data Flow ..................................................................................................... 5-4
    Argus Safety UI Entry ........................................................................................... 5-4
    BI Publisher Data Flow ......................................................................................... 5-5

6 Extending a BIP Report

Database Layer ........................................................................................................ 6-1
    Configuration Extensibility .................................................................................. 6-2
    Extending with User Exits .................................................................................. 6-3
    Extending Global Temporary Tables .................................................................... 6-4
    Extending using Custom Objects .......................................................................... 6-4
        Adding New Columns to GTT or RM Tables ...................................................... 6-4
        Filling Custom Tables ....................................................................................... 6-4
        Filling Custom Views ....................................................................................... 6-5
        Filling Custom Packages ................................................................................ 6-5
    BIP Custom Schema for Cloud ........................................................................... 6-5
    Adding or Modifying a View ................................................................................ 6-5
    Adding a Column to the Existing Table ................................................................ 6-5

Extending the BI Publisher Data Model ..................................................................... 6-5
    Data Model Query Naming Convention ............................................................... 6-6
    Data Model Nested Queries ................................................................................. 6-6
    Data Structure Groups ........................................................................................ 6-7
    DSUR Summary Table Naming Conventions ....................................................... 6-8
    Case Series Tables ............................................................................................... 6-10
        Common Queries (lexicals) Used in Reports ................................................. 6-10
    Lexical Parameters .............................................................................................. 6-11

Creating a Custom Report ......................................................................................... 6-12

A Parameters

B Blinding

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

C Flexible Data Re-Categorization

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

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

BI Publisher Periodic report does not Execute from Argus UI

Database Connection Errors

BI Publisher Scheduled Report Output does not Show up

ORA-01427: single-row sub-query returns more than one row
ORA-06512: at "<bip_owner>.PKG_AGG_RPT" for p_updclinicaldrugrole procedure
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.

This preface includes the following topics:

- Audience
- Documentation Accessibility
- Related Documents
- Finding Information and Patches
- Finding Oracle Documentation
- Conventions

The Oracle Health Sciences Safety Suite

This product is part of the Oracle Health Sciences Safety Suite, an integrated solution for end-to-end vigilance from adverse event management to signal management, through the entire lifecycle of a medicinal product from clinical trials to post-marketing surveillance.

The Oracle Health Sciences Safety Suite consists of the following components:
Oracle Argus Standard Edition: Manage and report adverse events through a workflow including case intake, data entry, coding, quality review, medical review, expedited reporting, and periodic reporting. Modules include Oracle Argus Safety, Oracle Argus Interchange, Oracle Argus Affiliate, Oracle Argus Dossier, Oracle Argus Unblinding, and the Oracle Health Sciences Adverse Event Integration Pack for Oracle Health Sciences InForm and Oracle Argus.

Oracle Argus Enterprise Edition: In addition to managing the adverse event workflow and reporting, employ a powerful and flexible business analytics and intelligence platform for both scientific analysis and operational metrics. Modules include Oracle Argus Analytics, Oracle Argus Insight, Oracle Argus Mart, Oracle Argus Safety, Oracle Argus Interchange, Oracle Argus Affiliate, Oracle Argus Dossier, Oracle Argus Unblinding, and the Oracle Health Sciences Adverse Event Integration Pack for Oracle Health Sciences InForm and Oracle Argus.

Oracle Argus Safety Japan: Manage and report adverse events in Japan, and connect the global and local workflows using a single database.

Oracle Health Sciences Empirica Topics: Manage and document safety signals through a workflow including validation, prioritization, assessment, confirmation/refutation, and resulting actions.

Oracle Health Sciences Empirica Study: Detect and analyze safety signals in clinical trial data including adverse events, clinically significant labs, electrocardiograms, vital signs, and shifts from baseline.

Oracle Health Sciences Empirica Signal: Detect and analyze safety signals in post-marketing spontaneous adverse reaction data including public health authority databases and/or private inhouse databases such as Oracle Argus.

Oracle Health Sciences Empirica Healthcare Analysis: Evaluate safety signals in healthcare data including electronic medical records and administrative claims, and support pharmacoepidemiology, comparative effectiveness analysis, and health economics and outcomes research.

For more information on Argus Safety, visit the Oracle Health Sciences Safety suite page at:
http://www.oracle.com/goto/pharmacovigilance.html

**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

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

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

Related Documents
For more information, see the following documents in the Oracle Argus Safety documentation set:
- Argus Safety Affiliate User Guide
- Argus Safety Administrator’s User Guide
- Argus Safety Dossier User Guide
- Argus Safety Interchange User Guide
- Argus Safety Installation Guide
- Argus Safety Service Administrator Guide
- Argus Safety Flexible Aggregate Reporting Extensibility Guide
- Argus Safety BIP Aggregate Reporting User’s Guide
- Argus Safety User’s Guide
- Argus Safety Unblinding User Guide
- Argus Safety Minimum Security Configuration Guide
- Argus Safety Japanese Administrator’s Guide
- Argus Interchange Japanese User’s Guide
- Argus Safety Japanese User’s Guide

Finding Information and Patches
Your source for the latest information about Argus Safety is Oracle Support’s self-service website My Oracle Support.

For more information on Argus Safety, visit the Oracle Health Sciences Safety suite page at:
http://www.oracle.com/goto/pharmacovigilance.html
Before you install and use Argus Safety, always visit the My Oracle Support website for the latest information, including alerts, White Papers, and bulletins.

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

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

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

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

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

To search by article ID:
2. Locate the Search box in the upper right corner of the My Oracle Support page.
3. Click the sources icon to the left of the search box, and then select Article ID from the list.
4. Enter the article ID number in the text box.
5. Click the magnifying glass icon to the right of the search box (or press the Enter key) to execute your search.

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

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

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

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

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

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

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

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

---

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

---

Finding Other Oracle Documentation
To get user documentation for other Oracle products:
1. Go to the following web page:

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

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

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

The following text conventions are used in this document:

<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><code>monospace</code></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Argus Safety Flexible Aggregate Reporting leverages the capabilities of the Oracle BI Publisher reporting solution for Argus Safety Periodic Reports. This solution helps customers overcome the challenges of periodic reporting to meet a variety of business needs.

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

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

<table>
<thead>
<tr>
<th>Table 1–1 Components of the Extensibility Guide</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter Name</td>
</tr>
<tr>
<td>Introduction</td>
</tr>
<tr>
<td>Overview</td>
</tr>
<tr>
<td>Generic Architecture</td>
</tr>
<tr>
<td>Database Architecture</td>
</tr>
<tr>
<td>Design Structure</td>
</tr>
<tr>
<td>Extending a BIP Report</td>
</tr>
</tbody>
</table>
Overview

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

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

2.1 Out-of-the-box Periodic Reports

The out-of-the-box Argus Safety Flexible Aggregate Reporting (Oracle Argus FAR) feature provides the following periodic reports:

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

2.2 Supported Output Formats

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

- PDF
- RTF

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

2.3 Security Limitations

2.3.1 Authorization

Oracle Argus FAR has been certified for the following security models:
2.3.2 Authentication
The Oracle Argus FAR feature is certified for Oracle Access Manager (OAM) and Single Sign-on (SSO).

2.4 Scheduler Support
Oracle Argus FAR supports Quartz scheduler.

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

2.6 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.
This chapter discusses BI Publisher Periodic Reporting generic architecture.

### 3.1 Generic Architecture

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

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

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

*Figure 3–1 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

<table>
<thead>
<tr>
<th>Component</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Argus Web Console</td>
<td>The first piece of the application for configuring Argus Safety. Handles the following:</td>
</tr>
<tr>
<td></td>
<td>- Enabling the BIP module</td>
</tr>
<tr>
<td></td>
<td>- Supplying the necessary BIP security credentials and setting up the persist duration</td>
</tr>
<tr>
<td></td>
<td>- Setting up the report template path</td>
</tr>
<tr>
<td>Argus Web</td>
<td>Provides the report configuration for ICH PSUR and CTPR reports. Supplies the criteria that generates the case series.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>AG Service</td>
<td>Internally invokes the BIP Periodic report through WebServices and runs the report in the background.</td>
</tr>
<tr>
<td>Argus Database - BIP</td>
<td>New schema created during the schema creation that holds all objects used for generating the periodic reports.</td>
</tr>
<tr>
<td>Schema</td>
<td>The objects in the schema are responsible for generating the data needed.</td>
</tr>
<tr>
<td>BIP Server</td>
<td>Server where BI Publisher reports are located and executed. Report output is temporarily stored in the BIP server.</td>
</tr>
</tbody>
</table>

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.
This chapter contains the following sections:

- Section 4.1, "Database Architecture"
- Section 4.2, "Argus Safety Database Updates"
- Section 4.3, "Access Grants to BI Publisher Owner"

4.1 Database Architecture

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

4.1.1 BI Publisher Periodic Reporting Owner Schema

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

---

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

---

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

For report generation, this schema holds:

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

Tables in the BIP schema populate the case data:

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

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

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Type</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPT_AGG_PARAMS</td>
<td>SUPPORT</td>
<td>Stores the list of parameters that are passed down to run the report.</td>
</tr>
<tr>
<td>RPT_AGG_CASE_SERIES</td>
<td>SUPPORT</td>
<td>Stores case series information.</td>
</tr>
<tr>
<td>RPT_AGG_CS_CASES</td>
<td>SUPPORT</td>
<td>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.</td>
</tr>
<tr>
<td>RPT_AGG_BIP_JOB</td>
<td>SUPPORT</td>
<td>Stores BIP job information and other transactional data. This table data is retained without purging.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_CASE</td>
<td>Global Temp</td>
<td>Stores case information.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_DRUG</td>
<td>Global Temp</td>
<td>Stores drug related information.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_EVENT</td>
<td>Global Temp</td>
<td>Stores event related information.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_EV2DRUG</td>
<td>Global Temp</td>
<td>Stores event to drugs related assessment details.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_HEALTHAUTHID</td>
<td>Global Temp</td>
<td>Stores health authority details.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_UNIQ_CASES</td>
<td>Global Temp</td>
<td>Maintains a list of unique cases.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_DET_LIST</td>
<td>Global Temp</td>
<td>Temporary support table that fetches assessment data.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_DRUGNAMES</td>
<td>Global Temp</td>
<td>Stores the drug names for reporting.</td>
</tr>
<tr>
<td>GTT_RPT_AGG_BIP_BLOB</td>
<td>CONFIG</td>
<td>Copies and holds the report output blob between the BIP Owner schema and the BIP Metadata repository database.</td>
</tr>
<tr>
<td>RPT_AGG_JOB_EXEC_STS</td>
<td>CONFIG</td>
<td>Used to avoid multiple report jobs fetching the report output at the same time.</td>
</tr>
<tr>
<td>RM_RPT_AGG_CASE</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_CASE.</td>
</tr>
<tr>
<td>RM_RPT_AGG_DET_LIST</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_DET_LIST.</td>
</tr>
<tr>
<td>RM_RPT_AGG_DRUG</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_DRUG.</td>
</tr>
<tr>
<td>RM_RPT_AGG_DRUGNAMES</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_DRUGNAMES.</td>
</tr>
<tr>
<td>RM_RPT_AGG_EV2DRUG</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_EV2DRUG.</td>
</tr>
<tr>
<td>RM_RPT_AGG_EVENT</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_EVENT.</td>
</tr>
<tr>
<td>RM_RPT_AGG_HEALTHAUTHID</td>
<td>PERSIST</td>
<td>Persist table for GTT_RPT_AGG_HEALTHAUTHID.</td>
</tr>
</tbody>
</table>
For details about tables, columns, and column mappings with Argus Safety, refer to the embedded Argus Aggregate Reporting Data Model.

### 4.1.3 Views

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

- `V$RPT_ALL_CLINICALSUMMARY`
4.1.4 Database Link
A default database link appears when you enable BI Publisher Periodic reporting. This link is created between the BI Publisher Owner schema and the BI Publisher Repository database and copies the report output blob to the Argus Safety database.

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

4.1.6 Packages
The BI Publisher Owner schema holds the following package types:
- Utility package
- Data load package
- User exits package

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

<table>
<thead>
<tr>
<th>Procedure/Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>f_get_query_details</td>
<td>Populates the case series or query prompts that the user can access.</td>
</tr>
<tr>
<td>f_print_as_text</td>
<td>Determines the water mark.</td>
</tr>
<tr>
<td>f_get_cs_name</td>
<td>Returns the case series name for a Case Series ID.</td>
</tr>
<tr>
<td>f_get_agency_name</td>
<td>Gets the agency name for the passed Agency ID.</td>
</tr>
<tr>
<td>p_fetchrptoutput</td>
<td>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.</td>
</tr>
<tr>
<td>f_get_duration</td>
<td>Returns a formatted duration for printing in a report. For example, 10 days.</td>
</tr>
<tr>
<td>f_get_codelist_val</td>
<td>Gets the code list display value.</td>
</tr>
<tr>
<td>f_get_enterprises</td>
<td>Gets the Active Enterprise list on the BIP console.</td>
</tr>
<tr>
<td>f_get_cmos_profile_flag</td>
<td>Fetches the cmn_profile value on key.</td>
</tr>
<tr>
<td>f_ConvertBlobToClob</td>
<td>Converts the blob data into clob.</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>Procedure/Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Variables</td>
<td>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.</td>
</tr>
<tr>
<td>Lexical Variables</td>
<td>Normal package variables described according to the lexical parameters used in the BIP report.</td>
</tr>
<tr>
<td>p_pop_psur_case_temp</td>
<td>Populates the temp table GTT_RPT_AGG_CASE.</td>
</tr>
<tr>
<td>p_pop_psur_drug_temp</td>
<td>Populates the temp table GTT_RPT_AGG_DRUG.</td>
</tr>
<tr>
<td>p_pop_psur_event_temp</td>
<td>Populates the temp table GTT_RPT_AGG_EVENT.</td>
</tr>
<tr>
<td>p_pop_psur_ev2drug_temp</td>
<td>Populates the table GTT_RPT_AGG_EV2DRUG.</td>
</tr>
<tr>
<td>p_pop_psur_healthauthids_temp</td>
<td>Populates the temp table GTT_RPT_AGG_HEALTHAUTHID.</td>
</tr>
<tr>
<td>p_pop_psur_drugnames_temp</td>
<td>Populates the table GTT_RPT_AGG_DRUGNAMES.</td>
</tr>
<tr>
<td>f_before_data</td>
<td>The main function invoked from BI Publisher. Called from the Before Report trigger of the BI Publisher report.</td>
</tr>
<tr>
<td>f_get_report_id</td>
<td>Retrieves the PN_REG_REPORT_ID parameter value.</td>
</tr>
<tr>
<td>p_set_report_id</td>
<td>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.</td>
</tr>
<tr>
<td>p_check_cs_case_ctr</td>
<td>Checks the counts of cases needed for the trailer page.</td>
</tr>
<tr>
<td>p_ins_rpt_status</td>
<td>Inserts the record into PER_RPT_STATUS for log reporting.</td>
</tr>
<tr>
<td>p_upd_rpt_status</td>
<td>Updates the status of the report on completion with success or failure.</td>
</tr>
<tr>
<td>f_after_report</td>
<td>Final trigger invoked by BI Publisher.</td>
</tr>
<tr>
<td>f_get EVTseriouscr_list</td>
<td>Gets the event seriousness criteria list.</td>
</tr>
<tr>
<td>f_get_dose_stringlist</td>
<td>Generates the dose string list.</td>
</tr>
<tr>
<td>f_get uniq_patient_id</td>
<td>Obtains the unique patient ID.</td>
</tr>
<tr>
<td>p_updclinicaldrugrole</td>
<td>Updates the clinicaldrugrole column in GTT_RPT_AGG_DRUG table.</td>
</tr>
<tr>
<td>p_update_gtt_tables</td>
<td>Updates the GTT tables for follow-up.</td>
</tr>
<tr>
<td>FindAggRptJobID</td>
<td>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.</td>
</tr>
<tr>
<td>pop_user_security_tables</td>
<td>Populates user security tables based on the user-security level.</td>
</tr>
<tr>
<td>p_populate_cover_params</td>
<td>Fills in the data for the RPT_AGG_PARAMS table needed for the cover page.</td>
</tr>
<tr>
<td>p_populate_listedness</td>
<td>Determines and populates listedness for each case-event-product based on the chosen algorithm.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Procedure/Function</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_modify_case_temp</td>
<td>Called at the end of the case population procedure. You can customize the populated cases here.</td>
</tr>
<tr>
<td>p_modify_event_temp</td>
<td>Called at the end of the event population procedure. You can customize the populated events here.</td>
</tr>
<tr>
<td>p_modify_drug_temp</td>
<td>Called at the end of the drug population procedure. You can customize the populated drugs here.</td>
</tr>
<tr>
<td>p_modify_evt_assess_temp</td>
<td>Called at the end of the assessment population procedure. You can customize the populated assessment here.</td>
</tr>
<tr>
<td>p_modify_healthauthids_temp</td>
<td>Called at the end of the health authority details population procedure. You can customize the populated health authority IDs here.</td>
</tr>
<tr>
<td>p_modify_drugnames_temp</td>
<td>Called at the end of the drug name details population procedure. You can customize the populated drug name here.</td>
</tr>
<tr>
<td>p_modify_rm_case_temp</td>
<td>Called after loading the RM_RPT_AGG_CASE table.</td>
</tr>
<tr>
<td>p_modify_rm_event_temp</td>
<td>Called after loading the RM_RPT_AGG_EVENT table.</td>
</tr>
<tr>
<td>p_modify_rm_drug_temp</td>
<td>Called after loading the RM_RPT_AGG_DRUG table.</td>
</tr>
<tr>
<td>p_modify_rm_evt_assess_temp</td>
<td>Called after loading the RM_RPT_AGG_EV2DRUG table.</td>
</tr>
<tr>
<td>p_modify_rm_healthauthids_temp</td>
<td>Called after loading the RM_RPT_AGG_HEALTHAUTHID table.</td>
</tr>
<tr>
<td>p_modify_rm_drugnames_temp</td>
<td>Called after loading the RM_RPT_AGG_DRUGNAMES table.</td>
</tr>
</tbody>
</table>
4.1.7 BI Publisher Owner Schema Database Jobs

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

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

**Job for Report Output Copying**

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

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

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

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

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

**Job for Persist Data Purging**

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

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

4.2 Argus Safety Database Updates

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

4.2.1 Table Changes Required

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

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

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

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

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

4.3 Access Grants to BI Publisher Owner

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

■ SELECT access for Case related, Code list, and Configuration tables.

■ SELECT, INSERT, UPDATE access for process tables such as CMN_REG_REPORTS, PER_RPT_QUEUE, PER_RPT_STS, CMN_SUB_REPORTS, and so on.

■ EXECUTE access for various common packages such as PKG_RLS and GSS_UTIL.

Note: You need access for compiling BIP Packages in the BIP Owner schema.
This chapter contains the following sections:

- Section 5.1, "BI Publisher Periodic Reports Design Structure"
- Section 5.2, "Report Data Flow"

5.1 BI Publisher Periodic Reports Design Structure

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

Each report in Oracle Argus FAR has categories:

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

5.1.1 BI Publisher Data Model

The BI Publisher data model comprises:

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

5.1.1.1 Data Sets

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

Argus Safety does not use date type parameters.

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

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

For example, for Main Case Series, when the parameter value is not selected and is left at -- SELECT --, the report selects the main case associated with the report configuration.
Enterprise ID and Report Configuration Name are mandatory parameters. The report fails if either of these parameters is not submitted.

5.1.2 BI Publisher Report Layout

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

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

Figure 5–4  PBRER Data Model

5.2 Report Data Flow

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

5.2.1 Argus Safety UI Entry

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

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

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

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

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

The case series are stored in the case series tables.
5.2.2 BI Publisher Data Flow

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

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

2. Select the mandatory parameter ENTERPRISE ID.
   This selection executes the associated LOV query.
   The selected value is verified by the query present in the CFG_RPT_AGG_PARMS table for the parameter.
   After verification, the enterprise ID is set for the session.

3. Select the mandatory parameter REPORT_CONFIGURATION.
   This parameter determines whether the report will be in the Data Lock Point (DLP).
   If the case series parameters are not selected, the default case series associated with this report configuration is set automatically.
   This parameter is an LOV and the query present in the CFG_RPT_AGG_PARMS table for the parameter validates the selected value.

4. Input the other parameters, and click Submit.
   This invokes the Before data trigger that calls the f_beforedata function of the pkg_agg_rpt package.

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

1. After the PLSQL code is complete, the BI report internally generates the data XML, renders the pages, and stores the output in its repository data.

2. The database job calls the `pkg_rpt_util.p_fetchrptoutput` function, which obtains the list of reports that are run based on the RPT_AGG_BIP_JOB table.

3. The database then queries the BIP Metadata Repository (MDR) and verifies that the report is complete. If the report has failed, the database updates the failure status and proceeds to the next report.

4. If the report is successful, the database checks if the Draft/Final tables are updated.

5. The system copies the report output blob to the reporting tables. It updates `CMN_REG_REPORTS` and the corresponding records in the `PER_RPT_QUEUE` and `PER_RPT_STATUS` tables.

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

---

**Note:** The report output in BIP might be completed earlier than in Argus UI. This is because the job picks up the output and pushes it to the Argus database.
This chapter describes the options available for extending an out-of-the-box BI Publisher Periodic report. It contains the following sections:

- Section 6.1, "Database Layer"
- Section 6.2, "Extending the BI Publisher Data Model"
- Section 6.3, "Creating a Custom 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.
### 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.

### 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>
<thead>
<tr>
<th>Procedure</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_modify_case_temp</td>
<td>Executed at the end of case population procedure. Customization to populated cases can be done here.</td>
</tr>
<tr>
<td>p_modify_event_temp</td>
<td>Executed at the end of event population procedure. Customization to populated cases can be done here.</td>
</tr>
<tr>
<td>p_modify_drug_temp</td>
<td>Executed at the end of drug population. Customization of populated drugs can be taken up here.</td>
</tr>
<tr>
<td>p_modify_evt_assess_temp</td>
<td>Executed at the end of event assessment population. Customization of populated event assessment data can be taken up here.</td>
</tr>
<tr>
<td>p_modify_healthauthids_temp</td>
<td>Executed at the end of Health authority details population.</td>
</tr>
<tr>
<td>p_modify_drugnames_temp</td>
<td>Executed at the end of drug name details population.</td>
</tr>
</tbody>
</table>

There is also a user exit for each RM table.

Figure 6–1 Extending with User Exits

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
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:

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

### 6.1.3 Extending Global Temporary Tables

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

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

You can fill these columns by using user exits.

### 6.1.4 Extending using Custom Objects

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

#### 6.1.4.1 Adding New Columns to GTT or RM Tables

To add new columns to GTT and RM tables:

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

#### 6.1.4.2 Filling Custom Tables

Follow the same procedure for filling custom tables. You do not need to replicate a table from the BIP schema.
6.1.4.3 Filling Custom Views
You might need to deploy custom views for solving complicated logistics. This functionality is limited to accessing the objects of the BIP schema only.

To fill custom views, perform the following steps:
1. Create the view in the custom schema.
2. Grant BIP schema access to the view.
3. Create a synonym for the view (public synonym).

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

6.1.4.4 Filling Custom Packages
Follow the same procedure for filling custom packages.

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

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

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

6.1.7 Adding a Column to the Existing Table
Temp tables provide custom columns for customer use. However, there might be instances when you want to add further tables. You can do this in the following ways:
- Adding the columns directly into the table and manipulating it.
- If you are not allowed to add directly, you can create a replica of the table in the custom schema along with the new columns needed.

The BIP schema is provided a grant for the new object and a synonym is created. Then, the data can be inserted in the new table.

6.2 Extending the BI Publisher Data Model

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

This section contains the following:
- Section 6.2.1, ”Data Model Query Naming Convention”
- Section 6.2.2, ”Data Model Nested Queries”
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

```sql
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

```sql
SELECT ct.reaction g2pbrer62reaction,
```

Oracle Argus Safety Flexible Aggregate Reporting Extensibility Guide
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>

<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"/>
Extending the BI Publisher Data Model

Oracle Argus Safety Flexible Aggregate Reporting Extensibility Guide

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_TEXTURE 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*
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.
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 (lexical) Used in Reports

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

<table>
<thead>
<tr>
<th>Table 6–3 pkg_agg_rpt Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_MESSAGE</td>
</tr>
<tr>
<td>Q_ERROR</td>
</tr>
<tr>
<td>Q_CASES</td>
</tr>
<tr>
<td>Q_REPORTCASESUMMARY</td>
</tr>
<tr>
<td>Q_WMARK</td>
</tr>
<tr>
<td>Q_MODCS</td>
</tr>
<tr>
<td>Q_HEAD</td>
</tr>
<tr>
<td>Q_TITLE</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Table 6–4 QA Section Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_QA1</td>
</tr>
<tr>
<td>Q_QA2</td>
</tr>
<tr>
<td>Q_QA3</td>
</tr>
<tr>
<td>Q_QA4</td>
</tr>
<tr>
<td>Q_QA5</td>
</tr>
<tr>
<td>Q_QA6</td>
</tr>
<tr>
<td>Q_QA7</td>
</tr>
<tr>
<td>Q_QA9</td>
</tr>
<tr>
<td>Q_QA13</td>
</tr>
<tr>
<td>Q_QA14</td>
</tr>
</tbody>
</table>
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.

   ```
   IF PN_LL_SOC = 1 THEN
   ```

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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GL_LL_MAIN_NONPRI_CASEREF</td>
<td>PMAR main line listing section uses this variable.</td>
</tr>
<tr>
<td>GL_LL_ADHOC1_NONPRI_CASEREF</td>
<td>PMAR Adhoc1 line listing section uses this variable.</td>
</tr>
<tr>
<td>GL_LL_ADHOC2_NONPRI_CASEREF</td>
<td>PMAR Adhoc2 line listing section uses this variable.</td>
</tr>
<tr>
<td>GL_LL_ADHOC3_NONPRI_CASEREF</td>
<td>PMAR Adhoc3 line listing section uses the variable.</td>
</tr>
<tr>
<td>GL_LL_ADHOC4_NONPRI_CASEREF</td>
<td>PMAR Adhoc4 line listing section uses the variable.</td>
</tr>
<tr>
<td>GL_LL_PRI_CASESOC_ONLY</td>
<td>To print case details under primary case SOC.</td>
</tr>
<tr>
<td>GL_LL_DSUR_PRI_CASESOC_ONLY</td>
<td>DSUR reports.</td>
</tr>
<tr>
<td>GL_LL_DSURMAIN_NONPRI_CASEREF</td>
<td>DSUR reports.</td>
</tr>
<tr>
<td>L_LL_DSURDTH_NONPRI_CASEREF</td>
<td>DSUR reports.</td>
</tr>
</tbody>
</table>
5. **Print Serious Adverse Events or Reactions**: Filters out related events.

```sql
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.
For information on Parameters, refer to Appendix A in the *Argus Safety BIP Aggregate Reporting User’s Guide.*
This appendix discusses the blinding functionality.

**B.1 Blinding Logic**

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

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

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

<table>
<thead>
<tr>
<th>Study Status</th>
<th>Case or Code Broken</th>
<th>User Access to Blinded Information</th>
<th>Print Unblinded Data</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blinded</td>
<td>Blinded</td>
<td>No</td>
<td>No</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Blinded</td>
<td>No</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Blinded</td>
<td>Yes</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Blinded</td>
<td>Yes</td>
<td>No</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Unblinded</td>
<td>No</td>
<td>No</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Unblinded</td>
<td>No</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Blinded</td>
<td>Unblinded</td>
<td>Yes</td>
<td>Yes</td>
<td>Unblinded</td>
</tr>
<tr>
<td>Blind</td>
<td>Blinded</td>
<td>No</td>
<td>No</td>
<td>Blinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Blinded</td>
<td>Yes</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Blinded</td>
<td>No</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Blinded</td>
<td>Yes</td>
<td>Yes</td>
<td>Blinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Blinded</td>
<td>Yes</td>
<td>No</td>
<td>Blinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Unblinded</td>
<td>No</td>
<td>No</td>
<td>Unblinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Unblinded</td>
<td>No</td>
<td>Yes</td>
<td>Unblinded</td>
</tr>
<tr>
<td>Unblinded</td>
<td>Unblinded</td>
<td>Yes</td>
<td>Yes</td>
<td>Unblinded</td>
</tr>
</tbody>
</table>
Flexible Data Re-Categorization

This appendix describes flexible data re-categorization.

C.1 Flexible Data Re-Categorization

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

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

The following code lists are provided for BIP aggregate reports:

- **REPORT_TEMPLATE**: Use this code list to manage the BIP report templates available in the system (OOB and Custom) and assign an Argus Safety periodic configuration (ICH PSUR or CTPR) with which you can execute this report template. Whenever a new report template is added in BIP, the system modifies this code list to provide the report template name, path and corresponding Argus Safety configuration.

- **ADHOC_LINE_LISTING**: All list names added to the `adhoc_line_listing` code list are available in the UD Summaries tab and can be used to attach memorized reports to a particular line listing section of periodic reports. You can rename these using the Flexible Re-categorization UI. The system provides four ad hoc line listings by default. You can increase them using this code list if you have more ad hoc line listing sections in your report. After configuring them, use the Argus Safety UI to attach UD summaries with line listings of your custom report.

- **SOC_DISPLAY_ORDER**: Use this code list to reorder the printing of SOCs in various tabulations.

- **STATE_2, STATE_3 and STATE_4**: These code lists have been modified to print follow-up text, relatedness text, and so on. You can further modify these code list and add new attributes for use in custom reports. For example, you can use the CAUSAL attribute for printing the RELATEDNESS of an event. These are printed as Yes/No or Related/Unrelated using this attribute.

- **AGGREGATE_REPORT_FORMAT**: Use this code list to define the report formats to use with BIP reports. Ensure that you use formats that are supported by the BI Publisher.

- **DOSAGE_STRING_FORMAT**: Use this code list to restrict or add the dosage string formats that are available while BIP report is being executed. You can add more dosage string formats using custom code.
Table C-1 contains the dosage string formats provided out-of-the-box and the print values for each dosage string.

<table>
<thead>
<tr>
<th>Code</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do</td>
<td>Dose</td>
</tr>
<tr>
<td>DoFo</td>
<td>Dose, Formulation</td>
</tr>
<tr>
<td>DoFoFr</td>
<td>Dose, Formulation, Frequency</td>
</tr>
<tr>
<td>DoFoFrRt</td>
<td>Dose, Formulation, Frequency, Route</td>
</tr>
<tr>
<td>DoFoRt</td>
<td>Dose, Formulation, Route</td>
</tr>
<tr>
<td>DoFr</td>
<td>Dose, Frequency</td>
</tr>
<tr>
<td>DoRt</td>
<td>Dose, Route</td>
</tr>
</tbody>
</table>

**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>
<thead>
<tr>
<th>Code</th>
<th>EN</th>
</tr>
</thead>
<tbody>
<tr>
<td>CePt</td>
<td>Center, Patient</td>
</tr>
<tr>
<td>InPt</td>
<td>Investigator, Patient</td>
</tr>
<tr>
<td>Pt</td>
<td>Patient</td>
</tr>
<tr>
<td>StCeInPt</td>
<td>Study, Center, Investigator, Patient</td>
</tr>
<tr>
<td>StCePt</td>
<td>Study, Center, Patient</td>
</tr>
<tr>
<td>StCnCeInPt</td>
<td>Study, Country name, Center, Investigator, Patient</td>
</tr>
<tr>
<td>StCnCePt</td>
<td>Study, Country name, Center, Patient</td>
</tr>
<tr>
<td>StCoCeInPt</td>
<td>Study, Country ISO Code, Center, Investigator, Patient</td>
</tr>
<tr>
<td>StCoCePt</td>
<td>Study, Country ISO code, Center, Patient</td>
</tr>
<tr>
<td>StInPt</td>
<td>Study, Investigator, Patient</td>
</tr>
</tbody>
</table>

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

- **BIP Icon not Displayed Against Corresponding BIP Specific Fields in the Periodic Reporting Configuration**
- **BI Publisher Periodic report does not Execute from Argus UI**
- **Database Connection Errors**
- **BI Publisher Scheduled Report Output does not Show up**
- **ORA-01427: single-row sub-query returns more than one row ORA-06512: at 
"<bip_owner>.PKG_AGG_RPT" for p_updclinicaldrugrole procedure**

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

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

- **Cause**
  
  BI Publisher Periodic reports are not enabled.

- **Solution**

  On the Argus Console, navigate to System Configuration > Enabled Modules. Make sure that BIP Aggregate Reporting is checked.
Figure D–1  BIP Aggregate Reporting Enabled

- **Cause**
  The page level cache is not refreshed.
  
- **Solution**
  Restart the IIS server.

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

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

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

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

- **Cause**
  In Linux/Unix the report path is case sensitive. The Report path in the REPORT_TEMPLATE is empty or has an incorrect path.

- **Solution**
  The Report path present in the code list REPORT_TEMPLATE must have a proper value. The value can be verified by comparing it against the actual report path in the BIP Publisher catalogs.
Figure D–2 Verify Report Template Path

<table>
<thead>
<tr>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The endpoint address present in the AGProc.config file is incorrect.</td>
<td>Confirm the following:</td>
</tr>
<tr>
<td></td>
<td>– The endpoint address is present.</td>
</tr>
<tr>
<td></td>
<td>– It points to the BI Publisher Server (including the port number).</td>
</tr>
<tr>
<td></td>
<td>– No unnecessary space is present in the URL.</td>
</tr>
<tr>
<td></td>
<td>– Services are properly mentioned—SecurityService is mapped against</td>
</tr>
<tr>
<td></td>
<td>SecurityService, and ScheduleService is mapped against SchedulingService.</td>
</tr>
<tr>
<td>The Oracle DataAccess version present in the AGProc.config file is</td>
<td>Refer to the Install guide and make sure the correct entry is provided.</td>
</tr>
<tr>
<td>incorrect.</td>
<td></td>
</tr>
<tr>
<td>The BI Publisher report cannot utilize the JDBC provided.</td>
<td>Make sure that the Argus UI user exists and is allowed to login and run the required reports.</td>
</tr>
</tbody>
</table>
Note: Sometimes, customer may have configured each service associated to one exe files. In such scenarios, the AG_Proc config related to Peridic Reporting should be modified.

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

Make sure:

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

D.5 ORA-01427: single-row sub-query returns more than one row
ORA-06512: at "<bip_owner>.PKG_AGG_RPT" for p_updclinicaldrugrole procedure

Cause

The issue has been caused due to the study configuration having duplicated attributes.

Make sure:

To check the study configuration and ensure that it does not have any duplicates.