Oracle Financial Services Trade-Based Anti Money Laundering

Administration Guide

Release 8.1.2.7.0

February 2024

E98716-01

ORACLE

Financial Services

Table of Contents

OFS TBAML

Document Control

Table 1: Revision History

Date	Edition	Description
February 2024	First edition of 8.1.2.7	Updated this guide to reflect the data structure of the 8.1.2.7 release.
November 2021	Second edition of 8.0.8.0.0	Removed references to Swift and EDQ support.
December 2019	First edition of 8.0.8.0.0	In Chapter 4, *Behavior Detection Jobs*, added a note to the *Setting Environment Variables* section.
May 2019	Second edition of 8.0.7.0.0	In Chapter 7, *Managing Administrative Utilities*, added the *Adding a New Threshold Set* section.
December 2018	First edition of 8.0.7.0.0	In Chapter 9, *Creating JSON*, removed sections, *Example of MT 101 with Sequences*, and *Example of MT 101 without Sequences*. In Appendix F, TBAML Datamap Details, updated the following tables to reflect datamap changes: • Trade Finance - Pre-Watch List Datamaps • Trade Finance - Post-Watch List Datamaps • Watch List Datamaps
August 2018	First edition of 8.0.6.0.0	First publication of this document.

0 About this Guide

This guide explains the concepts behind Oracle Financial Services Trade-Based Anti Money Laundering (TBAML), and provides comprehensive instructions for proper system administration, as well as daily operations and maintenance. This section focuses on the following topics:

0.1 Who Should Use this Guide

This *Administration Guide* is designed for use by the Installers and System Administrators. Their roles and responsibilities, as they operate within TBAML, include the following:

- **Installer:** Installs and configures TBAML at a specific deployment site. The Installer also installs and upgrades any additional Oracle Financial Services solution sets and requires access to deployment-specific configuration information, such as machine names and port numbers).
- **System Administrator:** Configures, maintains, and adjusts the system, and is usually an employee of a specific Oracle customer. The System Administrator maintains user accounts and roles, monitors data management and event management, archives data, loads data feeds, and performs post-processing tasks. In addition, the System Administrator can reload cache.

0.1.1 Prerequisites for an Administrator User

User must have knowledge of UNIX and LINUX.

0.2 Scope of this Guide

This guide describes the physical and logical architecture of TBAML. It also provides instructions for installing and configuring TBAML, its subsystem components, and any third-party software required for operation.

TBAML is powered by advanced data mining algorithms and sophisticated pattern recognition technologies. It provides an open and scalable infrastructure that supports rich, end-to-end functionality across all Oracle Financial Services solution sets. TBAML's extensible, modular architecture enables a customer to deploy new solution sets readily as the need arises.

0.3 How this Guide is Organized

The *Oracle Financial Services TBAML Administration Guide*, includes the following chapters:

- Chapter 1, About TBAML, provides a brief overview of the Behavior Detection Platform Framework and its components.
- Chapter 2, Managing User Administration and Security Configuration, covers the required day-to-day operations and maintenance of TBAML users, groups, and organizational units.

- Chapter 3, Managing Data,, describes the operation and process flow of data management subsystem components.
- Chapter 4, Behavior Detection Jobs, provides an overview of the BDF job protocol and procedures for performing various tasks that relate to starting, stopping, and recovering jobs.
- Chapter 5, Post-Processing Tasks, explains how to customize the TBAML features that affect presentation of user information on the desktop.
- Chapter 6, Managing Batch Processing Utilities, provides information about the TBAML utilities related to the batch process.
- Chapter 7, Managing Administrative Utilities, provides information about the TBAML utilities that are independent of the batch process.
- Appendix A, Logging, describes the TBAML logging features.
- Appendix B, Oracle Software Updates, describes the application of Oracle software updates (hotfix) and their impact on customization.
- Appendix C, User Administration, describes the user administration of TBAML.
- Appendix D, Managing Data, describes the BDF file parameters, the FSDF datamaps, the Data Quality group names and related T2T names, the BDF interface files, and the directory structures.
- Appendix E, Processing Derived Tables and Fields, describes the additional data processing activities that can be performed.
- Appendix F, TBAML Datamap Details lists the Datamap XML and their use in TBAML.

0.4 Where to Find More Information

For more information about Behavior Detection Platform, refer to the following TBAML application documents, which can be found at https://docs.oracle.com/cd/E60570_01/tbamlhome.htm:

- Trade-Based Anti Money Laundering Data Interface Specification (DIS)
- TBAML Installation Guide
- Trade-Based Anti Money Laundering Matching Guide

Additionally, you may find pertinent information in other OFSAAI documentation, found at the following link:

http://docs.oracle.com/cd/E60058_01/homepage.htm:

- Oracle Financial Services Analytical Applications Infrastructure User Guide
- Oracle Financial Services Analytical Applications Infrastructure Installation and Configuration
- Administration Tools User Guide

For installation and configuration information about Sun Java System, BEA, and Apache software, refer to the appropriate documentation that is available on the associated websites.

0.5 Conventions Used in this Guide

This table lists the conventions used in this guide and their associated meanings.

Table 2: Conventions Used in this Guide

Convention	Meaning
Italics	• Names of books, chapters, and sections as references • Emphasis
Bold	• Object of an action (menu names, field names, options, button names) in a step-by-step procedure • Commands typed at a prompt • User input
`Monospace`	• Directories and subdirectories • File names and extensions • Process names • Code sample, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text
<Variable>	• Substitute input value

0.6 Abbreviations Used in this Guide

This table lists the abbreviations used in this guide and their associated descriptions.

Table 3: Abbreviations Used in this Guide

Abbreviation	Description
TBAML	Oracle Financial Services Trade-Based Anti Money Laundering
OFSBD	Oracle Financial Services Behavior Detection
T2T	Table to Table
AAI	Analytical Applications Infrastructure
CSA	Common Staging Area
FCDM	Financial Crime Data Model
BDF	Behavior Detection Framework
OFS	Oracle Financial Services
DQ	Data Quality
DT	Data Transformation

1 About TBAML

This chapter provides a brief overview of Oracle Financial Services Trade-Based Anti Money Laundering (TBAML) in terms of its architecture and operations.

This chapter focuses on the following topics:

- TBAML Architecture
- Operations
- Utilities

1.1 About TBAML

Oracle Financial Services Trade-Based Anti Money Laundering (TBAML) offers a comprehensive compliance solution to:

- Efficiently screen goods, ports and involved parties against various lists such as sanctions lists, watch lists, and so on.
- Continuously monitor trade finance transactions using a risk based approach for potential TBML activities, such as TBML red flag topologies, by assessing the trade finance customer, transactions (specifically goods, contract amount, goods price), and involved counterparties (name and address).

1.1.1 TBAML Case Workflow

The following figure describes how TBAML cases are investigated through Oracle Financial Services Enterprise Case Management (ECM).

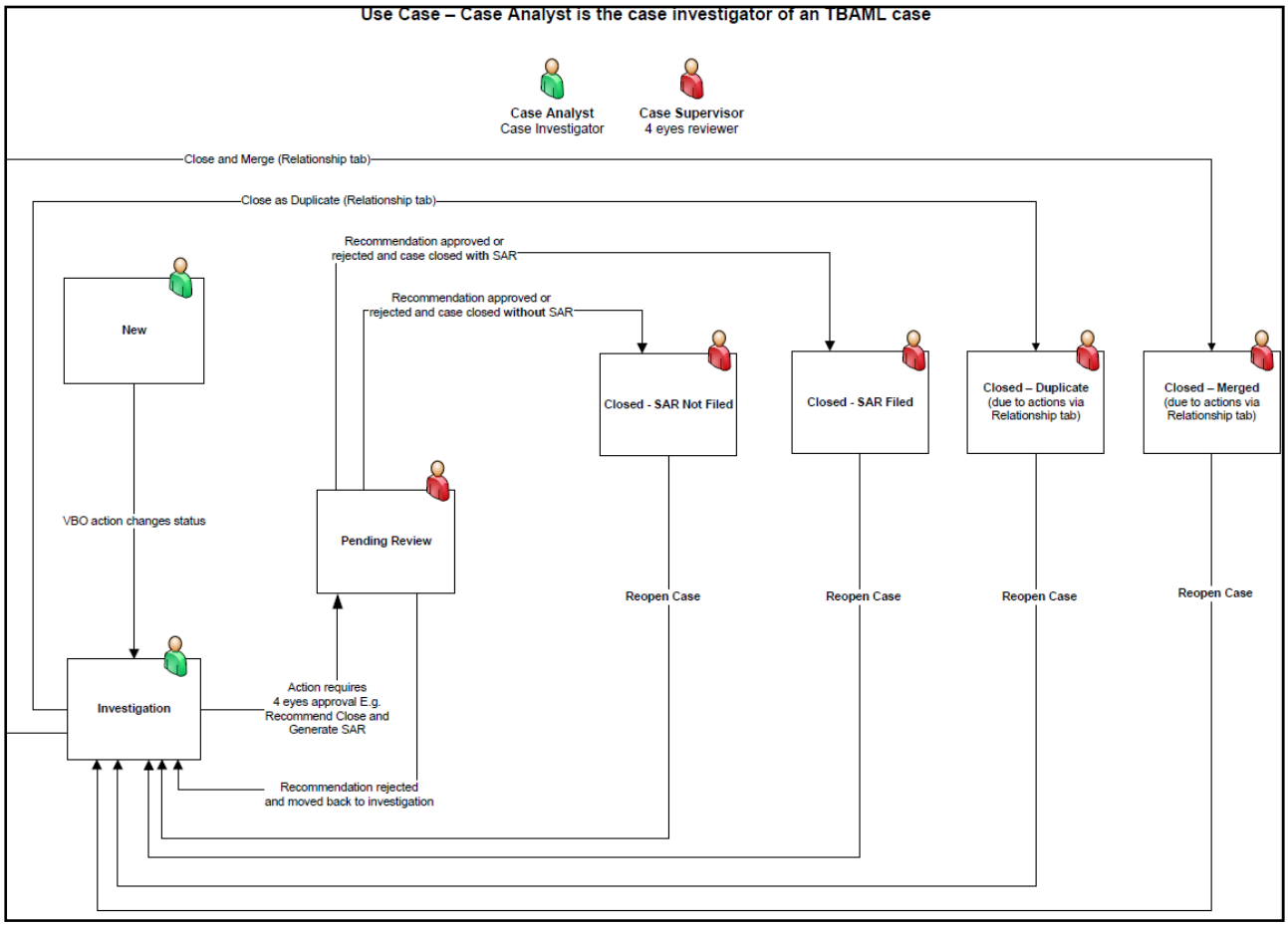


Figure 1: TBAML Case Workflow

For more information about managing TBAML cases in ECM, refer to the Enterprise Case Management User Guide.

1.2 TBAML Architecture

An architecture is a blueprint of all the parts that together define the system: its structure, interfaces, and communication mechanisms. A set of functional views can describe an architecture.

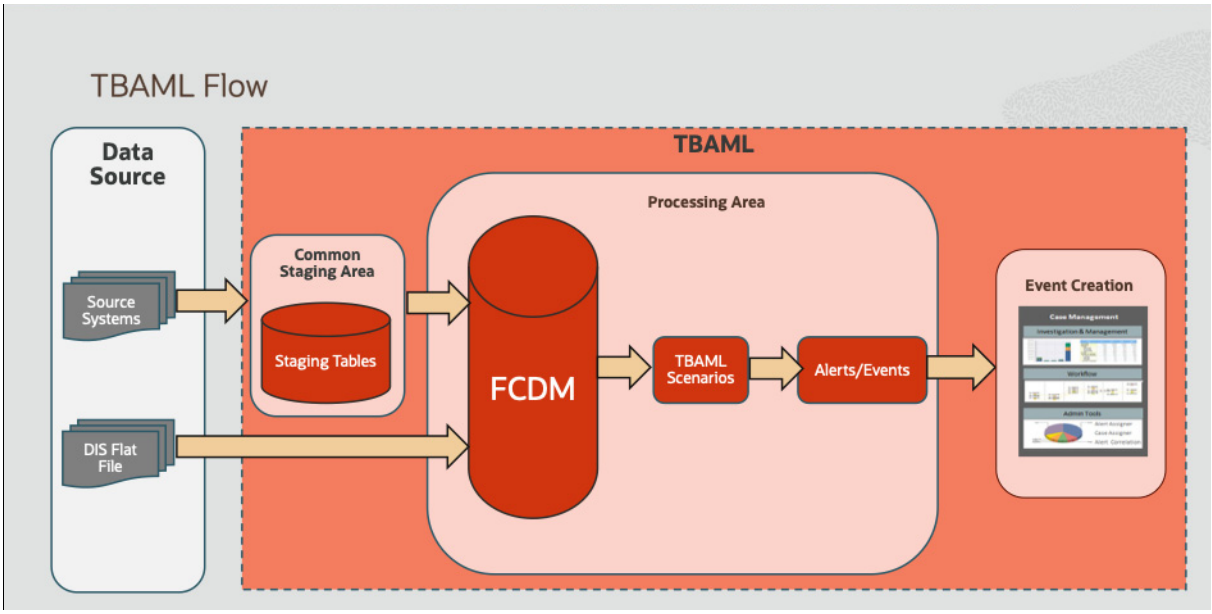


Figure 2: TBAML Architecture

TBAML extracts data provided by the Oracle client via DIS File or another source, where it is fed into staging tables and then into the FCDM where the data is either standardized (Port) and screened (Port, Goods, Name and Address) or run through scenarios to generate an FCM event.

1.2.1 Deployment View

The TBAML architecture from the perspective of its deployment illustrates deployment of the major subsystems across servers. Additionally, the deployment view shows the primary communications links and protocols between the processing nodes.

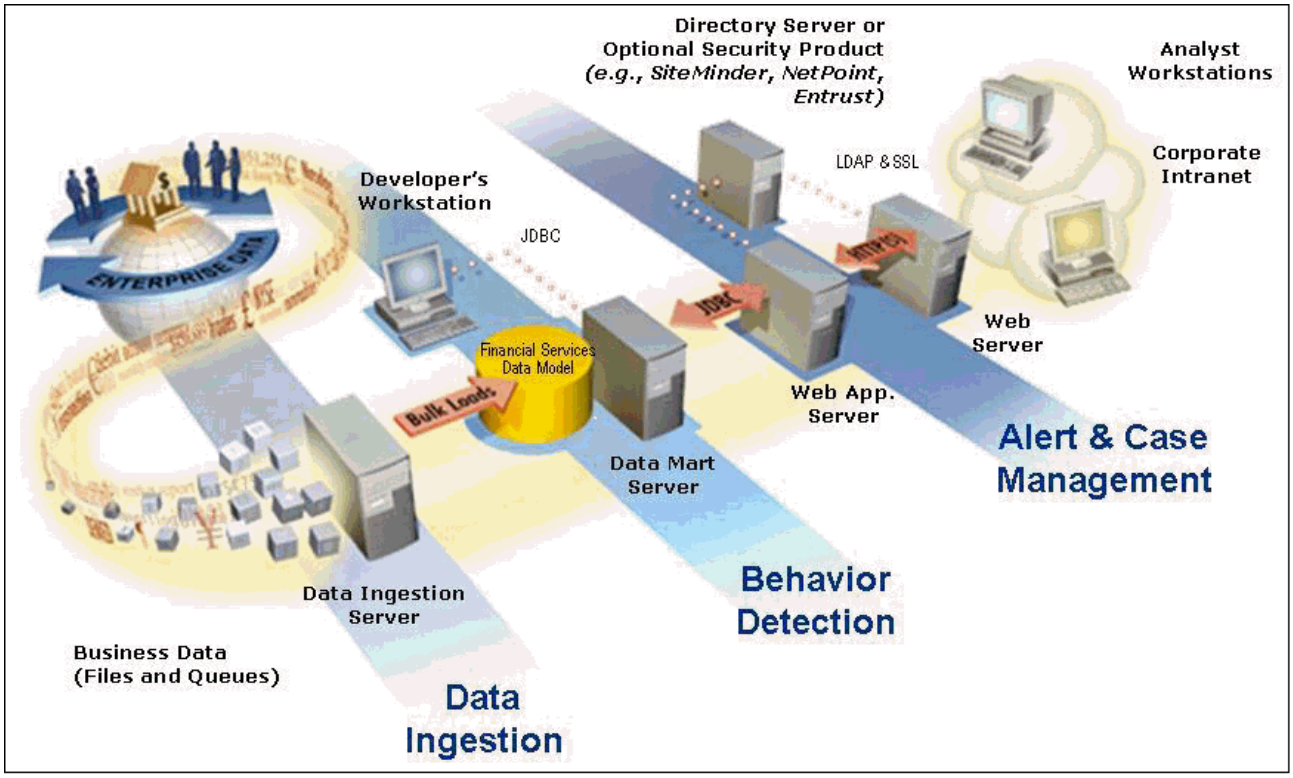


Figure 3: TBAML Architecture - Deployment View

The complex interactions between the components of the Alert & Case Management tiers becomes apparent in the deployment view. The Alert & Case Management tiers require the following:

- Web browser
- Web server
- Web application server

Alert & Case Management tiers use OFSAAI for handling both authentication and authorization. The Alert & Case Management subsystem also supports the use of an External Authentication Management (EAM) tool to perform user authentication at the web server, if a customer requires it.

TBAML components can operate when deployed on a single computer or when distributed across multiple computers. In addition to being horizontally scalable, TBAML is vertically scalable in that replication of each of the components can occur across multiple servers.

1.2.2 Security View

The security view describes the architecture and use of security features of the network in a TBAML architecture deployment. TBAML uses an inbuilt Security Management System (SMS) for its authentication and authorization. The SMS has a set of database tables which store information about user authentication.

Installation of 128-bit encryption support from Microsoft can secure the web browser. Oracle encourages using the Secure Socket Layer (SSL) between the web browser and web server for login transaction, while the web Application server uses a browser cookie to track a user's session. This cookie is temporary and resides only in browser memory. When the user closes the browser, the system deletes the cookie automatically.

TBAML uses Advanced Encryption Standard (AES) security to encrypt passwords that reside in database tables in the ATOMIC schema on the database server and also encrypts the passwords that reside in configuration files on the server.

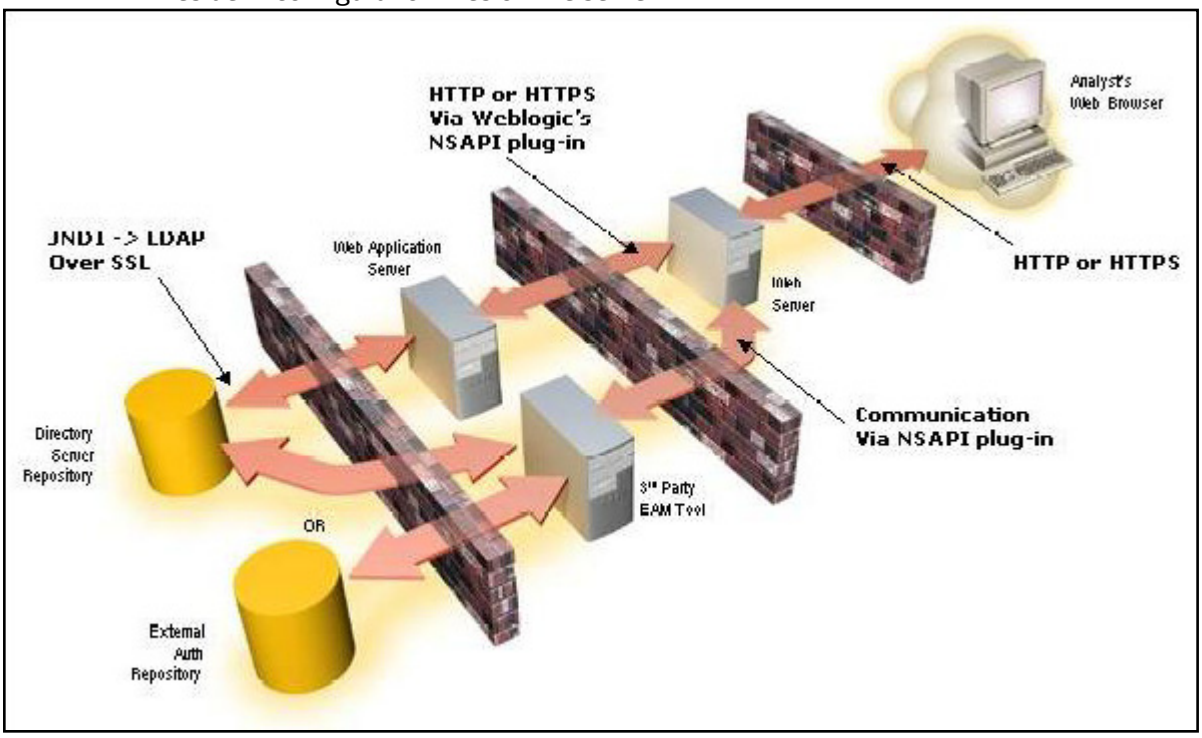


Figure 4: Security View

The EAM tool is an optional third-party pluggable component of the security view. The tool's integration boundaries provide an Authorization header, form field with principal, or embedded principal to the web Application server through a web server plug-in. The tool also passes the same user IDs that the TBAML directory server uses.

1.3 Operations

As the administrator, you coordinate the overall operations of TBAML: Data Management, Behavior Detection, and Post-Processing.

In a production environment, an Oracle client typically establishes a processing cycle to identify occurrences of behaviors of interest (that is, scenarios) at a specific frequency.

Each cycle begins with Data Management, Behavior Detection, and Post-Processing, which prepares the detection results for presentation for the users.

Several factors determine specific scheduling of these processing cycles, including availability of data and the nature of the behavior that the system is to detect. The following sections describe each of the major steps in a typical production processing cycle:

- Start Batch
- Managing Data
- Behavior Detection
- Post-Processing
- End Batch

1.3.1 Start Batch

Using the Batch Control Utility, you can manage the beginning of the batch process (see Managing Batch Processing Utilities for more information).

1.3.2 Managing Data

The Ingestion Manager controls the Data Management process. The Data Interface Specification (DIS) contains specific definition of the types and format of business data that can be accepted for ingestion.

The Ingestion Manager supports files and messages for the ingestion of data. Data Management involves receiving source data from an external data source in one of these forms. The Ingestion Manager validates this data against the *DIS*, applies required derivations and aggregations, and populates the database with the results (see Managing Data for more information).

1.3.3 Behavior Detection

During Behavior Detection, OFSBD Algorithms control the scenario detection process. The Detection Algorithms search for events and behaviors of interest in the ingested data in the FCDM. Upon identification of an event or behavior of interest, the algorithms record a match in the database.

A match is created by executing scenarios. These scenarios are used to detect the behaviors of interest that correspond to patterns or the occurrences of prespecified conditions in business data. The process also records additional data that the analysis of each match may require.

1.3.4 Post-Processing

During post-processing of detection results, Behavior Detection prepares the detection results for presentation to users. Preparation of the results depends upon the following processes:

- **Match Scoring**: Computes a ranking for scenario matches indicating a degree of risk associated with the detected event or behavior.
- **Alert Creation**: Packages the scenario matches as units of work (that is, events), potentially grouping similar matches together, for disposition by end users. This is applicable when multiple matches with distinct scores are grouped into a single event.
- **Alert Scoring**: Ranks the events (including each match within the events) to indicate the degree of risk associated with the detected event or behavior.
- **Highlight Generation**: Generates highlights for events that appear in the event list in the behavior detection subsystem and stores them in the database.
- **Historical Data Copy**: Identifies the records against which the current batch's scenario runs generated events and copies them to archive tables. This allows for the display of a snapshot of information as of the time the event behavior was detected.
- **Alert Correlation:** Uncovers relationships among events by correlating events to business entities and subsequently correlating events to each other based on these business entities. The relationships are discovered based on configurable correlation rule sets.

1.3.5 End Batch

The system ends batch processing when processing of data from the Oracle client is complete (see Ending a Batch Process for more information). The Alert & Case Management subsystem then controls the event and case management processes. See the Behavior Detection User Guide and Enterprise Case Management User Guide for more information.

1.4 Utilities

TBAML database utilities enable you to configure and perform pre-processing and post-processing activities. The following sections describe these utilities.

- Batch Utilities
- Administrative Utilities

1.4.1 Batch Utilities

Behavior Detection database utilities enable you to configure and perform batch-related system pre-processing and post-processing activities.

- **Alert Purge Utility**: Provides the capability to remove erroneously generated matches, events, and activities.
- **Batch Control Utility**: Manages the start and termination of a batch process (from Data Management to event post-processing) and enables access to the currently running batch.
- **Calendar Manager Utility**: Updates calendars in the system based on pre-defined business days, holidays, and *days off*, or non-business days.
- **Data Retention Manager:** Provides the capability to manage the processing of partitioned tables in Behavior Detection. This utility purges data from the system based on configurable retention period defined in database.
- **Database Statistics Management**: Manages Oracle database statistics. These statistics determine the appropriate execution path for each database query.
- **Notification**: Enables you to configure users to receive UI notifications based upon actions taken on events or cases to which they are associated or when the event or case is nearing a due date.
- **Truncate Manager**: Truncates tables that require complete replacement of their data.

For more information on Administrative Utilities, see Managing Batch Processing Utilities.

1.4.2 Administrative Utilities

The following database utilities that configure and perform system pre-processing and post-processing activities are not tied to the batch process cycle:

- **Scenario Migration Utility:** Extracts scenarios, datasets, networks, and associated metadata from a database to flat files and loads them into another environment.
- **Threshold Editor:** Allows you to run the same scenario multiple times against a variety of sources (for example, exchanges, currencies, or jurisdictions) with separate threshold values for each source.

  For more information on Administrative Utilities, see Managing Administrative Utilities.

2 Managing User Administration and Security Configuration

This chapter provides instructions for setting up and configuring the Security Management System (SMS) to support Oracle Financial Services applications, user authentication, and authorization.

This chapter focuses on the following topics:

2.1 About User Administration

User administration involves creating and managing users and providing access rights based on their roles. This section discusses the following:

- Administrator permissions
- Creating and mapping users and user groups
- Loading and mapping security attributes

2.2 Administrator User Privileges

The following table lists the access permissions of administrators under TBAML:

Table 4: Access Permissions for Administrators

Privileges	Case Administrator User Role
User Security Administration	x
Preferences	x
User Administration	x
Security Management System	x
Data Management Tools	x
Unified Metadata Manager	x

2.3 User Provisioning Process Flow

The following table lists the various actions and associated descriptions of the user administration process flow:

Table 5: User Provisioning Process Flow

Action	Description
Managing User Administration	Create users and map users to user groups. This allows Administrators to provide access, monitor, and administer users.

2.3.1 Requirements to Access TBAML

A user gains access to TBAML based on the authentication of a unique user ID and password.

To access the TBAML applications, you must fulfill the following conditions:

Table 6: Requirements

Applications	Conditions
TBAML	• Set of privileges that associate functional role with access to specific system functions. • One or more associated organizational affiliations that control the user's access to events. • Relationship to one or more scenario groups. • Access to one or more jurisdictions. • Access to one or more business domains.

2.4 Managing User Administration

This section allows you to create, map, and authorize users defining a security framework which has the ability to restrict access to the respective Oracle applications.

2.4.1 Managing Identity and Authorization

This section explains how to create a user and provide access to Oracle applications.

This section covers the following topics:

- Managing Identity and Authorization Process Flow
- Creating and Authorizing Users and User Groups
- Mapping Users with User Groups

2.4.1.1 Managing Identity and Authorization Process Flow

The following figure shows the process flow of identity management and authorization:

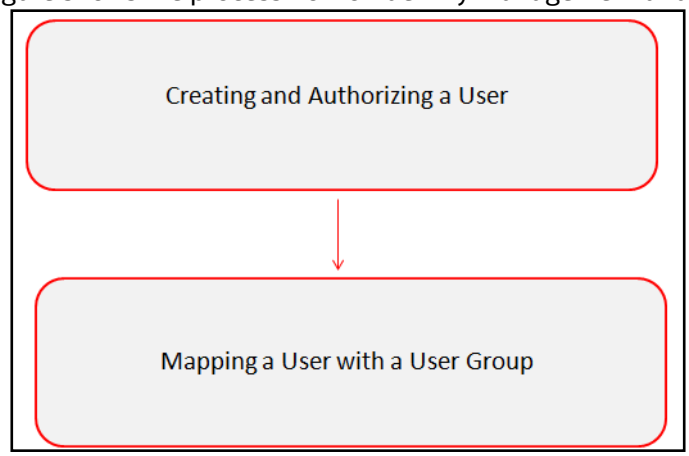


Figure 5: Managing Identity and Authorization Process Flow

The following table lists the various actions and associated descriptions of the user administration process flow:

Table 7: Administration Process Flow

Action	Description
Creating and Authorizing Users and User Groups	Create a user. This involves providing a user name, user designation, and the dates between which the user is active in the system.
Mapping Users with User Groups	Map a user to a user group. This enables the user to have certain privileges that the mapped user group has.

2.4.1.2 Creating and Authorizing Users and User Groups

The SYSADMN and SYSAUTH roles can be provided to users in the TBAML application. User and role associations are established using Security Management System (SMS) and are stored in the config schema. User security attribute associations are defined using Security Attribute Administration.

For more information on creating and authorizing a user, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.

2.4.1.3 Mapping Users with User Groups

This section explains how to map Users and User Groups. With this, the user will have access to the privileges as per the role. The SYSADMN user maps a user to a user group in the TBAML application. The following table describes the predefined User Roles and corresponding User Groups.

Table 8: TBAML Roles and User Groups

Role	Group Name	User Group Code
Case Administrator	Case Administrator User Group	CMMANADMNUG
Case Supervisor	Case Supervisor User Group	CMSUPERVISORUG

If you want to change the user group mapping for users who are already mapped to one or more groups, you must deselect the preferences for the Home page if it has been set. To change the preferences, follow these steps:

1. In the Home page, click the user name. A drop-down list appears.
2. Click **Preferences**. The Preferences page appears.
3. Select the appropriate Property Value.
4. Click **Save**.

For customized user group creation and user group-role mapping, see Appendix C, User Administration.

3 Managing Data

This chapter explains how your raw business data can be loaded into the Oracle Financial Crime Data Model (FCDM) in various ways. The following approaches are available either through the OFSDF Common Staging Area Model (CSA) or converting the raw data into Data Interface Specification (DIS) flat files.

This chapter focuses on the following topics:

3.1 About Data Management

Data Management consists of two main activities:

- **Data Loading**: Data is loaded into the Financial Crime Data Model (FCDM) using various approaches such as Analytical Applications Infrastructure Table-to-table (AAI T2T).
- **Data Processing**: Data loaded into the FCDM is processed for data derivation and data aggregation using the BDF processing datamaps. The processing refers to the wide range of activities to include data enrichment and data transformation.

3.2 Data Loading and Processing Flow Overview

The following figure provides an overview of the data loading and processing flow:

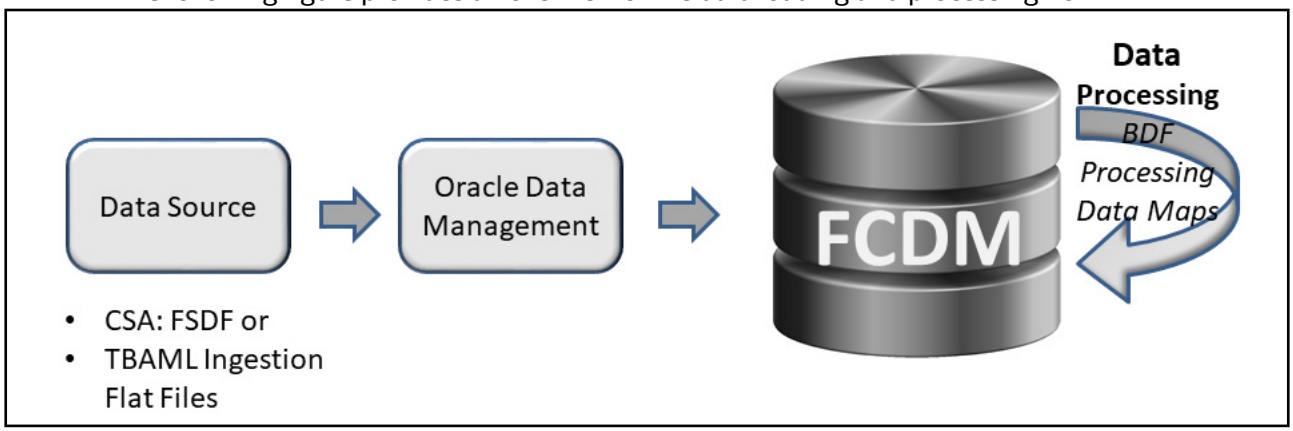


Figure 6: Data Loading and Processing Flow Overview

In TBAML, data is loaded into the FCDM from the following data sources:

- Common Staging Area (CSA) in FSDF
- TBAML Flat File Interface

Data stored in the FCDM is then processed using processing datamaps where additional data derivations and aggregations are stored in the FCDM.

3.2.1 CSA

The CSA provides a single repository for data storage for multiple functional areas and applications having the Common Staging Area Model and Reporting Data Model. The Common Staging Area Model provides a simplified, unified data sourcing area for inputs required by TBAML using BDF.

3.2.2 Flat Files

The flat files contain data provided by the client. This data is loaded into the Financial Crime Data Model (FCDM).

3.2.3 FCDM

The FCDM is a database which consists of well organized business data for analysis. It determines the structured data which stores persistent information in a relational database and is specified in a data modeling language.

3.2.4 Datamaps

The datamaps load Business and Reference data required for event processing. It does the data derivation and aggregation after BDF loads the base tables.

3.3 Managing Data Loading

Your raw business data can be loaded into the Oracle Financial Crime Data Model (FCDM) in various ways. The following approaches are available either through the OFSDF Common Staging Area Model (CSA) or converting the raw data into Data Interface Specification (DIS) files.

The following approaches are used to load the data:

- FSDF CSA Data Load
- Ingestion Flat File Data Load
- Managing Data Processing

3.3.1 FSDF CSA Data Load

This section covers the following topics:

- Overview
- Using Datamaps

3.3.2 Overview

The CSA Model provides a simplified, unified data sourcing area for inputs required by Oracle. It is the common data sourcing layer across all OFSAA applications and the OFSDF. The following figure provides an overview of the data loading flow using CSA:

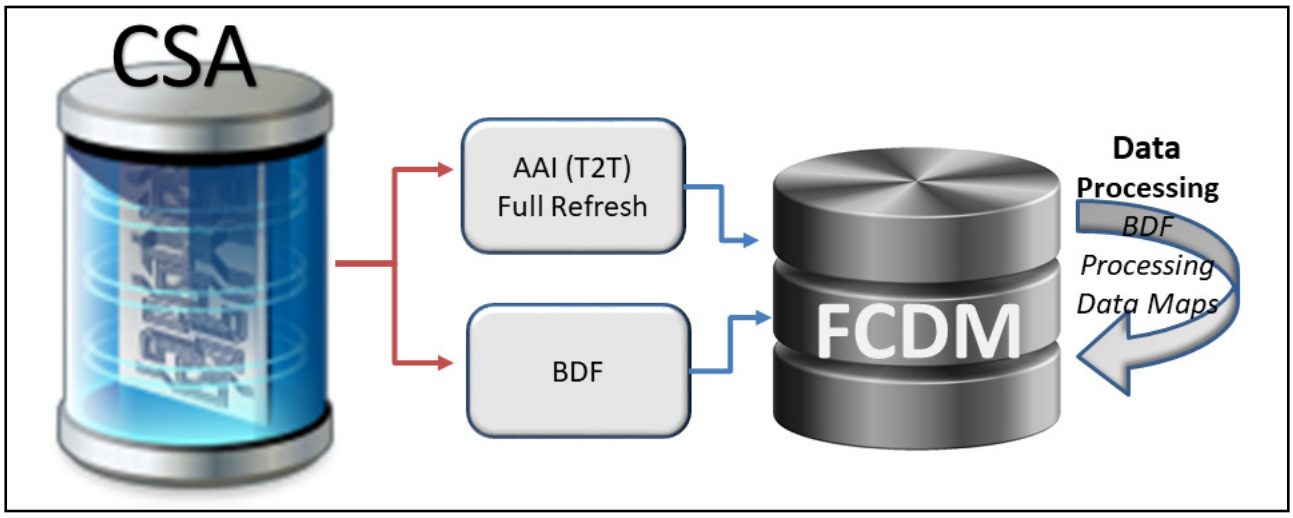


Figure 7: Data Management Flow Using CSA

3.3.2.1 Ingestion Flat File Data Load

The loading process receives, transforms, and loads Business and Reference data that event detection and assessment investigation processing requires. After loading the base tables, the Oracle client's job scheduling system invokes datamaps to derive and aggregate data.

This section covers the following topics:

- Overview
- Using TBAML Datamaps

3.3.2.1.1 Overview

The following figure provides an overview of the data management flow using Flat File Interface:

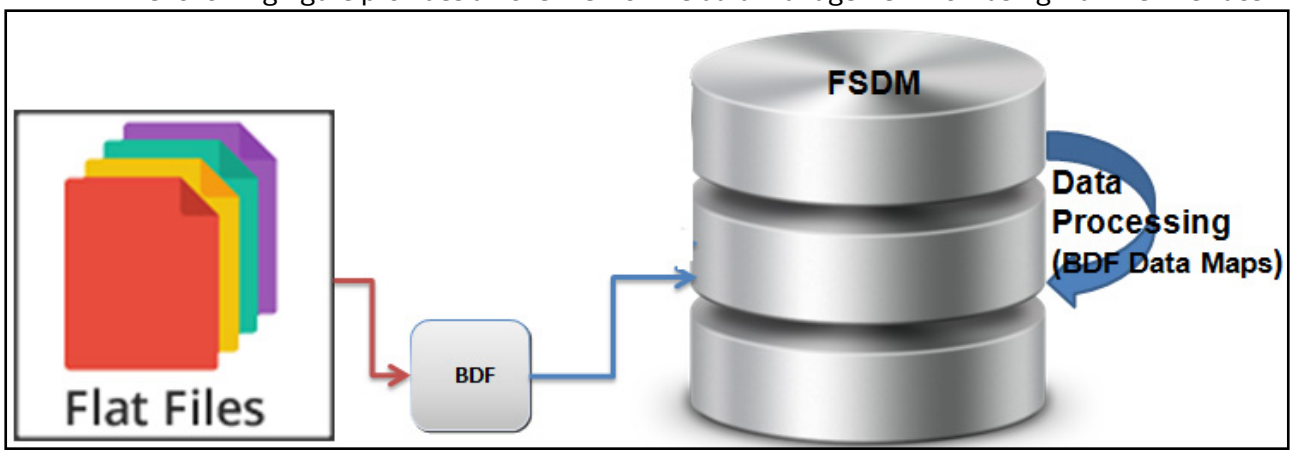


Figure 8: Data Loading Flow Using Flat File Interface

> **NOTE** All DIS datamaps in the Flat File Interface for which staging representation is marked as Yes are applicable for Flat File loading. For more information, see TBAML Flat File Interface.

3.3.2.1.2 Using TBAML Datamaps

The datamap takes the data from the flat files, enhances it, and then loads it into a target database table (FCDM).

To load data in the FCDM using Flat Files, follow these steps:

1. Place the `ASCII.dat` flat files in the `<OFSAAI Installed Directory>/bdf/inbox` directory.
2. Configure the `DIS.source` parameter to FILE.

   Configure the `DIS.Source` parameter to `FILE-EXT` for loading flat files through the external table. In order to load the flat files using the external table, the `ext_tab_dir_path` variable must also be set to the inbox directory and the database UNIX account must have read and write privileges to it.
3. Execute the Account datamap which loads into the Account (ACCT) table:

   ```
   <OFSAAI Installed Directory>/bdf/scripts/execute.sh Account
   ```

> **NOTE** If there are any errors in loading, refer to the <OFSAAI Installed Directory>/bdf/logs path.

3.3.2.1.3 Ways of Data Loading

This section covers the following topics:

- Full Refresh Data Loading
- Incremental (Delta) Data Loading

> **NOTE** The following ways of data loading is applicable only for DIS files defined with load operation as Overwrite.

3.3.2.1.3.1 Full Refresh Data Loading

For full refresh data loading, first data is truncated and then new data is inserted. For example, suppose five records are loaded on Day 1. If new data is required on Day 2 based on the business keys defined on the DIS files, a full refresh data load can be done.

To do a full refresh data load, set `load.fullrefresh` to true in the `<OFSAAI Installed Directory>/bdf/config/BDF.xml` path. For more information, see BDF.xml Configuration Parameters.

The time taken to do a full refresh data load is less than for an incremental load, although complete data must be provided every time.

3.3.2.1.3.2 Incremental (Delta) Data Loading

For incremental data loading, the following can be done:

- Data can be merged
- Existing data can be updated
- New data can be inserted

For example, five records are loaded on Day 1. If four new records need to be inserted and one existing record needs to be updated based on the business keys defined on the DIS files, an incremental data load can be done.

To do an incremental data load, set `load.fullrefresh` to false in the `<OFSAAI Installed Directory>/bdf/config/BDF.xml` path. For more information, see BDF.xml Configuration Parameters.

> **NOTE** It takes more time to do an incremental data load than a full refresh data load, although there is no need to give complete data every time. Only updated or new data is required.

3.3.3 Managing Data Processing

This section explains the concept of data processing and various methods of data processing. It covers the following topics:

- About Datamaps
- Datamap Categories
- Datamap Categories
- Datamaps
- Datamaps

3.3.3.1 About Datamaps

The datamap component is responsible for taking data from one or more source files or staging tables, transforming and enhancing it, and then loading it into a target database table.

The following types of datamaps are available:

- **DIS datamaps**: DIS datamaps are used to ingest client provided data, either through DIS files as specified in the DIS or through tables in the FSDF.
- **Derived datamaps**: Derived datamaps are used to transform the client provided data and populate other tables for use by scenarios and/or UI functionality.

Datamaps can perform the following activities:

- Update summaries of transaction activity
- Assign transaction and entity risk through watch list processing
- Update various Balances and Positions derived attributes
- Update data related to Trade Finance attributes

For a complete list of the datamaps used in OFSAAI and a brief explanation of the each datamap, see Appendix F, TBAML Datamap Details.

3.3.3.2 Datamap Categories

Each datamap can include one or more of the following categories:

- Optional
- Pre-watch List
- Watch List
- Post-watch List
- Summary

- Balances And Positions

> **NOTE** The Datamap categories may or may not be required for all solutions.

3.3.4 Datamaps

For detailed information about the datamaps that are required for deriving and aggregating data for the TBAML Solution, see the following sections:

- Trade Finance - Pre-Watch List Datamaps
- Trade Finance- Post-Watch List Datamaps

3.4 Managing Data For TBAML

This section explains different methods used to load and process data. Figure 9 shows the sequence for data loading:

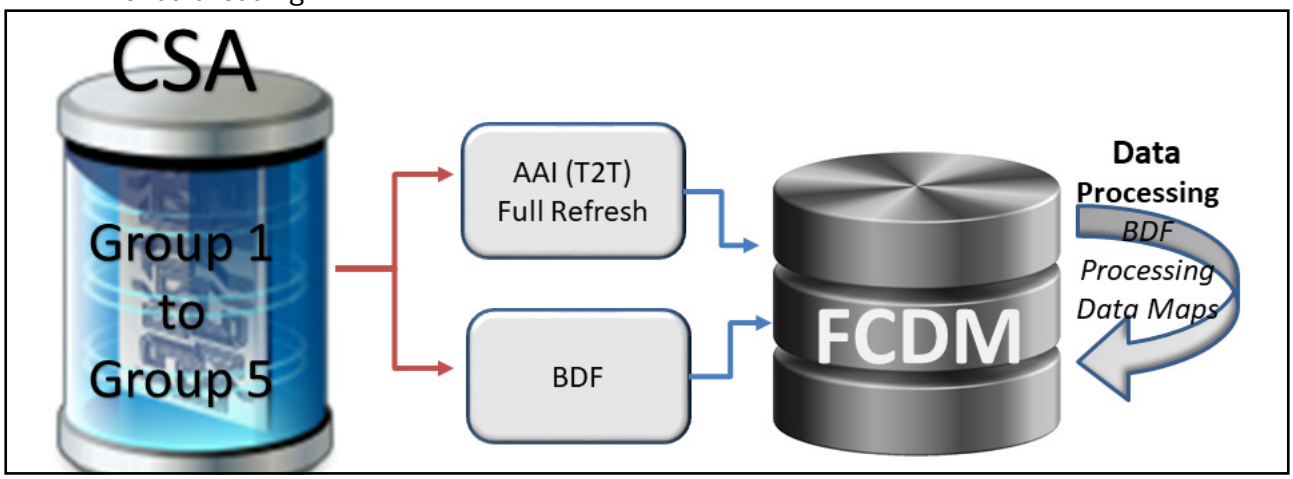


Figure 9: Data Loading For TBAML Application

The following table provides the steps required to load data for TBAML. .

Table 9: Managing Application Data

Steps	Group
1. Execute Group 1 through Group 6 in sequence in the CSA. For more information on the interface files available in Group 1 to Group 5, see TBAML Flat File Interface. 1. Process the loaded data using datamaps in FCDM. For more information, see Managing Data Processing. 2. Interface files in the same group loaded through different loading method can be executed in parallel. 3. Run AM transformation. 4. For network scenarios, refresh the temporary tables.	Group 1 Group 2 Group 3 Group 4 Group 5
1. Execute Group 7 using FDT/MDT. For more information, see Table . 5. Process the derived datamaps for TBAML. For more information, see Managing Data.	Group 6

3.0.1 Post Load Changes

For more information about the Post Load Changes Data Management tool in the TBAML UI, see the *Oracle Financial Services Analytical Applications Infrastructure User Guide.*

4 Behavior Detection Jobs

This chapter provides an overview of the OFSBD Job Protocol and explains how the System Administrator monitors jobs, and starts and stops jobs when necessary. In addition, it describes the necessary scripts that you use for OFSBD jobs. This chapter focuses on the following topics:

- About the OFSBD Job Protocol
- Performing Dispatcher Tasks
- Performing Job Tasks
- Clearing Out the System Logs
- Recovering Jobs from a System Crash
- Executing Batches Through the OFSAAI User Interface

> **NOTE** If you are using a job script that allows for multiple parameters, the values for the parameters must be separated by spaces ( ) and not commas (,).

4.1 About the OFSBD Job Protocol

The system initiates all OFSBD jobs by using a standard operational protocol that utilizes each job's metadata, which resides in a standard set of database tables. OFSBD Job Protocol processes include the following:

- `Dispatcher`**:** Polls the job metadata for new jobs that are ready for execution. This daemon process starts a MANTAS process for each new job.
- `Mantas`**:** Creates a new job entry based on a template for the job that has the specific parameters for this execution of the job (that is, it clones a new job).

The OFSBD administrator invokes the `dispatcher` and MANTAS processes by running the shell scripts that are mentioned in the following table

Table 10: OFSBD Job Protocol Shell Scripts

OFSBD Job Protocol Process Shell Script	Description
start_mantas.sh	Starts all OFSBD jobs. This script invokes the cloner and MANTAS processes. This is the integration point for a third-party scheduling tool such as Maestro or AutoSys.
start_chkdisp.sh	Calls on the check_dispatch.sh script to ensure that the *dispatcher* runs.
stop_chkdisp.sh	Stops the ***dispatcher*** process.
restart_mantas.sh	Changes job status codes from the ERR status to the RES status so that the ***dispatcher*** can pick up the jobs with the RES status.
recover_mantas.sh	Changes job status codes for jobs that were running at the time of a system crash to the ERR status. After running this script, the `restart_mantas.sh` script must be run to change the ERR status code to RES in order for the **dispatcher** to be able to pick up these jobs.

In the OFSBD Job Protocol, the processes use a variety of metadata that the OFSBD database provides. Some of this metadata specifies the jobs and their parameters that are associated with

the regular operations of an OFSBD installation. Some of this metadata captures the status of job execution and is useful for monitoring the progress of an OFSBD operational cycle.

This section covers the following topics:

- Understanding the OFSBD Job Protocol
- Understanding the Dispatcher Process
- Understanding the MANTAS Process
- Applying a Dataset Override

4.1.1 Understanding the OFSBD Job Protocol

OFSBD Jobs are created through the Scenario Manager. Jobs are grouped together to run in parallel through Job Template Groups in the `KDD_JOB_TEMPLATE` table. These templates associate an algorithm to run with parameters that the algorithm requires. Template groups enable you to identify what jobs to run.

The following table provides an example of a job template group with two job templates.

Table 11: `KDD_JOB_TEMPLATE` with Sample Job Template Group

JOB_ID	TEMPLATE_GROUP_ID
37	1
41	1

4.1.2 Understanding the Dispatcher Process

The `dispatcher` process polls the job metadata waiting for jobs that must be run. To control system load, the *`dispatcher`* also controls the number of jobs that run in parallel.

Generally, the dispatcher process should be running continuously, although it is possible to run jobs without a dispatcher.

For each job in the template group, the dispatcher runs a MANTAS process. The `dispatcher` tracks jobs for status and completion, and reports any failure to the dispatch log.

> **NOTE** If you observe job failures when running on the AIX operating system, it may be due to resource constraints of the AIX system. In this case, you must try reducing the number of jobs you are attempting to run in parallel or try running the jobs sequentially.

Refer to Starting the Dispatcher and Stopping the Dispatcher for more information.

4.1.3 Understanding the MANTAS Process

The `dispatcher` runs jobs using the MANTAS process. This process runs the appropriate algorithm, tracks status in the `KDD_JOB` and `KDD_RUN` tables. One MANTAS process can result in multiple `KDD_RUN` records.

The MANTAS process also logs job progress and final status.

4.1.4 Applying a Dataset Override

The dataset override feature permits dataset customizations specific to your site, which can be retained outside of the scenario metadata. The override to a dataset definition is stored in a file accessible by the Behavior Detection engine. The dataset override feature allows improved performance tuning and the ability to add filters that are applicable only to your site's dataset.

When the system runs a job, it retrieves the dataset definition from the database. The Behavior Detection engine looks in the configured directory to locate the defined dataset override. The engine uses the override copy of the dataset instead of the copy stored in the scenario definition in the database, if a dataset override is specified.

The following constraints apply to overriding a dataset:

- The columns returned by the dataset override must be identical to those returned by the product dataset. Therefore, the dataset override does not support returning different columns for a pattern customization to use.
- The dataset override can use fewer thresholds than the product dataset, but cannot have more thresholds than the product dataset. Only thresholds applied in the dataset from the scenario are applied.

If a dataset override is present for a particular dataset, the override applies to all jobs that use the dataset.

4.1.4.1 Configuring the Dataset Override Feature

To configure a dataset override, follow these steps:

4. Modify the `install.cfg` file for algorithms to identify the directory where override datasets are stored.

   The file resides in the following directory:

   ```
   <OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/
   mantas_cfg/
   install.cfg
   ```

   The dataset override is specified with this property:

   ```
   kdd.custom.dataset.dir
   ```

   > **NOTE** Specify the directory for the above given property using a full directory path, not a relative path. If you do not (or this property is not in the `install.cfg` file), the system disables the dataset override automatically.

5. Create the dataset override file in the specified directory with the following naming convention:

   ```
   dataset<DATASET_ID>.txt
   ```

   The contents of the file should start with the SQL definition in `KDD_DATASET.SQL_TX`. This SQL must contain all of the thresholds still represented such as `@Min_Indiv_Trxn_Am`.

4.2 Performing Dispatcher Tasks

The **dispatcher** service runs on the server on which TBAML is installed. Once the dispatcher starts, it runs continuously unless a reason warrants shutting it down or it fails due to a problem in TBAML

This section covers the following topics:

- Setting Environment Variables
- Starting the Dispatcher
- Stopping the Dispatcher
- Monitoring the Dispatcher

4.2.1 Setting Environment Variables

Environment variables are set up during the installation process. These generally do not require modification thereafter.

All behavior detection scripts and processes use the `system.env` file to establish their environment.

4.2.1.1 About the System.env File

The following table describes environment variables in the `system.env` file. This file can be found at `<OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/share`

Table 12: OFSBD Environment Variables in system.env File

Variable	Description
`KDD_HOME`	Install path of the Oracle software.
`KDD_PRODUCT_HOME`	Install path of the solution set. This is a directory under `KDD_HOME`.

The following table describes database environment variables in the `system.env` file.

Table 13: Database Environment Variables in system.env File

Variable	Environment	Description
`ORACLE_HOME`	Oracle	Identifies the base directory for the Oracle binaries. You must include: • `$ORACLE_HOME` and `$ORACLE_HOME/bin` in the PATH environment variable value. • `$ORACLE_HOME/lib` in the `LD_LIBRARY_PATH` environment variable value.
`ORACLE_SID`	Oracle	Identifies the default Oracle database ID/name to which the application connects.
`TNS_ADMIN`	Oracle	Identifies the directory for the Oracle network connectivity, typically specifying the connection information (SID, Host, Port) for accessing Oracle databases through SQL*NET.

The following table shows operating system variables in the `system.env` file.

Table 14: Operating System Environment Variables in system.env File

Variable	Description
PATH	Augmented to include `<OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/bin` and the `$ORACLE_HOME, $ORACLE_HOME/bin` pair (for Oracle).
LD_LIBRARY_PATH, LIBPATH, SHLIB_PATH (based on operating system)	Augmented to include `<OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/lib` and `$ORACLE_HOME/lib` (for Oracle)

NOTE To support C++ binaries which are built for 12c , create a symbolic link inside the $ORACLE_HOME/lib folder as follows:

```
$ cd $ORACLE_HOME/lib
$ ln -s  libclntsh.so.18.1
libclntsh.so.12.1
```

4.2.2 Starting the Dispatcher

Although multiple jobs and MANTAS instances can run concurrently, only one `dispatcher` service per database per installation should run at one time.

Oracle provides a script to check the status of the `dispatcher` automatically and restart it, if necessary. Oracle recommends this method of running the `dispatcher`.

To start the `dispatcher`, follow these steps:

1. Verify that the dispatcher is not already running by typing `ps -ef | grep dispatch` and pressing **Enter** at the system prompt.

   If the dispatcher is running, an instance of the dispatcher appears on the screen for the server. If the dispatcher is not running, proceed to Step 2.

2. Type `start_chkdisp.sh <sleep time>` and press **Enter** at the system prompt to start the `dispatcher`.

   The dispatcher queries the database to check for any new jobs that must be run. In between these checks, the dispatcher sleeps for the time that you specify through the `<sleep time>` parameter (in minutes).

   Optional parameters include the following:

   - `dispatch name`: Provides a unique name for each `dispatcher` when running multiple `dispatcher`s on one machine.
   - `JVM size`: Indicates the amount of memory to allocate to Java processing.

   The script executes and ends quickly. The dispatcher starts and continues to run in the background.

4.2.3 Stopping the Dispatcher

You do not normally shut down the dispatcher except for reasons such as the following:

- Problems while executing scenarios, make it necessary to stop processing.

- The dispatcher and job processes are reporting errors.
- The dispatcher is not performing as expected.
- You must shut down the system for scheduled maintenance.
- You want to run the `start_mantas.sh`, `restart_mantas.sh`, or `recover_mantas.sh` script without the dispatcher already running. You can then save your log files to the server on which you are working rather than the server running the dispatcher.

> **NOTE** The dispatcher which started from the Behavior Detection jobs in the UI should be stopped before restarting servers.

> **ATTENTION** If you shut down the dispatcher, all active jobs shut down with errors.

When you are ready to restart the dispatcher and you want to see which jobs had real errors and which jobs generated errors only because they were shut down during processing, review the error messages in the job logs.

For those jobs that shut down and generate errors because the dispatcher shut down, a message similar to the following appears: `Received message from dispatcher to abort job`. If the job generates a real error, a message in the job log file indicates the nature of the problem.

To view active jobs and then shut down the dispatcher, follow these steps:

1. Type **`ps -efw | grep mantas`** and press **Enter** at the system prompt.

   All instances of the MANTAS process that are running appear on the screen. Only one instance of MANTAS should run for each active job.

2. Type **`stop_chkdisp.sh <`**dispatcher **`name>`** and press **Enter** at the system prompt.

   This script shuts down the dispatcher.

4.2.4 Monitoring the Dispatcher

The `install.cfg` file that was set up during server installation contains the `kdd.dispatch.joblogdir` property that points to a log file directory. The log directory is a repository that holds a time-stamped record of `dispatcher` and job processing events.

Each time the dispatcher starts or completes a job, it writes a status message to a file called `dispatch.log` in the log directory. This log also records any failed jobs and internal dispatcher errors. The `dispatch.log` file holds a time-stamped history of events for all jobs in the chronological sequence that each event occurred.

To monitor the `dispatch.log` file as it receives entries, follow these steps:

1. Change directories to the log directory.
2. Type **`tail -f dispatch.log`** and press **Enter** at the system prompt.

   The log file scrolls down the screen.

3. Press **Ctrl+C** to stop viewing the log file.
4. Type **`lpr dispatch.log`** and press **Enter** at the system prompt to print the `dispatch.log` file.

> **ATTENTION** The dispatch.log file can be a lengthy printout.

4.3 Performing Job Tasks

At the system level, the Oracle administrator can start, restart, copy, stop, monitor, and diagnose jobs.

This section cover the following topics:

4.3.1 Understanding the Job Status Codes

The following status codes are applicable to job processing and the dispatcher. The administrator sets these codes through an OFSBD Job Editor:

- **NEW (start):** Indicates a new job that is ready to be processed.
- **RES (restart):** Indicates that restarting the existing job is necessary.
- **IGN (ignore):** Indicates that the dispatcher should ignore the job and not process it. This status identifies Job Templates.

The following status codes appear in the `KDD_JOB` table when a job is processing:

- **RUN (running):** Implies that the job is running.
- **FIN (finished):** Indicates that the job finished without errors.
- **ERR (error):** Implies that the job terminated due to an error.

4.3.2 Starting Behavior Detection Jobs

The administrator starts jobs by running the `start_mantas.sh` script.

To start a new job, follow these steps:

1. Create the new job and job description through an OFSBD Job Editor in the Scenario Manager.

   OFSBD automatically assigns a unique ID to the job when it is created.
2. Associate the new job to a Job Template Group using the `KDD_JOB_TEMPLATE` table (Refer to section Understanding the OFSBD Job Protocol for more information).
3. Execute the `start_mantas.sh` script as follows:

   ```
   start_mantas.sh <template id>
   ```

The following events occur automatically:

1. The job goes into the job queue.
2. The dispatcher starts the job in turn, invoking the MANTAS process and passing the job ID and the thread count to the MANTAS process.
3. The MANTAS process creates the run entries in the OFSBD metadata tables. Each job consists of one or more runs.

4. The MANTAS process handles the job runs.

After a job runs successfully, you can no longer copy, edit, or delete the job. The `start_mantas.sh` script waits for all jobs in the template group to complete.

4.3.3 Starting Jobs Without the Dispatcher

Clients who use multiple services to run jobs for one OFSBD database must run the jobs without dispatcher processes. If the client does use dispatchers on each machine, each dispatcher may run each job, which causes duplicate detection results.

To run a job template without a dispatcher, add the parameter `-nd` to the command line after the template ID, as follows:

```
start_mantas.sh <template id> -nd
```

Doing so causes the `start_mantas.sh` script to execute all jobs in the template, rather than depending on the dispatcher to run them. The jobs in the template group run in parallel.

The dispatcher can ensure that it is only running a set number of max jobs at any given time (so if the max is set to 10 and a template has 20 jobs associated to it, only 10 run simultaneously). When running without the dispatcher, you must ensure that the number of jobs running do not overload the system. In the event a job run dies unexpectedly (that is, not through a caught exception but rather a fatal signal), you must manually verify whether any jobs are in the RUN state but do not have a MANTAS process still running, which would mean that the job threw a signal. You must update the status code to ERR to restart the job.

To start a new job in Behavior Detection Framework without the **`dispatcher`**, follow these steps:

1. Create the new job and job description through an OFSBD Job Editor.

   OFSBD automatically assigns a unique ID to the job when it is created.

2. Associate the job to a Job Template Group using the `KDD_JOB_TEMPLATE` table.

3. Execute the `start_mantas.sh` script with the following parameters:

   ```
   start_mantas.sh <template id> [-sd DD-MON-YYYY]
   [-ed DD-MON-YYYY] [-nd]
   ```

   where the optional job parameters `-sd` and `-ed` (start date and end date, respectively) are used to constrain the data that an algorithm job pulls back.

   For example, if these parameters are passed into an Alert Creator job, the Alert Creator considers only matches for a grouping that has a creation date within the range that the parameters specify.

After a job runs successfully in OFSBD, you can no longer copy, edit, or delete the job.

4.3.4 Restarting a Job

Restarting a job is necessary when one or both of the following occurs:

- The dispatcher generates errors and stops during MANTAS processing. When the dispatcher is running, the OFSBD administrator can restart a job (or jobs) by changing each job's status code from ERR to RES.
- A job generates errors and stops during MANTAS processing. If a job stops processing due to errors, correct the problems that caused the errors in the job run and restart the job.

If the dispatcher stops, all jobs stop. You must restart the dispatcher and restart all jobs, including the job that generated real errors.

To restart a job, follow these steps:

> **NOTE** If the dispatcher has stopped, restart it.

1. Type `restart_mantas.sh <template group id>` at the system prompt.
2. Press **Enter**.

   When the dispatcher picks up a job from the job queue that has a code of RES, it automatically restarts the job (Refer to section Starting Behavior Detection Jobs for more information).

   By default, the `restart_mantas.sh` script looks for jobs run on the current day. To restart a job that was run on a specific date, you must provide the optional date parameter such as `restart_mantas.sh <template group id> <DD-MON-YYYY>`.

4.3.5 Restarting Jobs Without the Dispatcher

Restarting a job without the dispatcher is necessary when a job generates errors and stops during MANTAS processing. If a job stops processing due to errors, correct the problems that caused the errors in the job run and restart the job.

To start a new job, execute the `restart_mantas.sh` script with the following parameters:

```
restart_mantas.sh <template id> [-sd DD-MON-YYYY] [-ed DD-MON-YYYY] [-nd]
```

where the optional job parameters `-sd` and `-ed` (start date and end date, respectively) are used to constrain the data that an algorithm job pulls back.

4.3.6 Stopping Jobs

It may be necessary to stop one or more job processes when dispatcher errors, job errors, or some other event make it impossible or impractical to continue processing. In addition to stopping the processes, administrative intervention may be necessary to resolve the cause of the errors.

To stop a job, you must stop its associated MANTAS process. To obtain the process IDs of active jobs and `mantas` processes, follow these steps:

1. Type **`ps -efw | grep mantas`** and press **Enter** at the system prompt.

   The MANTAS processes that are running appear on the computer screen as shown in the following example:

   ```
   00000306 7800 1843   0 Jul 16   ttyiQ/iAQM 0:00
    /kdd_data1/kdd/server/bin/mantas -j 123
   ```

   The MANTAS process ID number appears in the first display line in the second column from the left (7800). The job ID number appears in the second display line in the last column (-j 123).
2. Find the job and MANTAS process ID that you want to stop.
3. Type **`kill <mantas process ID>`** at the system prompt and press **Enter**.

   This command stops the MANTAS process ID, which also stops its associated job.

4.3.7 Monitoring and Diagnosing Jobs

In addition to the `dispatch.log` file that records events for all jobs, the system creates a job log for each job. A job log records only the events that are applicable to that specific job. By default, a job log resides in the `$KDD_PRODUCT_HOME/logs` directory. You can configure the location of this log in the

<OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/mantas_cfg/install.cfg file.

> **NOTE** $KDD_PRODUCT_HOME is the path of <OFSAAI Installed Directory>/behavior_detection/algorithms/MTS

If you do not know the location of the log directory, check the install.cfg file. The log.mantaslog.location property indicates the log location. The default is $KDD_PRODUCT_HOME/logs, but this location is configurable.

When troubleshooting a job processing problem, first look at the file dispatch.log for the sequence of events that occurred before and after errors resulted from a job. Then, look at the job log to diagnose the cause of the errors. The job log provides detailed error information and clues that can help you determine why the job failed or generated errors.

The log file name for a job appears in the following format in the log directory:

job<job_id>-<date>-<time>.log

where <job_id> is the job ID and <date> and <time> represent the job's starting timestamp.

If the job errors occurred due to a problem at the system level, you may must resolve it. If you believe that the job errors were generated due to incorrect setups in OFSBD, you should notify the System Administrator, who can correct the problem setups.

> **NOTE** The dispatch.log may contain a JVM core dump. This does not indicate the actual cause of an error. In order to find the underlying error, you must refer to the job log.

To monitor a specific job or to look at the job log history for diagnostic purposes, follow these steps:

1. Type **tail -f <log>** at the system prompt and press **Enter**, where <log> is the name of the job log file.

   The job log scrolls down the screen.
2. Press **Ctrl+C** to stop the display.
3. Type **lpr** job<job_id>-<date>-<time> at the system prompt and press **Enter** to print the job log.

> **ATTENTION** This job log file may be a lengthy printout.

4.4 Clearing Out the System Logs

Periodically, you must clear out the dispatch and job log files. Otherwise, the files become so large that they are difficult to use as diagnostic tools and their size can impact the performance of the system.

> **TIP** Oracle recommends that the Oracle client establish a policy as to the frequency for clearing the logs and whether to archive them before clearing.

> **ATTENTION** Before you shut down the dispatcher to clear the system logs, verify that no jobs are active.

This section covers the following topics:

- Clearing the Dispatch Log
- Clearing the Job Logs

4.4.1 Clearing the Dispatch Log

To clear the `dispatch.log` file, follow these steps:

1. Shut down the `dispatcher` by following the procedure for Stopping the dispatcher (Refer to section Stopping the Dispatcher for more information).
2. Type `cd <$KDD_PRODUCT_HOME>/logs` at the system prompt, where `<$KDD_PRODUCT_HOME>` is your product server installation directory.
3. Type `rm dispatch.log` to clear the dispatcher log.
4. Type **`start_chkdisp.sh <sleep time>`** and press **Enter** to restart the dispatcher.

   Refer to Starting the Dispatcher for more information.

4.4.2 Clearing the Job Logs

To clear the job logs, follow these steps:

1. Stop the `dispatcher`. (Refer to section Stopping the Dispatcher for more information).
2. Type `cd <directory>` at the system prompt, where `<directory>` is your log directory.

   By default, a job log resides in the directory `$KDD_PRODUCT_HOME/logs`. You can configure the location of this log in the `<OFSAAI Installed Directory>/behavior_detection/algorithms/MTS/mantas_cfg/install.cfg` file.

   If you do not know the location of the log directory, check the `install.cfg` file. The `log.mantaslog.location` property indicates the log location; the default is `$KDD_PRODUCT_HOME/logs` but this location is configurable.
3. Do either of the following:
   - Type `rm job<job_id>-<date>-<time>.log` at the log directory prompt to clear one job log, where `<job_id>-<date>-<time>` is the name of a specific job log.
   - Type `rm job*` to clear all job logs.
4. Restart the `dispatcher`.

4.5 Recovering Jobs from a System Crash

If the system crashes, all active jobs (`status_cd = RUN`) fail. You can recover the jobs by running the script `recover_mantas.sh`. This script changes the status_cd to RES so that these jobs can restart and finish running. The `recover_mantas.sh` script has an optional parameter—the date on which the system ran the `start_mantas.sh` script. This parameter has a `DD-MM-YYYY` format. The default value is the current date.

Running the `recover_mantas.sh` script with this parameter ensures the script recovers only the jobs started that day. The dispatcher must be running to pick up the restarted jobs. This results in either a successful completion (`status_cd = FIN`) or failure (`status_cd = ERR`).

You can restart jobs that ended in failure by running the `restart_mantas.sh` script. The `restart_mantas.sh <template group id>` script changes the status_cd from ERR to RES for any jobs passed in the template group that have a status_cd of ERR for the `dispatcher` to pickup.

4.6 Executing Batches Through the OFSAAI User Interface

System Administrator users can run Behavior Detection jobs and Post Processing jobs from the OFSAAI UI. Activities can be performed through a batch process that can be executed once a year or periodically such as Daily, Weekly, Monthly, Quarterly, and Half-yearly depending on a firm's requirement.

> **NOTE** For the batches to start, iccserver, router, AM and message server must be started in the same sequence as mentioned. For more information on starting servers, refer to the Oracle Financial Services Advanced Analytical Applications Infrastructure (OFS AAAI) Applications Pack Installation and Configuration Guide.

This section includes the following topics:

- Adding Behavior Detection Batches
- Adding Tasks to a TBAML Batch
- Setting Task Precedence
- Running a Single Task Using a Batch
- Scheduling a Batch Once
- Scheduling a Daily Batch
- Scheduling a Weekly Batch
- Configuring a Monthly Batch
- Monitoring a Batch After Execution
- Canceling a Batch After Execution
- Re-starting a Batch
- Re-running a Batch
- Managing the Batch Processing Report

> **NOTE** Available cursors in database should be set to a minimum of 1000. Before restarting the Webserver, dispatcher should be ended.

4.6.1 Adding Behavior Detection Batches

To add a batch, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.

3. In the Navigation List, select **Operations**, then **Batch Maintenance**. The Batch Maintenance page is displayed.

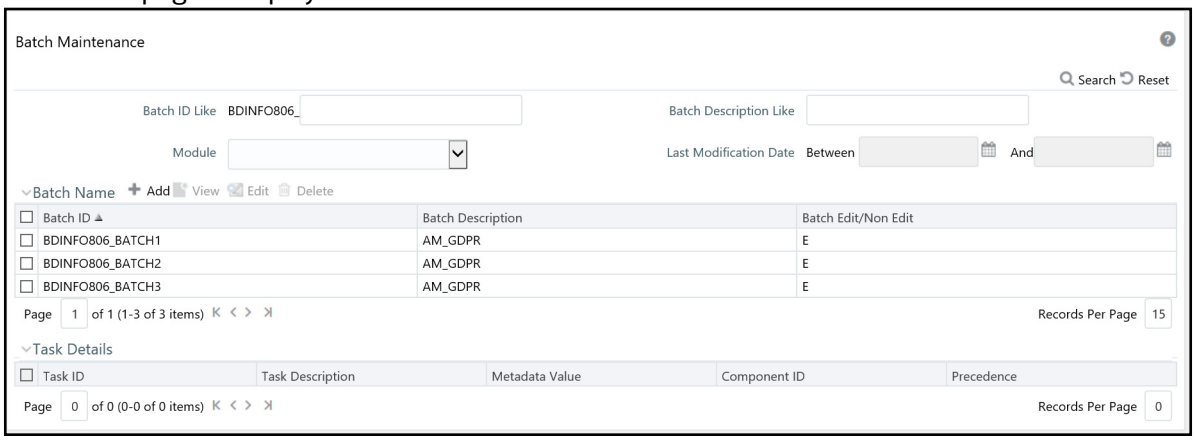


Figure 10: Batch Maintenance Page

4. In the Batch Name section, click **Add**. The Add Batch Definition page is displayed.

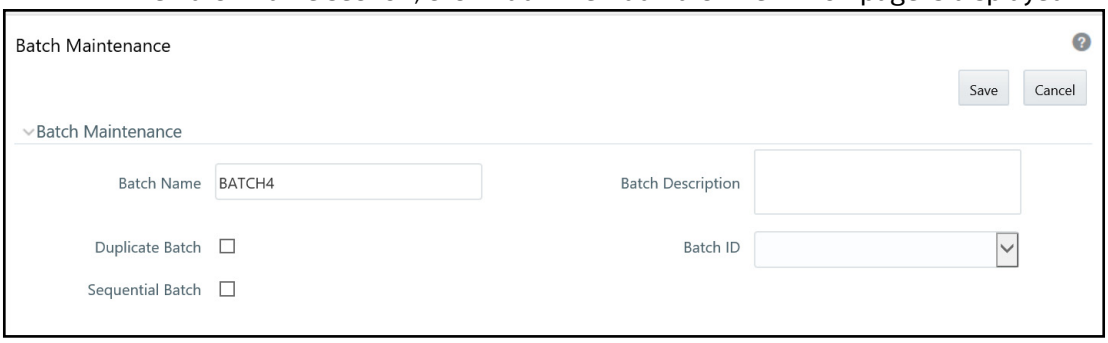


Figure 11: Add Batch Definition page

5. Enter the batch details as described in the following table:

Table 15: New Batch Details

Field	Description
Batch Name	Enter the name for the new batch.
Batch Description	Enter a description for this batch.
Duplicate Batch	Select this check box if the batch is a duplicate batch.
Sequential Batch	Select this check box if the batch must be run sequentially to another batch.
Batch ID	The Batch ID will be auto-populated.

6. Click **Save**. The added batch appears in the Batch Name section of the Batch Maintenance page.

4.6.2 Setting Up Ingestion through AAI

Ingestion through AAI can be achieved by calling the customized shell scripts from the OFSAA Framework Batch Operations Module. The following scripts can be customized through OFSAAI:

- `set_mantas_date.sh`
- `start_mantas_batch.sh`

- `execute.sh`
- `end_mantas_batch.sh`

The custom shell script must be kept under `<FIC_HOME>/ficdb/bin` and associated to an OFSAAI Data Transformation (DT).

The following Custom shell scripts are present in `<FIC_HOME>ficdb/bin`, which can be used directly in OFSAAI Data Transformation (DT).

- `SetMantasDate.sh`
- `StartMantasBatch.sh`
- `EndMantasBatch.sh`

For more information about OFSAAI Data Transformation (DT), see the *Post Load Changes* section in the Oracle Financial Services Analytical Applications Infrastructure User Guide.

Similarly, you must create custom shell scripts for execute.sh and associate them to an OFSAAI Data Transformation (DT).

4.6.3 Adding Tasks to a TBAML Batch

To add tasks to an existing batch or newly created batch definition, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Maintenance**. The Batch Maintenance page is displayed.

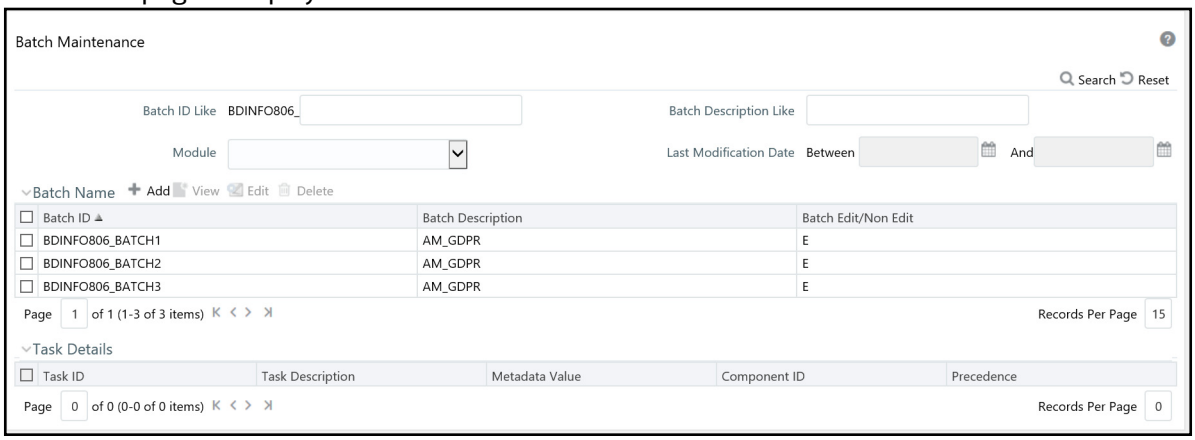


Figure 12: Batch Maintenance Page

For further instructions on how to add a new batch or add tasks to an existing batch, see the *Batch Maintenance* section in the *Oracle Financial Services Advanced Analytical Applications Infrastructure (OFSAAAI) User Guide*.

4.6.4 Setting Task Precedence

After you have created a task, you must indicate which tasks must be executed prior to the newly created task in a batch.

To set task precedence, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.

2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Maintenance**. The Batch Maintenance page is displayed.

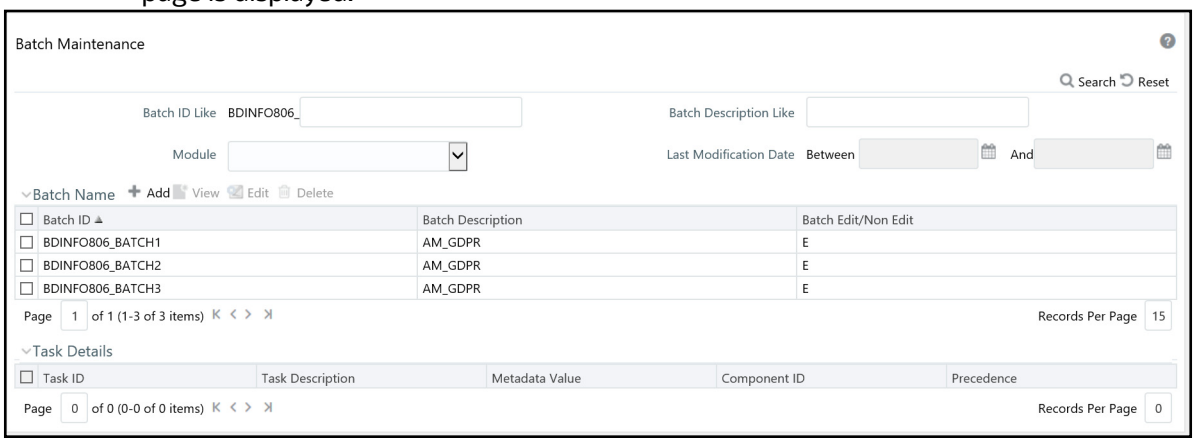


Figure 13: Batch Maintenance page

4. In the Batch Name section, select the batch that you want to set task precedence for.
5. In the Task Details section, click . The Task Precedence Mapping window is displayed.

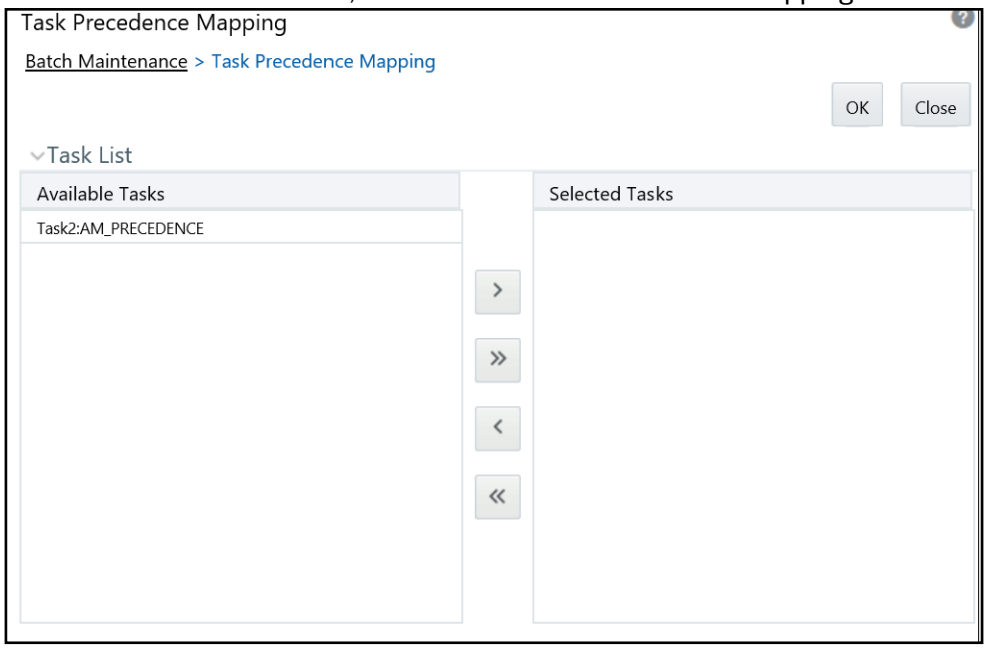


Figure 14: Task Precedence Mapping

6. Move the tasks which must be executed prior to this task from the Available Tasks pane to the Selected Tasks pane.
7. Click **OK** after you have selected all tasks which must precede the task. The selected tasks are listed in the Precedence column of the Task Details section.

4.6.5 Running a Single Task Using a Batch

From the Batch Execution page, you can also run a single task from a batch.

> **NOTE** Running a single task using a batch is not a recommended approach and should be done only for debugging a particular task.

To run a single task using a batch, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Execution**. The Batch Execution page is displayed.

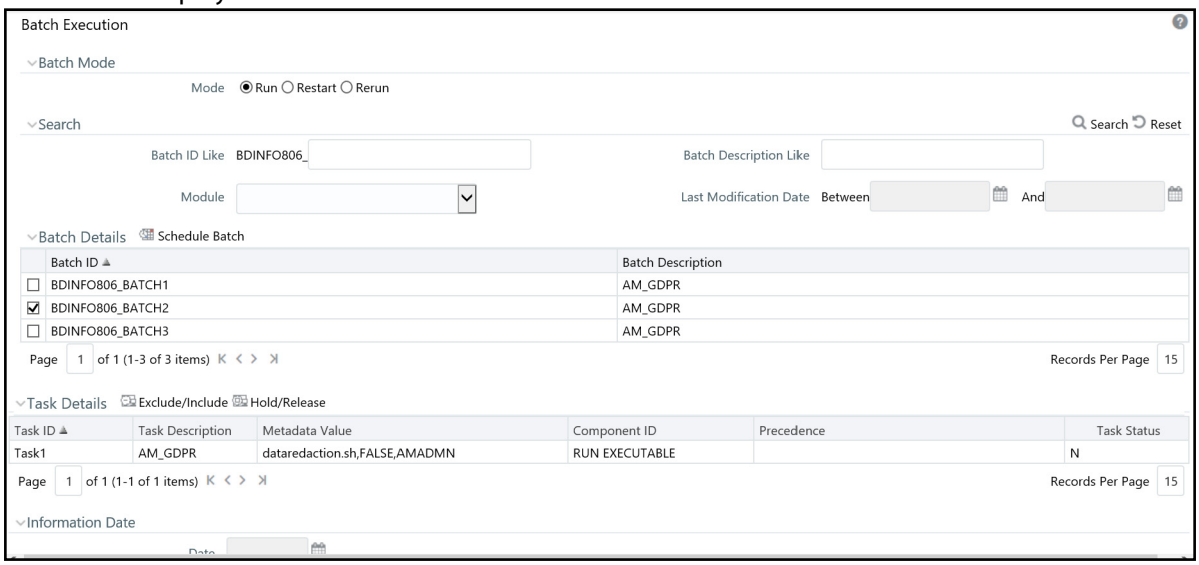


Figure 15: Batch Execution page

4. In the Batch Details section, select the particular batch that you want to execute.
5. In the Task Details section, click **Exclude/Include**. The Task Mapping window is displayed.

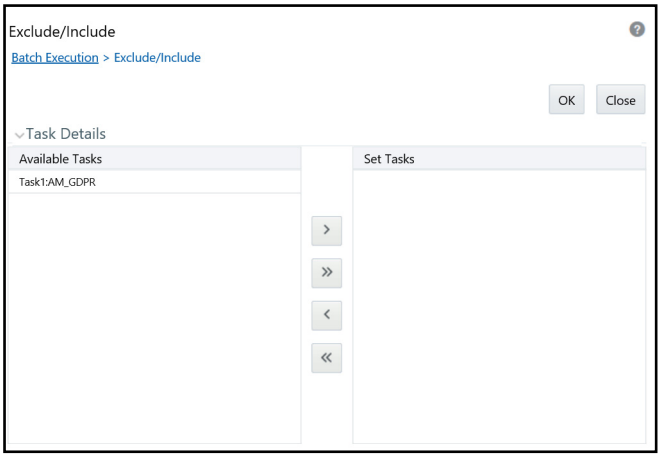


Figure 16: Task Mapping Window

6. Retain the tasks that you want to execute under Available Tasks section and move the rest to the Set Tasks section.
7. Click **OK**. The following warning message is displayed:

   If you exclude a task, it will be skipped when executing the batch but, the precedence will not be altered. Do you want to exclude the selected tasks)?
8. Click **OK**.
9. Click **Execute Batch**.

4.6.6 Scheduling a Batch Once

To schedule a batch that you want to run only once, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Scheduler**. The Batch Scheduler page is displayed.
4. Select a batch that you want to schedule from the list of available batches. The Batch Scheduler section is expanded and displays additional options.
5. Click **New Schedule**.
6. Set the frequency of the new schedule as **Once**.
7. Enter the schedule time of the batch by specifying the **Start Date** and the **Run Time**.

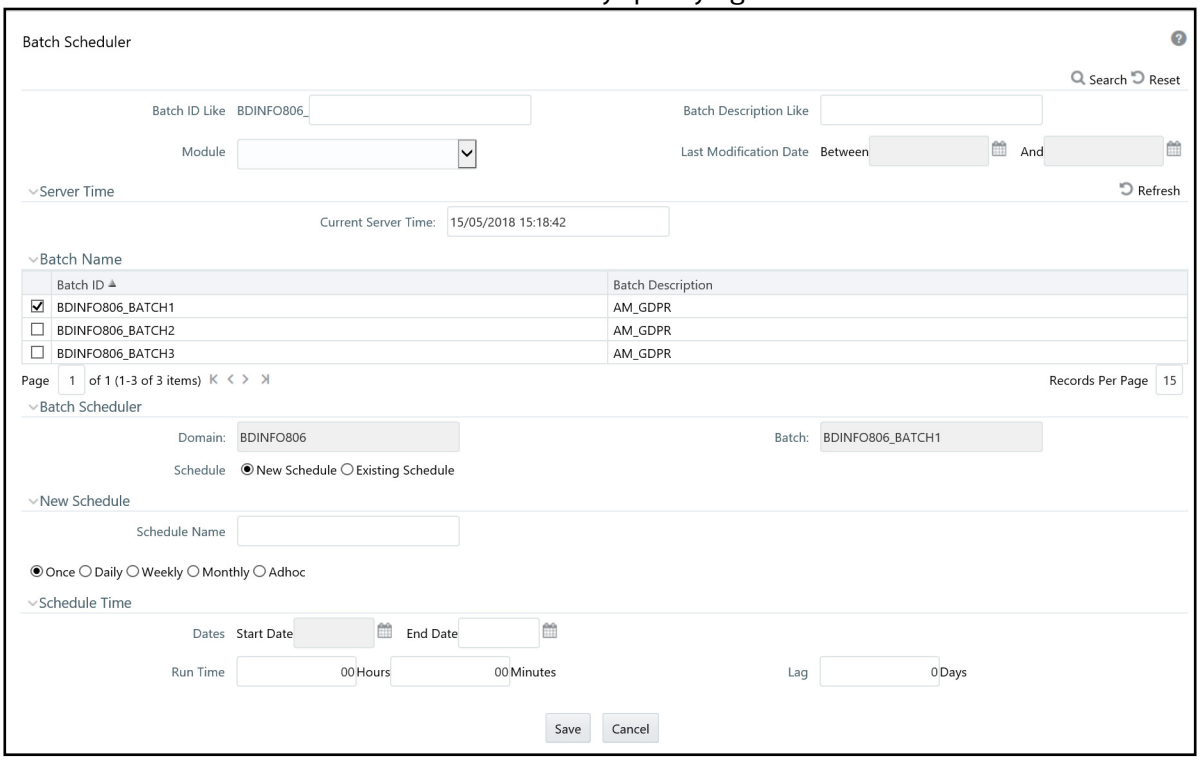


Figure 17: Scheduling a Batch Once

8. Click **Save**. The batch will run at the specified date and time.

4.6.7 Scheduling a Daily Batch

To schedule a batch that you want to run daily, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Scheduler**. The Batch Scheduler page is displayed.
4. Select a batch that you want to schedule from the list of available batches. The Batch Scheduler section is expanded and displays additional options.
5. Click **New Schedule**.
6. Set the frequency of the new schedule as **Daily**.
7. Enter the schedule time of the batch by specifying the **Dates**, **Run Time**, and **Every** information.

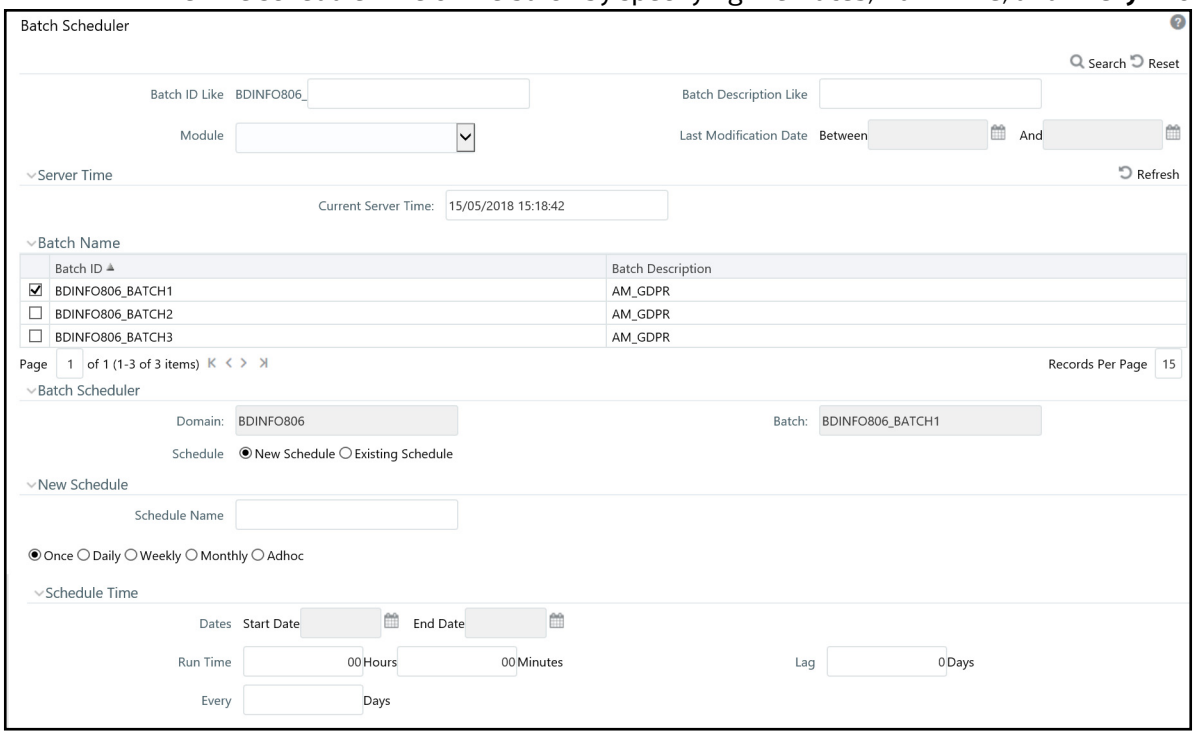


Figure 18: Scheduling a Daily Batch

8. Click **Save**. The batch will run at the specified date and time.

4.6.8 Scheduling a Weekly Batch

To schedule a batch that you want to run weekly, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Scheduler**. The Batch Scheduler page is displayed.
4. Select a batch that you want to schedule from the list of available batches. The Batch Scheduler section is expanded and displays additional options.

5. Click **New Schedule**.
6. Set the frequency of the new schedule as **Weekly**.
7. Enter the schedule time of the batch by specifying the **Dates**, **Run Time**, **Every, Working days of the Week** information.

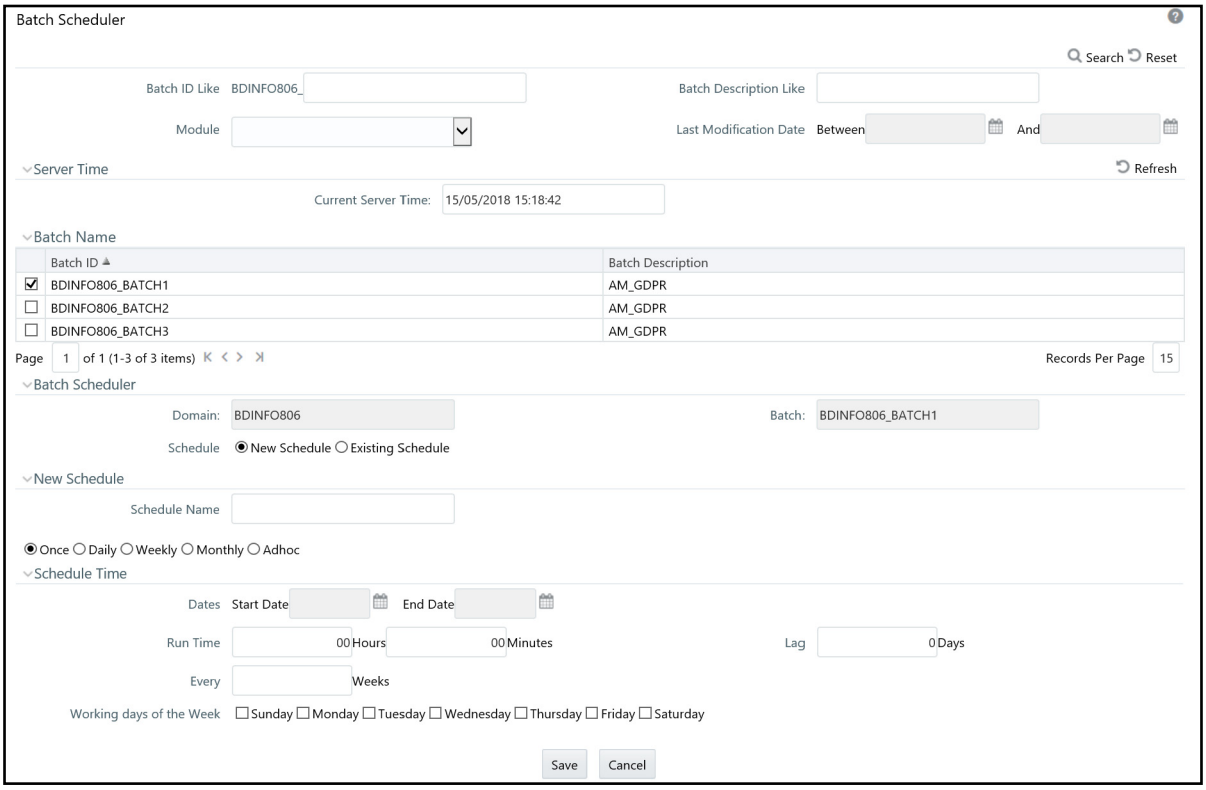


Figure 19: Scheduling a Weekly Batch

8. Click **Save**. The batch will run at the specified date and time.

4.6.9 Configuring a Monthly Batch

To schedule a batch that you want to run monthly, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Scheduler**. The Batch Scheduler page is displayed.
4. Select a batch that you want to schedule from the list of available batches. The Batch Scheduler section is expanded and displays additional options.
5. Click **New Schedule**.
6. Set the frequency of the new schedule as **Monthly**.

7. Enter the schedule time of the batch by specifying the **Dates**, and **Run Time** information.

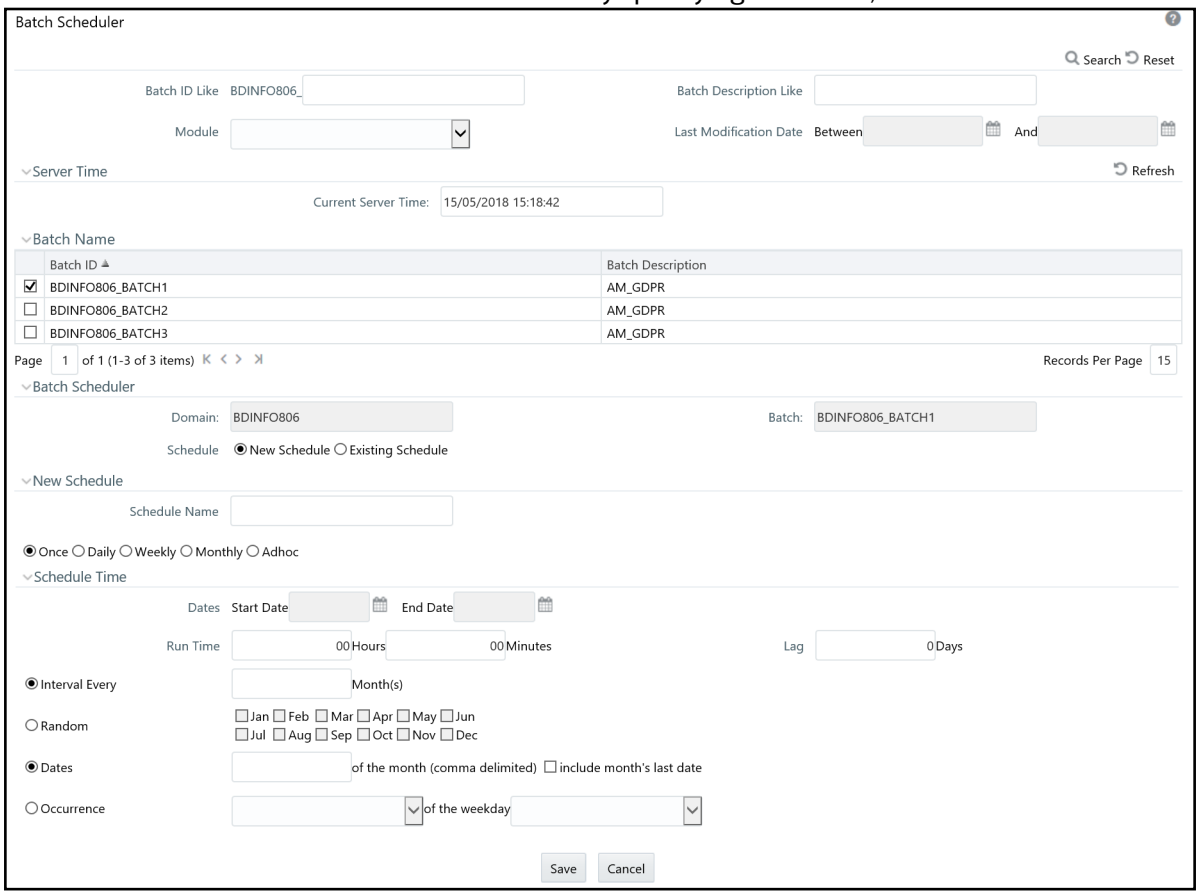


Figure 20: Configuring a Monthly Batch

8. Click **Save**. The batch will run at the specified date and time.

4.6.10 Monitoring a Batch After Execution

Monitoring a batch helps you track the status of execution of an individual task that was included in the batch. Through monitoring, you can also track the batch status which in turn helps you in debugging.

To monitor a batch after it is executed, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then **Batch Monitor**. The Batch Monitor page is displayed.

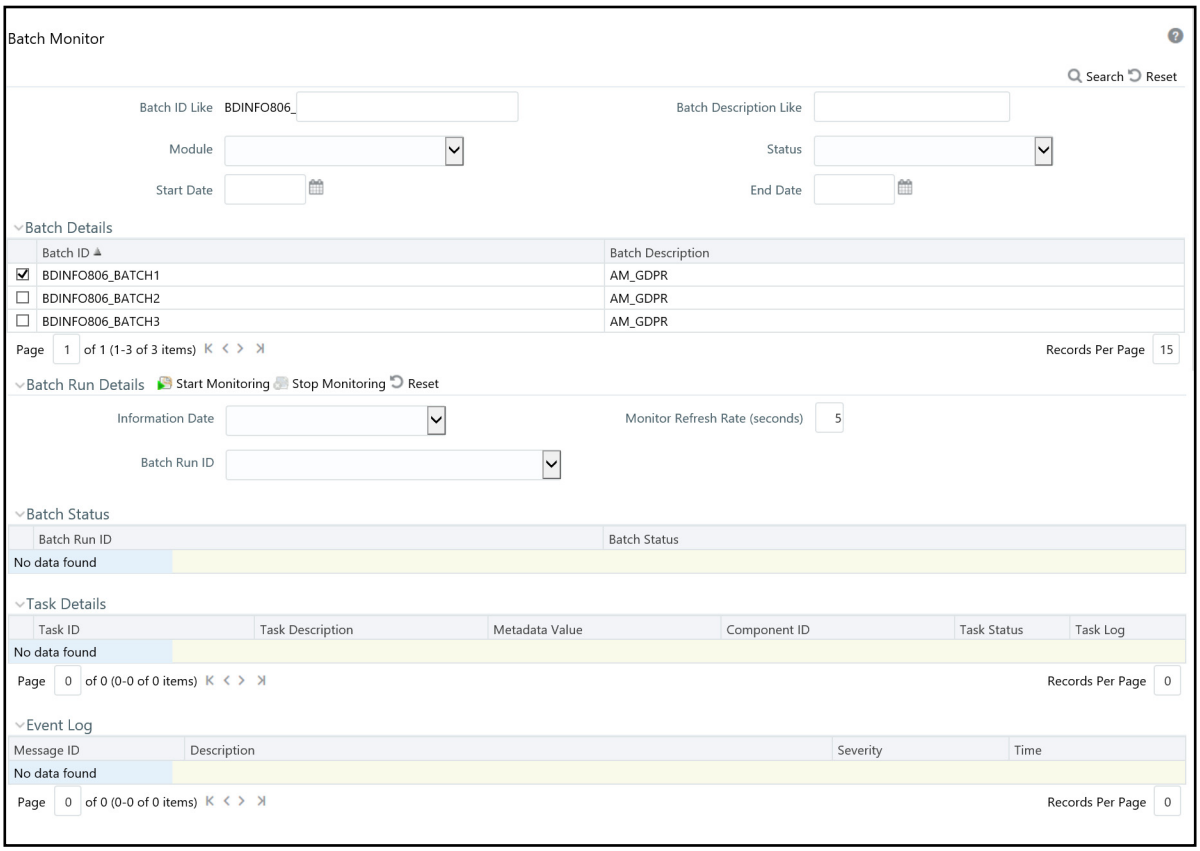


Figure 21: Batch Monitor Page

4. Select a batch from the Batch Details lists that you want to monitor.
5. From Batch Run Details section, select an Information Date and the Batch Run ID from the drop-down list.
6. Click **Start Monitoring** to start the monitoring. The Batch Status, Task Details, and Event Log sections are populated with information about this batch's execution.

4.6.11 Canceling a Batch After Execution

Cancellation of a batch cancels a current batch execution.

> **ATTENTION** This is not recommended and should be done only when the batch was fired accidentally or when a particular is taking too long to execute.

To cancel a batch after it is executed, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Batch Cancellation**. The Batch Cancellation page is displayed.

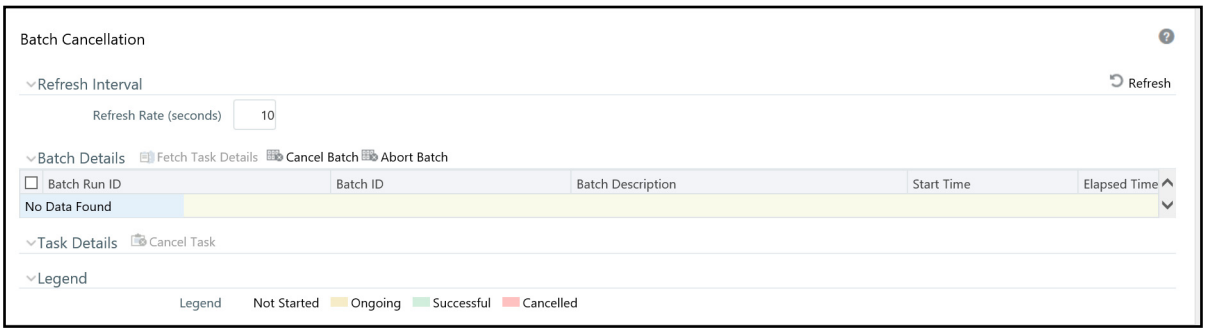


Figure 22: Batch Cancellation Page

4. Under the Batch Details section, select the batch whose execution you want to cancel.
5. Click **Cancel Batch**.

4.6.12 Re-starting a Batch

You can restart a batch execution when they have fail in their execution. When you restart a batch, it starts from the task at which it had failed. This happens when the failed task issue is debugged and resolved.

> **TIP** It is recommended that you debug and resolve a failed task before restarting the batch execution.

To restart a batch execution, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Batch Execution**. The Batch Execution page is displayed.
4. Select **Restart** from the Batch Mode section.

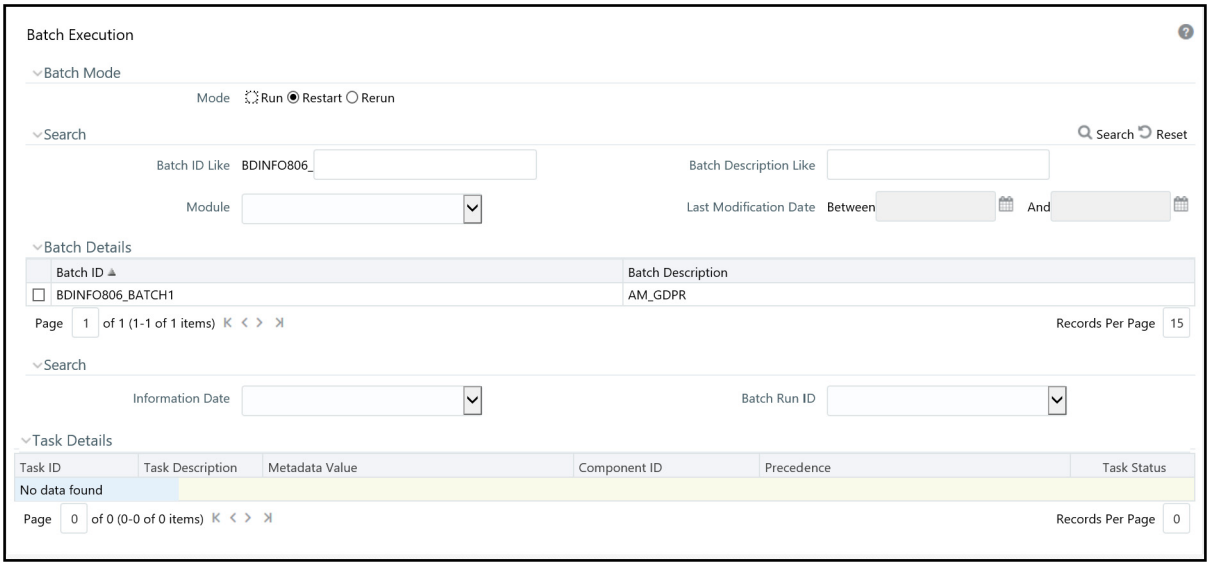


Figure 23: Re-starting a Batch

5. Select the batch from the Batch Details section that you want to restart.
6. Select the Information Date and Batch Run ID for the selected batch from the drop-down list.
7. Click **Execute Batch**.

4.6.13 Re-running a Batch

You can rerun a batch execution when you want all the tasks from a successful batch execution to be executed again from the beginning. When a successfully executed batch is rerun, a different Batch Run ID is created for each instance for the same Information Date.

> **NOTE** Creation of different Batch Run ID for each rerun of a batch is optional depending upon a firm's requirement.

To rerun a batch, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Batch Execution**. The Batch Execution page is displayed.
4. Select **Rerun** from the Batch Mode section.

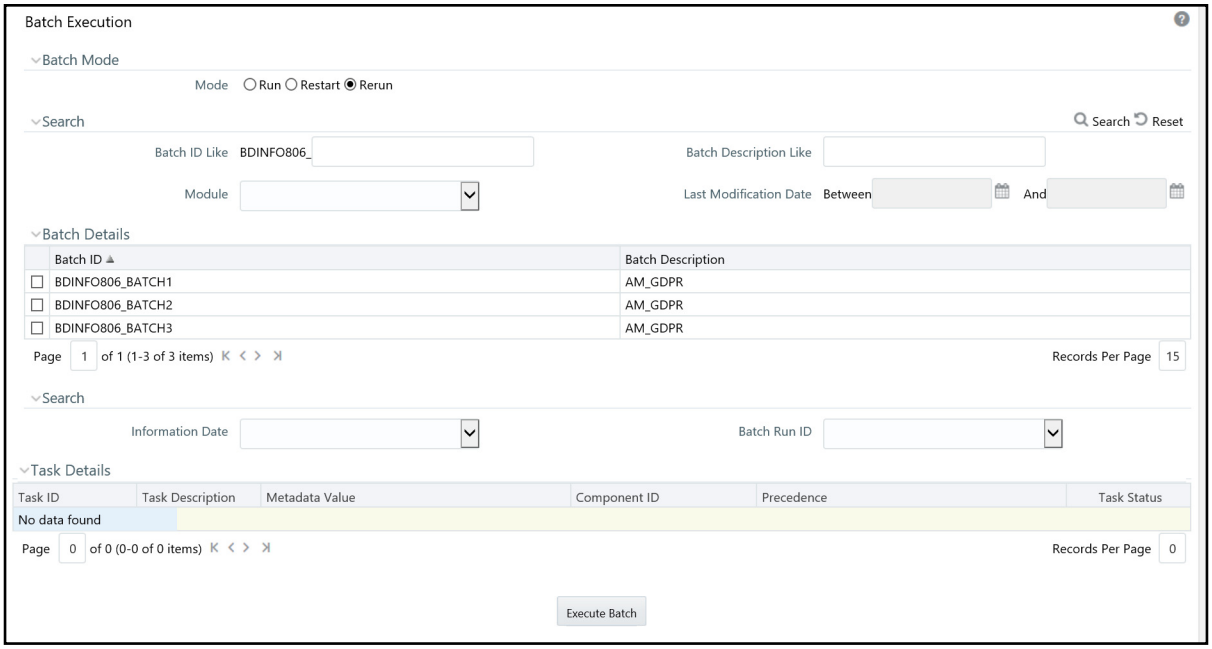


Figure 24: Re-running a Batch

5. Select the batch from the Batch Details section that you want to rerun.
6. Select the Information Date and Batch Run ID for the selected batch from the drop-down list.
7. Click **Execute Batch**.

4.6.14 Managing the Batch Processing Report

The Batch Processing Report allows you to view parameter details for batches in the following statuses:

- Not Started
- Ongoing
- Complete
- Failed
- Canceled

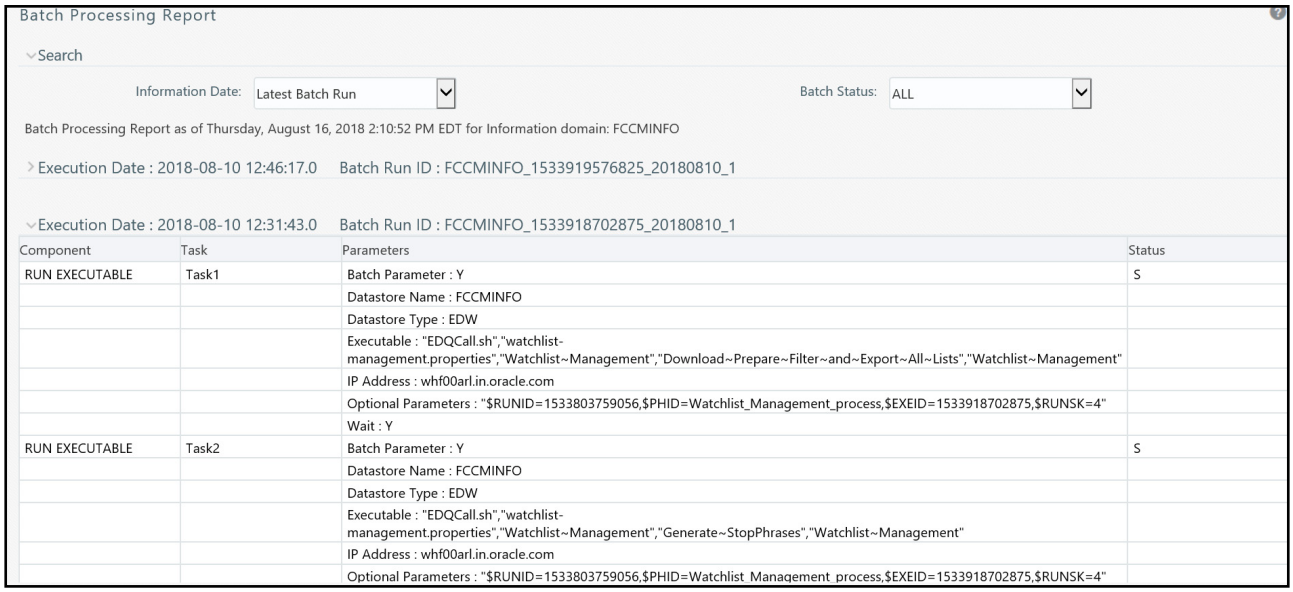


Figure 25: Batch Processing Report

To view a report, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Processing Report**.
4. In the Search bar, select the **Information Date** and **Batch Status** from the drop-down lists. All applicable reports will be listed by Execution Date and Batch Run ID.
5. Click to expand the report you wish to view details for.

4.6.15 Managing the View Log

The View Log allows you to view the following details for batches:

- Component
- Task Name
- Task ID
- Batch Start Date
- Batch End Date

- Batch Status
- Elapsed Time
- User who ordered the batch

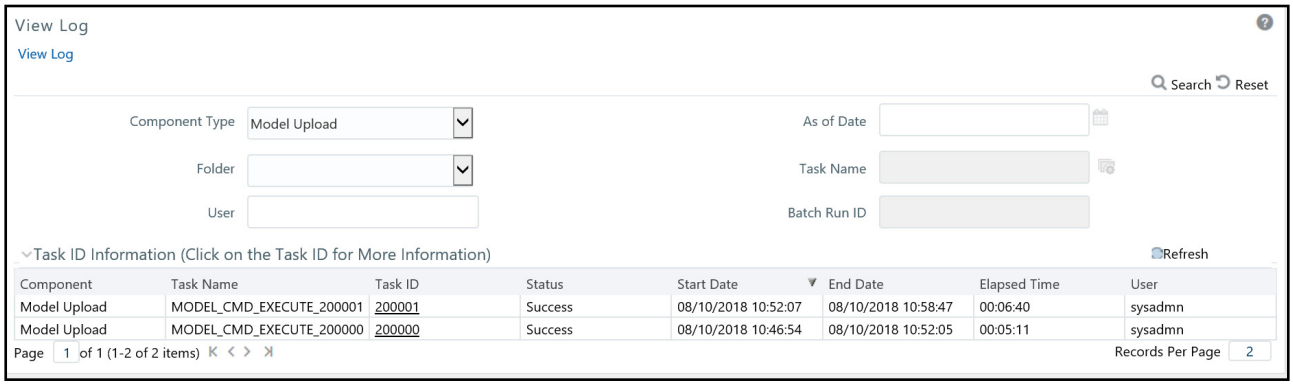


Figure 26: View Log

To view a report, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **View Log**.
4. In the Search bar, enter the search criteria for the log you wish to view. The Task ID Information section displays the details.

4.7 Executing Batches Through the Run Rules Framework Interface

System Administrator users can also run jobs from the Run Rules Framework. The following sections describe this process.

4.7.1 Starting a Batch Run

> **NOTE** For executing a batch, you cannot start two batches simultaneously for same processing group.

This section explains how to start the batch run.

To start the batch run, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Run Rule Framework**.
4. Click **Run**. The Run page is displayed with the available applications.

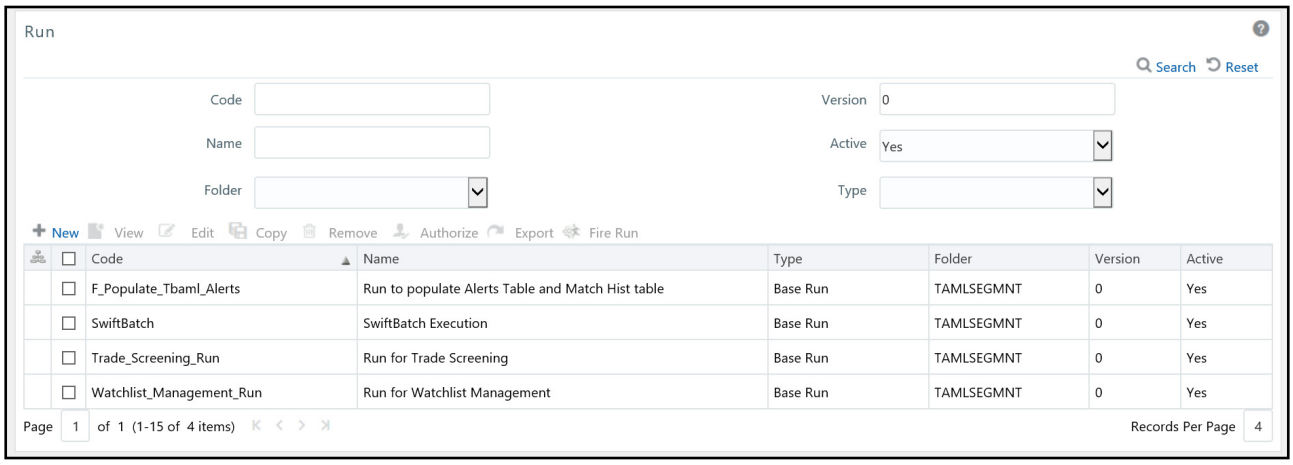


Figure 27: Run page

5. Select an application from the Code column (for example, F_Populate_Tbaml_Alerts) and click **Edit**. The Run Definition page is displayed with the process or processes associated to this application.

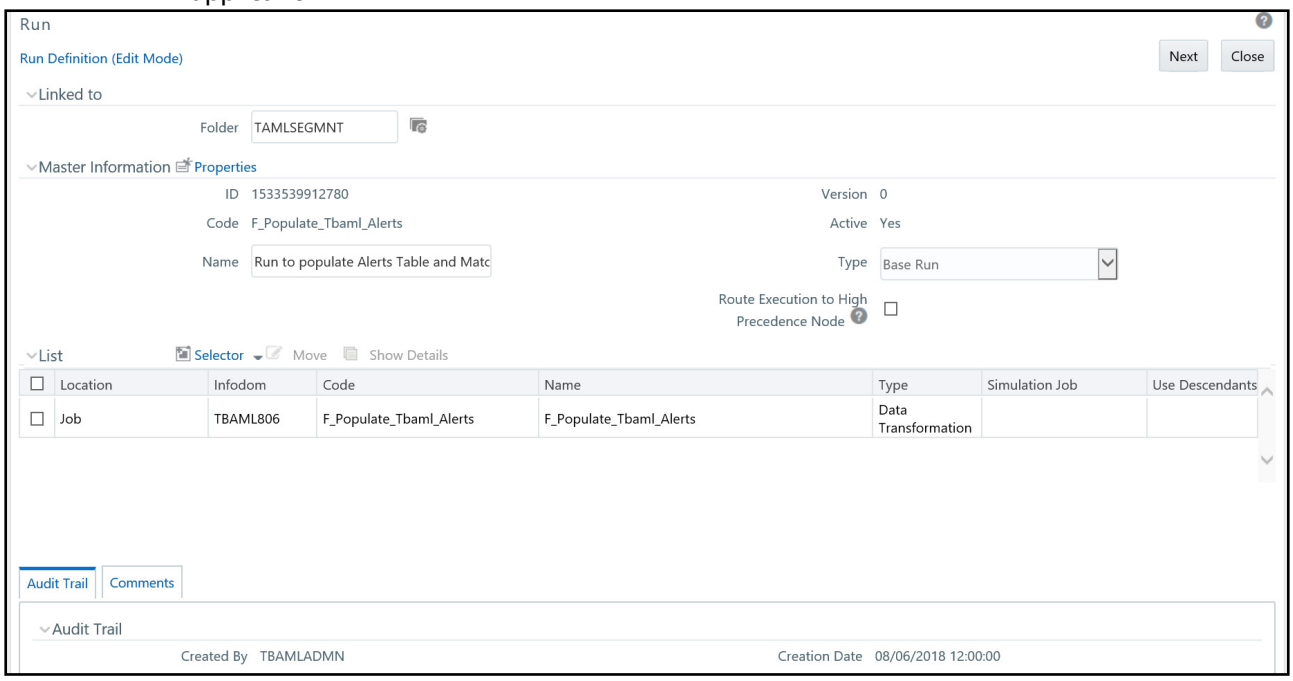


Figure 28: Run Definition page

6. Select the job or jobs for the batch and click **Next**. The Detail section is populated. Complete your edits and click **Save**.
7. In the Navigation List, select **Process** to open the Process page.

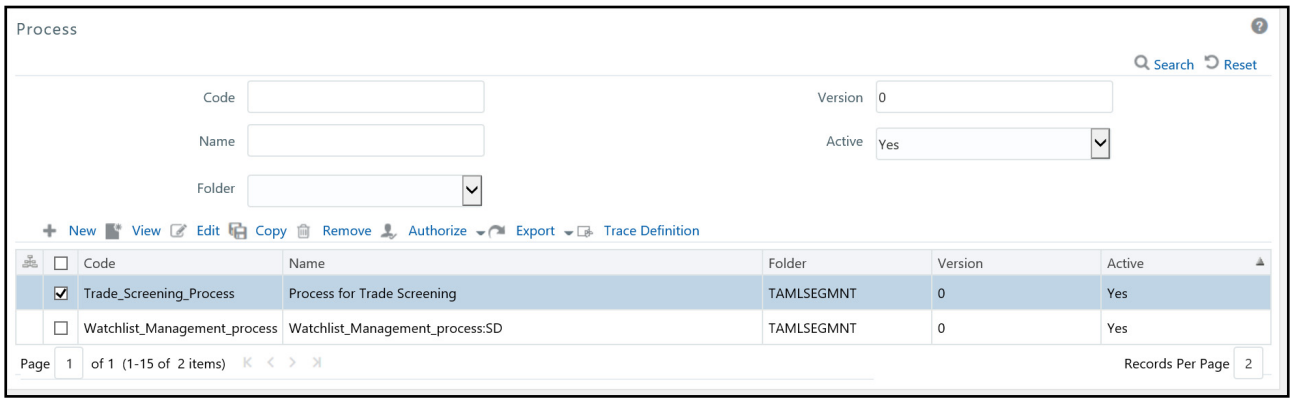


Figure 29: Process page

8. Select an application from the Code column and click **Edit**. The Process Definition page is displayed with the process or processes associated to this application.

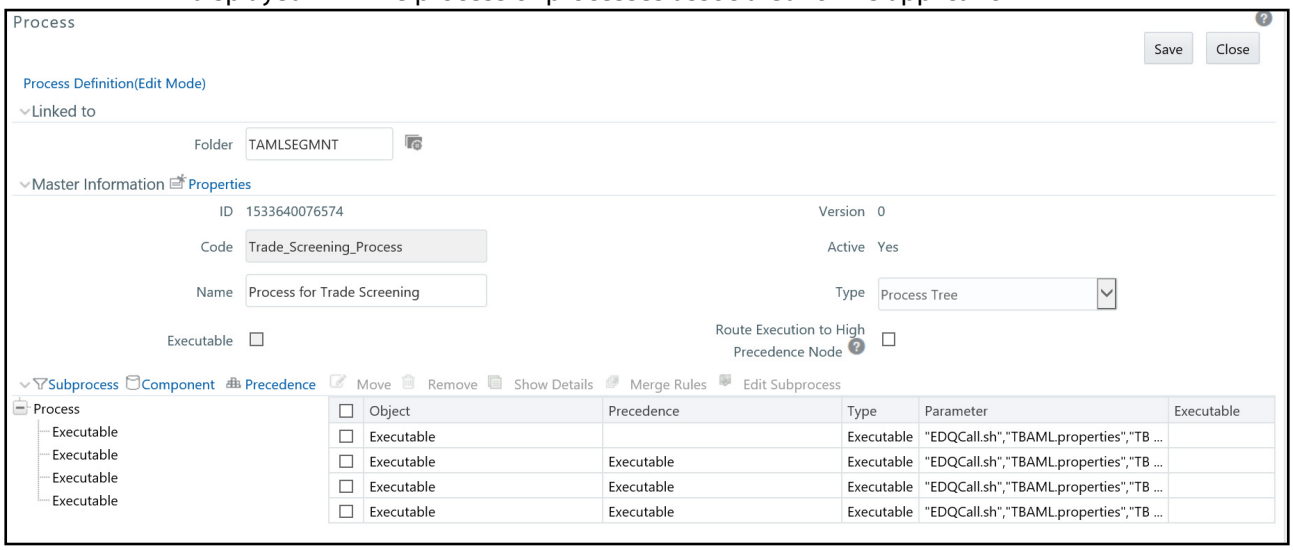


Figure 30: Process Definition page

9. Click **Component**. The Component Selector window is displayed.

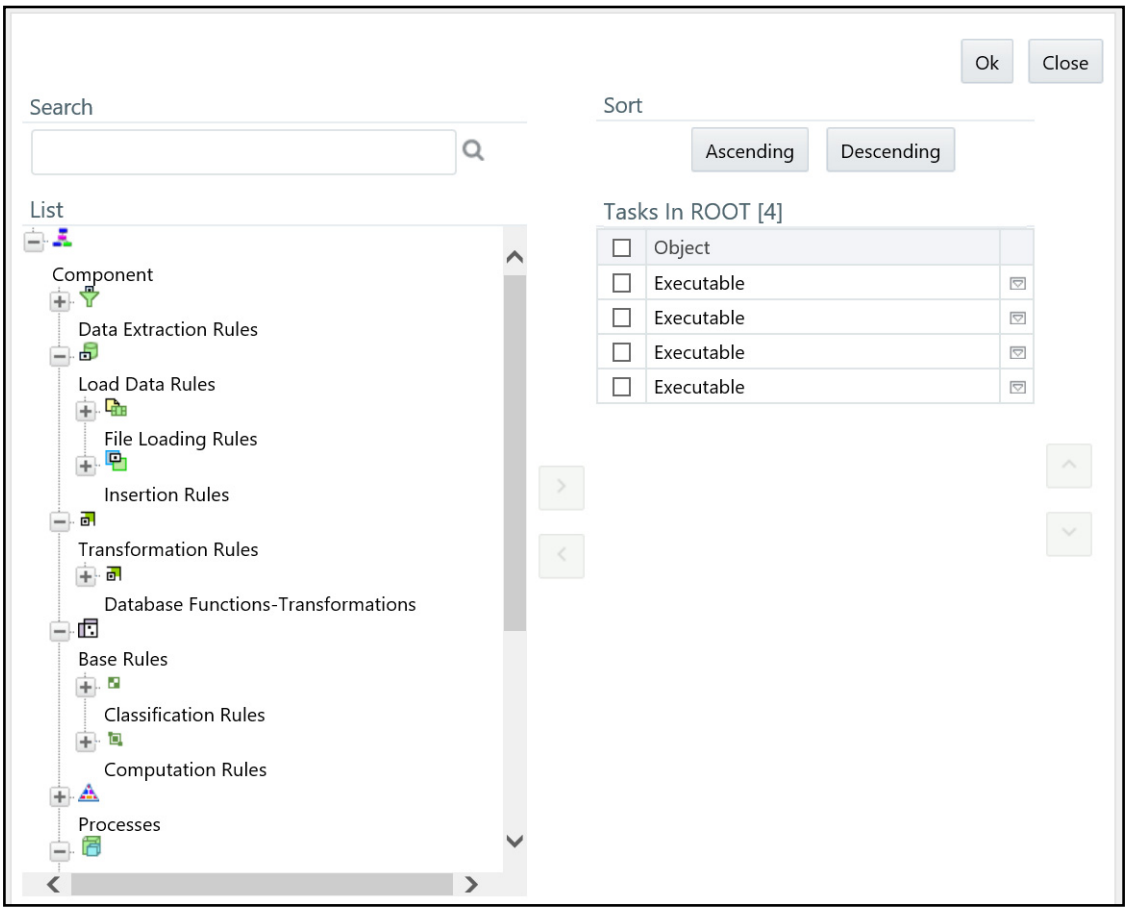


Figure 31: Component Selector page

The following are default parameters:

```
"MAN","","ALL","START","DLY"
```

- MAN: is group name. Modify the name of group as mentioned in FCC_PROCESSING_GROUP table. For example, E2E BATCH ALL SOURCE
- "" Source Batch for Correlation
- ALL: is component that can be modified if required
- START: is used to start the batch
- DLY: is Data Origin

The following is an example of a parameter

```
"E2E BATCH ALL SOURCE","","ALL","START","IND"
```

10. Modify the parameters and click **OK.**

4.7.2 Ending a Batch Run

To end the batch run, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.

3. In the Navigation List, select **Operations**, then click **Run Rule Framework**.
4. Click **Process**. The Process page is displayed with the available applications.

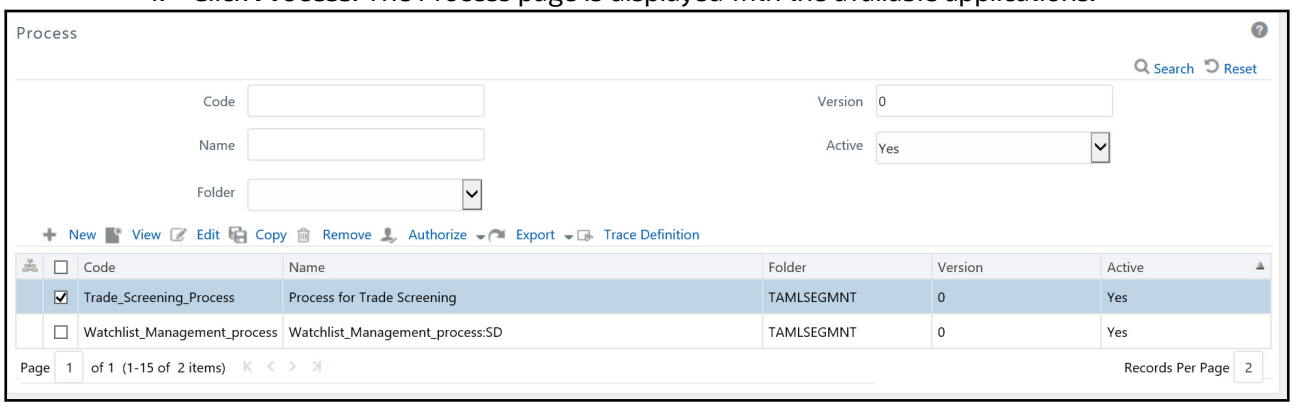


Figure 32: Process page

5. Select an application from the Code column and click **Edit**. The Process Definition page is displayed with the process or processes associated to this application.
6. Select an End Batch, for example BD_TBAML_End_E2E.
7. Click **Edit**. The Process Definition page is displayed.
8. Click **Component**. The Component Selector window is displayed.

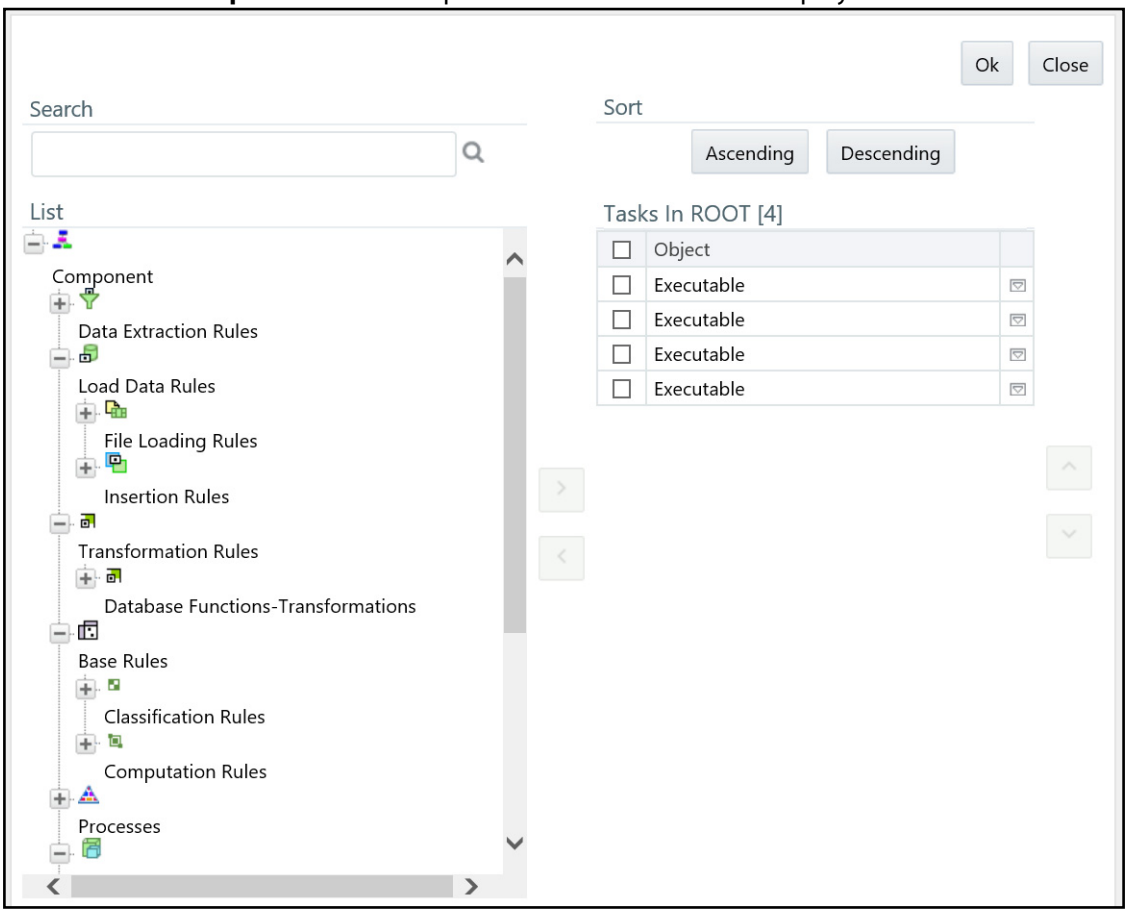


Figure 33: Component Selector page

9. Click **Parameters** option. The Parameters window is displayed. The following are default parameters:

```
"","","ALL","END",""
```

- Source Batch for Correlation
- ALL: is the component. Modify the component if required.
- END: is used to end the batch.

10. Modify the parameters and click **OK**.

4.7.3 Executing a Batch Run

This section explains how to execute the batch run.

To access and execute the batch run, follow these steps:

1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.
3. In the Navigation List, select **Operations**, then click **Run Rule Framework**.
4. Click **Run**. The Run page is displayed with the available batch jobs.

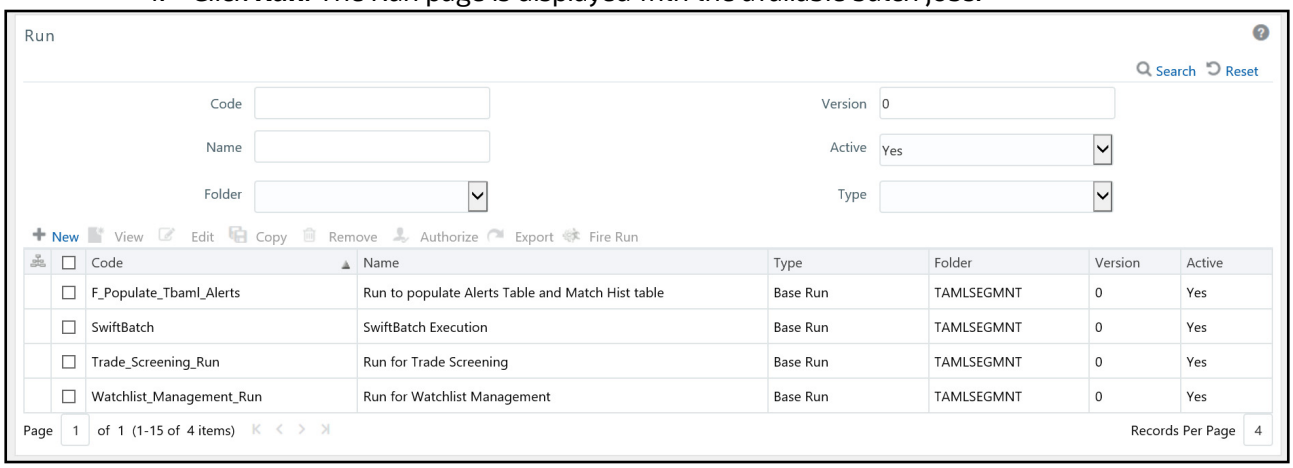


Figure 34: Run page

5. Select the batch job that is to be executed and click **Fire Run**. The Fire Run window is displayed.

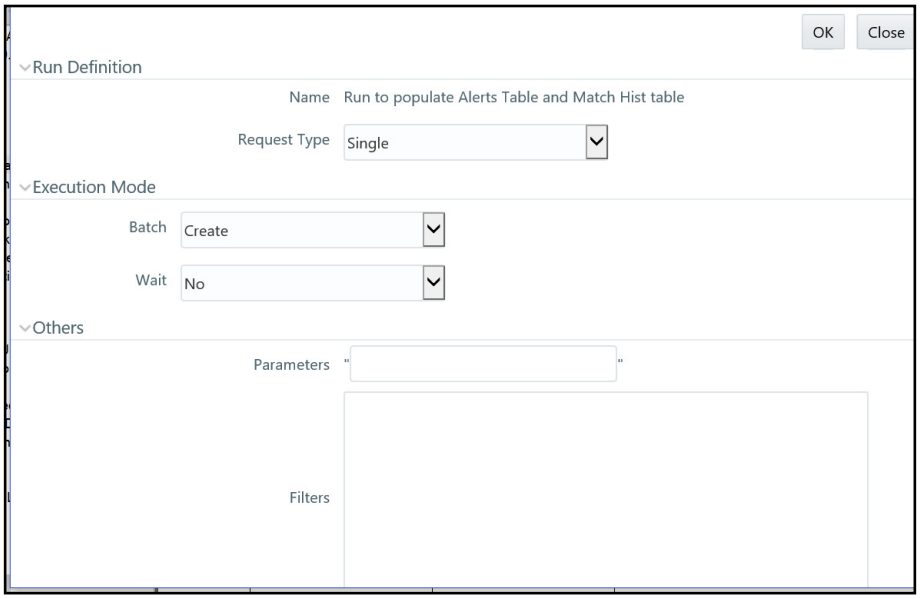


Figure 35: Fire Run

6. Enter the following details:

Table 16: Adding Fire Run Details

Field	Description
Request Type	Select Request Type based on the following options: • Single: If the batch must be executed once. • Multiple: If the batch must be executed multiple times at different intervals.
Batch	Select Batch. It has the following options: • Create • Create & Execute From these options, select Create & Execute.
Wait	Select Wait. It has the following options: • Yes: This executes the batch after a certain duration. Enter the duration as required. • No: This executes the batch immediately.
Parameters	Enter the parameters for this batch.
Filters	Enter the filter details. Note: $MISDATE option can be used to execute the run for that particular day. The format for it to enter in the filter details is: to_date(<ACTIVITY_TABLE_NAME>.<ACTIVITY_DT_COL>)=$MISDATE Note: For $MISDATE option: • For either Date or Timestamp datatypes, to_date is mandatory for the filter. • Activity Table Name and Activity Column Name should be in capital.

7. Click **OK** to run the batch. The following message is displayed: *Batch Execution is in progress*.

> **NOTE** If batch execution fails, then see the batch details in Batch Monitor. For more information on Batch Monitor, see the Oracle Financial Services Analytical Applications Infrastructure User Guide.

5 Post-Processing Tasks

This chapter defines the following post-processing administrative tasks:

- About Post-Processing
- Match Scoring
- Alert Creation
- Alert Scoring
- Highlight Generation
- Historical Data Copy

5.1 About Post-Processing

During post-processing of ingested data, TBAML prepares the detection results for presentation to users. Preparation of the results depends upon the following processes:

- Match Scoring**:** Computes a ranking for scenario matches indicating a degree of risk associated with the detected event or behavior (Refer to Match Scoring for more information).
- Alert Creation**:** Packages the scenario matches as units of work (that is, events), potentially grouping similar matches together, for disposition by end users (Refer to Alert Creation for more information).
- Alert Scoring**:** Ranks the events (including each match within the events) to indicate the degree of risk associated with the detected event or behavior (Refer to Alert Scoring for more information).
- Highlight Generation**:** Generates highlights for events that appear in the event list in the Behavior Detection subsystem and stores them in the database (Refer to Highlight Generation for more information).
- Historical Data Copy**:** Identifies the records against which the current batch's scenario runs generated events and copies them to archive tables (Refer to Historical Data Copy for more information).

> **NOTE** You can re-run any failed post-processing job.

5.1.1 Order of Running Post-Processing Administrative Tasks

Run the post-processing administrative tasks in this order:

8. Match Scoring (501)
9. Single Match Alert Creation (503)
10. Alert Scoring (504)
11. Highlight Generation
12. Historical Data Copy
13. Alert Correlation (508)

For all the post processing jobs MANTAS batch should be up and running.

5.2 Match Scoring

Behavior Detection provides a mechanism to compute a score for matches to provide an initial prioritization. Match Scoring rules are created using the Scoring Editor from the Administration Tools. Refer to the Administration Tools User Guide for more information.

5.2.1 Running the Match Scoring Job

The Match Scoring job is part of the Behavior Detection subsystem. Behavior Detection delivers job template group 501 to run the Match Scoring job.

To run the Match Scoring job, follow these steps:

1. Verify that the dispatcher is running.
2. Run the `start_mantas.sh <template id>` script as follows:

   ```
   start_mantas.sh 501
   ```

All new matches in the system are scored.

5.3 Alert Creation

Matches are converted into events with the Alert Creator processes. These processes are part of the Behavior Detection subsystem.

The system uses two types of Alert Creator jobs:

- Multi-match Alert Creator: Generates events for matches that share a common focus, are from scenarios in the same scenario group, and possibly share other common attributes. Each focus type has a separate job template.
- Single-match Alert Creator: Generates one event per match.

> **NOTE** The `KDD_JRSDCN` table is empty after system initialization and requires populating before the system can operate. If a new jurisdiction is to be added, it should be added to `KDD_JRSDCN` table.

5.3.1 Running the Alert Creation Job

The Alert Creator is part of the Behavior Detection subsystem. Behavior Detection provides default job templates and job template groups for running Alert Creator. These jobs can be modified using Administration Tools. Refer to the Administration Tools User Guide, for more information.

The following section describes running Alert Creator.

5.3.1.1 To Run Single Match Alert Creator

To run the single match Alert Creator, follow these steps:

1. Verify that the dispatcher is running.
2. Run the `start_mantas.sh` script as follows:

   ```
   start_mantas.sh 503
   ```

   where `503` is the job template that Behavior Detection provides to run the Alert Creator algorithm.

5.3.2 Understanding Advanced Alert Creator Configuration

The Alert Creator algorithm can support grouping strategies that the Administration Tools do not support. To use these advanced strategies, you must enter Alert Creator rules directly into the database. The following section discusses these advanced rules.

5.3.2.1 Advanced Rules

The executable retrieves new, unowned single matches generated from specified types of scenarios. It then groups them based on one of four implemented algorithms and a specified list of bindings for grouping. It requires parameter settings to designate:

- Choice of grouping algorithm to use.
- Scenario types associated with the set of matches to consider for grouping.
- Bindings on which to base break group compatibility.

5.3.2.1.1 Grouping Algorithms

When grouping algorithms, choose from the following:

- **BIND_MATCH:** The Alert Creation module creates events based on matches with matching bindings/values based on a provided list of bindings to use when determining *groupability*.
- **BIND_BEHAVIOR_SCENARIO_CLASS:** The Alert Creation module creates events based on matches with matching scenario group code and with matching bindings/values based on a provided list of bindings to use when determining *groupability*.
- B**IND_BEHAVIOR_SCENARIO:** The Alert Creation module creates events based on matches with matching scenario ID and with matching bindings/values based on a provided list of bindings to use when determining *groupability*.
- **BIND_BEHAVIOR_PATTERN:** The Alert Creation module creates events based on matches with matching pattern ID and with matching bindings/values based on a provided list of bindings to use when determining *groupability*.
- **SINGLE_ALERT_MATCH:** The Alert Creation module creates events for all remaining matches. A event is created for each of the remaining matches, as long as they bind one of the centricity names in the bindings string. This is the *catch all* algorithm that ensures that all matches that have a bound centricity value and a corresponding event is created.

For a `BIND_MATCH` grouping rule, the system compares bindings (`KDD_BREAK_BINDING`) values for matches to determine whether it can group matches together into an FCC TBAML event.

For example, the grouping algorithm interprets `!TRADER ?ASSOC_SCRTY` to create an FCC TBAML event; each break set to be grouped must have a `TRADER` binding in which the values for that binding must match and each must either have an `ASSOC_SCRTY` binding in which the values match OR each must be missing the `ASSOC_SCRTY` binding. Events that mentioned `ASSOC_SCRTY` could only be grouped with other events that mentioned `ASSOC_SCRTY`. Similarly, events that did not mention `ASSOC_SCRTY` could only be grouped with other events that did not mention `ASSOC_SCRTY`.

This list is order-dependent and at least one binding should be marked as required using an exclamation point (!) to prevent grouping of all miscellaneous matches into one big break. The order helps determine the centricity in the first binding name in the binding string. The centricity name is used to determine the event's centricity ID.

5.4 Alert Scoring

TBAML provides a mechanism to compute a score for events to provide an initial prioritization. The score is an integer and will be bounded by a configurable minimum and maximum value.

This module has two different strategies for computing the event's score. All strategies are based on the score of the event's matches. The strategies are:

- **Max Match Score:** The score of the event equals the event's highest scoring match.
- **Average Match Score:** The score of the event equals the average of its matches score.

Refer to the Administration Tools User Guide for more information.

5.4.1 Running the Alert Scoring Job

To run an Alert Scoring Job, follow the steps:

1. Verify that the dispatcher is running.
2. Run the `start_mantas.sh` script as follows:

   ```
   start_mantas.sh 504
   ```

   where, `504` is the job template that OFSBD provides to run the Alert Scoring algorithm.

5.5 Highlight Generation

The behavior detection subsystem displays event and match highlights in the UI. The system calculates and stores these highlights in the database as part of the batch cycle using the following shell script:

```
run_highlights.ksh
```

This script is part of the Database Tools that resides in the `<OFSAAI Installed Directory>/database/db_tools/bin` directory. This script attaches to the database using the user that the `utils.database.username` property identifies in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file. You run highlight generation after the creation of events and before the system ends the batch with the `end_mantas_batch.sh` script.

By default, Behavior Detection writes log messages for this script in the `<OFSAAI Installed Directory>/database/db_tools/logs/highlights.log` file.

5.6 Historical Data Copy

TBAML maintains records that are directly involved with detected behaviors in a set of archive, or ARC, tables. The Historical Data Copy (HDC) process identifies the records against which the current batch's scenario runs generated events and copies them to the ARC tables.

The `run_hdc.ksh` and `upd_kdd_review_fin.sh` must run upon completion of all detection and other event post-processing , such as scoring and assignment, but before the system ends the batch with the following shell script:

```
end_mantas_batch.sh
```

This script is part of the Database Tools that reside in the `<OFSAAI Installed Directory>/database/db_tools/bin` directory.

The `run_hdc.ksh` shell script manages the HDC process. This process connects to the database as the user that the `truncate.database.username` property identifies in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file. This property should identify the

Atomic Schema user, a user in the database with write access to tables in the Behavior Detection Atomic schema.

To improve performance, you can adjust two configurable parameters in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file.

Table 17: HDC Configurable Parameters

Parameter	Recommended Value	Descriptions
hdc.batchsize	10000	Number of break match key IDs are included in each batch thread for data retrieval.
hdc.maxthreads	2x (Number of CPUs)	Maximum number of concurrent threads that HDC uses for retrieving data to tune performance.

To run the Historical Data Copy (HDC) process, follow these steps.

1. Navigate to `<OFSAA installed directory>/database/db_tools/bin/execute run_hdcTBAML.ksh`

   By default, log messages for this script are written in the `<OFSAAI Installed Directory>/database/db_tools/logs/hdc.log` file.
2. Verify TBAML-related tables to check the HDC data copy

6 Managing Batch Processing Utilities

Oracle provides utilities that enable you to set up and modify a selection of batch-related database processes. The chapter focuses on the following topics:

6.1 About Batch Processing Utilities

TBAML database utilities enable you to configure and perform batch-related system pre-processing and post-processing activities.

- Managing Alert Purge Utility: Provides the capability to remove alerts (along with their matches and activities) generated erroneously or which have exceeded the retention policies of the organization.
- Managing Batch Control Utility: Manages the start and termination of a batch process (from data management to event post-processing) and enables access to the currently running batch.
- Managing Calendar Manager Utility.: Updates calendars in the TBAML system based on predefined business days, holidays, and days off or non-business days.
- Managing Data Retention Manager**:** Provides the capability to manage the processing of partitioned tables in Behavior Detection. This utility purges data from the system based on configurable retention period defined in database.
- Database Statistics Management: The system uses a script to manage Oracle database statistics. These statistics determine the appropriate execution path for each database query.
- Managing Truncate Manager: Truncates tables that require complete replacement of their data.

Figure 36 illustrates the frequency with which you use these batch-related database utilities when managing activities: daily, weekly, monthly, annually, or as needed.

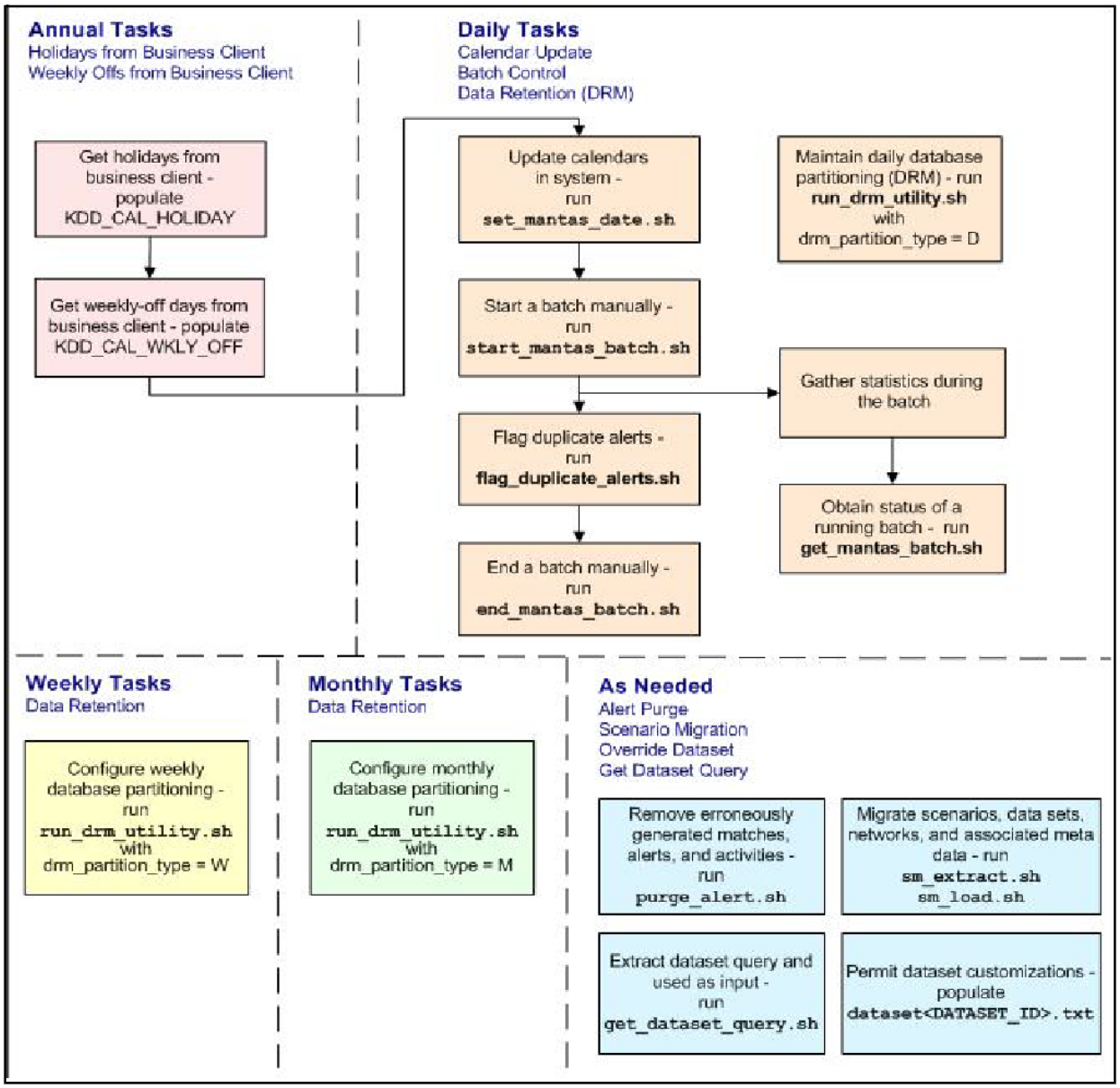


Figure 36: Managing Database Activities with Utilities

Figure 36 illustrates the following:

- Daily tasks are initially dependent on the annual tasks that you perform, such as obtaining holiday and weekly off-days from an Oracle client.
- Daily tasks can include updating calendars and managing batch processes. You may must configure data partitioning on a daily, weekly, or monthly basis.

Tasks that you perform when needed can include deleting extraneous or invalid matches and events, or migrating scenarios and other information from one environment to another, such as from test to production.

6.2 Managing Common Resources for Batch Processing Utilities

Configuration files enable the utilities to share common resources such as database configuration, directing output files, and setting up logging activities. Common resources include the following:

- Install Configuration
- Log4j2.xml Configuration

6.2.1 Install Configuration

Configuration information resides in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` configuration file. The configuration file contains modifiable instructions for Oracle database drivers and provides information that each utility requires. It also provides the user name and password that you must connect to the database. In this file, you can modify values of specific utility parameters, change the locations of output files, and specify database details for extraction and data loading.

The `install.cfg` file contains information unique to each utility and common configuration parameters; headings in the file clearly identify a utility's parameters. You can also modify the current logging configuration , such as activate or deactivate particular logging levels and specify locations for logging entries.

Figure 37 (which appears on the next several pages) provides a sample `install.cfg` file with common and utility-specific information. Logging information appears at the end of the file. Ensure that the ATOMIC schema name is in uppercase.

```
# @(#)Copyright (c) 2018 Oracle Financial Services Software Inc. All
Rights Reserved.
# @(#) $Id: install.cfg $
#
# This configuration file supports the following database utilities:
#  Calendar Mangager
#  Batch Control
#  Truncate Manager
#  Scenario Migration
#  Alert Purge
#  Data Retention Manager
#  Email Notification
#  Data Analysis Tool
# The file contains some properties that are common and specific
properties for each
# of the tools.
```

(*Continued on next page*)

(*Continued from previous page*)

```
################ COMMON CONFIGURATION ENTRIES ######################


NLS_LENGTH_SEMANTICS=CHAR

database.driverName=oracle.jdbc.driver.OracleDriver

utils.database.urlName=jdbc:oracle:thin:@ofss2221324.in.oracle.com:1521
:Ti5O12L64

utils.database.username=f802_fccm

utils.database.password=NzBXdzslR43hh0nWkaqYvA==

schema.algorithms.owner=f802_fccm

schema.algorithms.password=NzBXdzslR43hh0nWkaqYvA==

schema.web.owner=f802_fccm

schema.web.password=NzBXdzslR43hh0nWkaqYvA==

schema.report.owner=f802_fccm

schema.report.password=NzBXdzslR43hh0nWkaqYvA==


schema.mantas.owner=f802_fccm

schema.mantas.password=NzBXdzslR43hh0nWkaqYvA==

utils.miner.user=f802_fccm

utils.miner.password=NzBXdzslR43hh0nWkaqYvA==

schema.business.owner=f802_fccm

schema.business.password=NzBXdzslR43hh0nWkaqYvA==

schema.market.owner=f802_fccm

schema.market.password=NzBXdzslR43hh0nWkaqYvA==

utils.data.directory=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/data

ingest.user=f802_fccm

ingest.password=NzBXdzslR43hh0nWkaqYvA==


schema.kdd.owner=f802_fccm

schema.kdd.password=NzBXdzslR43hh0nWkaqYvA==

casemng.schema.owner=f802_fccm

casemng.schema.password=NzBXdzslR43hh0nWkaqYvA==
```

(Continued on next page)

(Continued from previous page)

```
################ CALENDAR MANAGER CONFIGURATION ##################


# The look back and look forward days of the provided date.

# These values are required to update the KDD_CAL table. The maximum
look back or forward

# is 999 days.

calendar.lookBack=400

calendar.lookForward=14


############### BATCH CONTROL CONFIGURATION ####################


# When ending the batch, age alerts in calendar or business days

age.alerts.useBusinessDays=Y


############### TRUNCATE MANAGER ################################


# Specify the database username and password for truncation manager

truncate.database.username=${ingest.user}

truncate.database.password=${ingest.password}


################ SCENARIO MIGRATION CONFIGURATION
######################


#### GENERAL SCENARIO MIGRATION SETTINGS


#Specify the flags for whether scoring rules and wrapper datasets need
to be extracted or loaded

score.include=N

wrapper.include=N


#Specify the Use Code for the scenario. Possible values are 'BRK' or
'EXP'

load.scnro.use=BRK
```

(Continued on next page)

(Continued from previous page)

```
#If custom patterns exist for a product scenario, set to 'Y' when
loading a scenario hotfix.

#This should normally be set to 'N'.

load.ignore.custom.patterns=N


#Specify the full path of depfile and name of fixfile used for
extraction and loading

#Note : fixfile need not be specified in case of loading

sm.depfile=/scratch/ofsaadb/BD802_Final/BD802FL/database/db_tools/
mantas_cfg/dep.cfg


sm.release=5.7.1


#### EXTRACT


# Specify the database details for extraction

extract.database.password=${utils.database.password}


# Specify the case schema name for both extraction and load .

caseschema.schema.owner=f802_fccm


# Specify the jdbc driver details for connecting to the source database

extract.conn.driver=${database.driverName}

extract.conn.url=jdbc:oracle:thin:@ofss2221324.in.oracle.com:1521/
Ti5O12L64


#Source System Id

extract.system.id=


# Specify the schema names for Extract

extract.schema.mantas=${schema.mantas.owner}

extract.schema.case=f802_fccm

extract.schema.business=${schema.business.owner}

extract.schema.market=${schema.market.owner}
```

(Continued on next page)

(Continued from previous page)

```
extract.user.miner=${load.user.miner}

extract.miner.password=${utils.miner.password}


# File Paths for Extract


#Specify the full path in which to place extracted scenarios

extract.dirname=/scratch/ofsaadb/BD802_Final/BD802FL/database/db_tools/
data


#Specify the full path of the directory where the backups for the
extracted scripts would be maintained

extract.backup.dir=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/data/temp


#Controls whether jobs and thresholds are constrained to IDs in the
product range (product.id.range.min

# through product.id.range.max). Values are Y and N. If the range is not
restricted, you can use range.check


# to fail the extract if there are values outside the product range.

extract.database.password=${utils.database.password}


# Specify the case schema name for both extraction and load .

caseschema.schema.owner=f802_fccm


# Specify the jdbc driver details for connecting to the source database

extract.conn.driver=${database.driverName}

extract.conn.url=jdbc:oracle:thin:@ofss2221324.in.oracle.com:1521/
Ti5O12L64


#Source System Id

extract.system.id=


# Specify the schema names for Extract
```

(Continued on next page)

(Continued from previous page)

```
extract.schema.mantas=${schema.mantas.owner}

extract.schema.case=f802_fccm

extract.schema.business=${schema.business.owner}

extract.schema.market=${schema.market.owner}

extract.user.miner=${load.user.miner}

extract.miner.password=${utils.miner.password}


# File Paths for Extract


#Specify the full path in which to place extracted scenarios

extract.dirname=/scratch/ofsaadb/BD802_Final/BD802FL/database/db_tools/
data


#Specify the full path of the directory where the backups for the
extracted scripts would be maintained

extract.backup.dir=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/data/temp


#Controls whether jobs and thresholds are constrained to IDs in the
product range (product.id.range.min

# through product.id.range.max). Values are Y and N. If the range is not
restricted, you can use range.check


# to fail the extract if there are values outside the product range.

extract.product.range.only=N

extract.product.range.check=N


#### LOAD


# Specify the jdbc driver details for connecting to the target database

load.conn.driver=${database.driverName}

load.conn.url=${utils.database.urlName}


#Target System ID
```

(Continued on next page)

(*Continued from previous page)*

```
load.system.id=Ti5O12L64
# Specify the schema names for Load
load.schema.mantas=${schema.mantas.owner}
load.schema.case=f802_fccm
load.schema.business=${schema.business.owner}
load.schema.market=${schema.market.owner}
load.user.miner=${utils.miner.user}
load.miner.password=${utils.miner.password}.
#Directory where scenario migration files reside for loading
load.dirname=/scratch/ofsaadb/BD802_Final/BD802FL/database/db_tools/
data
# Specify whether threshold can be updated
load.threshold.update=Y
# Specify whether score can be updated
load.score.update=Y

# Specify whether or not to verify the target environment on load
verify.target.system=N

################ ALERT PURGE CONFIGURATION ##########################
# Set the Alert Purge input variables here.
# (use the word "null" as the value of any parameters that are not
#  to be used)
#

# Specify whether or not to consider Matches
limit_matches=N

# Specify whether or not to purge the data
purge=Y

# Specify batch size for which commit should perform
batch_size=5000
```

(Continued on next page)

(Continued from previous page)

```
job=null
scenario=null
# enter dates, with quotes in the following format:
#   'DD-MON-YYYY HH24:MI:SS'
start_date=null
end_date=null
alert_status=NW

# Specify purge db user
purge.database.user=f802_fccm

# Specify purge db user password.
purge.database.password=

# Specify whether alerts has to be purged or not.
purge_alert_flag=Y

# Specify whether fatca cases/assessments has to be purged or not.
purge_fatca_flag=Y

# Specify whether case has to be purged or not.
purge_case_flag=Y

# Specify defualt rule set.
purge_default_rule_set=

# Specify total number of threads should be used for the process.
purge_threads_no=10

# Specify report directory for report on process performed.
purge_report_directory=

# Specify product version
```

(Continued on next page)

(Continued from previous page)

```
purge_product_version=

#Base Working Directory required to put the temporary log from Database
Server

ap.storedproc.logdir=/tmp


#The common Path required to put the SQL files to execute

commonSQLFilePath=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/data

######### DATA RETENTION MANAGER CONFIGURATION ########################

#

# Set the Data Retention Manager input variables here.

##

drm_operation=P

drm_partition_type=D

drm_owner=${schema.business.owner}

drm_object_name=A

drm_weekly_proc_fl=N

######### Email Notification ##########################################

#

# The following sections contain information on configuring email

# notification information. If you wish to use Exchange, you must
purchase

# Java Exchange Connector, obtain a license and the jec.jar file. The
license

# file must be placed in the mantas_cfg file, and the jec.jar file must
be

# copied to the db_tools/lib directory. Then, edit the file

# db_tools/bin/run_push_email.ksh, uncomment the JEC_JARS= line.

#

######################################################################
#

# Currently only smtp, smtps, or exchange

email.type=smtp
```

(Continued on next page)

(Continued from previous page)

```
# Number of notifications that can run in parallel
notification.threads=4


# Max number of active db connections
utils.database.max_connections=4
email.style.tr=font-size:10pt
email.style.td=border:1px solid #000; border-collapse:collapse;
padding: 4px
email.style.footer=font-family:Arial, Helvetica, sans-serif;font-
size:10pt; color:black;
email.style.disclaimer=font-style: italic;


######### PDF ARCHIVE CONFIGURATION ########################
# Set the maximum number of pdf export threads.
pdf.archival.maxthreads=3
# Number of alerts/cases per export web service call.
pdf.archival.service.batchsize=5
# URL of the Alert Management service
alertmanagement.service.url=@ALERT_MANAGEMENT_SERVICE_URL@
######### HIGHLIGHTS GENERATION CONFIGURATION ##########################
#
# Set the default currency code.
#
# See /mantas_cfg/etc/xml/CUR_Currencies.xml for supported currency
# codes.
#
currency.default=USD


######### HDC CONFIGURATION ########################
#
# Set the maximum number of hdc threads.
#
hdc.maxthreads=1
hdc.batchsize=10000
```

(Continued on next page)

(Continued from previous page)

```
######### Data Analysis Tool CONFIGURATION #########################
#

# Username and password for connecting to the database
dat.database.username=${ingest.user}
dat.database.password=${ingest.password}

# Input file for analysis
dat.analysis.input=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/mantas_cfg/analysis_aml.xml

# Output file and file format control
dat.analysis.output=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/data/analysis.html

# Valid values for dat.output.format are HTML and TEXT
dat.output.format=HTML
# Delimiter only applies to TEXT output format
dat.output.delimiter=,
######### Execute Query Tool CONFIGURATION #########################
#
# Username and password for connecting to the database

eqt.database.username=${ingest.user}
eqt.database.password=${ingest.password}
############# Database Builder Utility Configuration ###############
#
# File containing tokens and their value
db_tools.tokenfile=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/mantas_cfg/db_variables.cfg
Oracle.DuplicateRow=1
Oracle.ObjectExists=955,2260,2275,1430,1442,1451,957,1408,2261,1543
Oracle.ObjectDoesNotExist=942,1418,1434,2441,904,4043,1927,2443
```

(Continued on next page)

(Continued from previous page)

```
dbscript.execution.users=(system|business|mantas|market|miner|ingest|re
port|kdd|algorithms|case|config|fatca|ctr|kyc|fsdf|dbutil|web)


############# Correlation Migration Utility Configuration
###############
#
corrRuleMig.CorrRuleFileNm=
corrRuleMig.loadHistory=Y
aps.service.url=http://:8070/mantas/services/AlertProcessingService
aps.service.user=test
aps.service.user.password=


############# Config Migration Utility Configuration ###############
config.filenm.prefix=Config


################### LOG CONFIGURATION ################################
#
# Trace SQL exception.  Set to "true" for SQL tracing,
# "verbose" to trace low-level JDBC calls
#
com.sra.kdd.tools.database.debug=true
# Specify which priorities are enabled in a hierarchical fashion, i.e.,
if
# DIAGNOSTIC priority is enabled, NOTICE, WARN, and FATAL are  also
enabled,
# but TRACE is not.
# Uncomment the desired log level to turn on appropriate level(s).
# Note, DIAGNOSTIC logging is used to log database statements and will
slow
# down performance.  Only turn on if you need to see the SQL statements
being
# executed.
# TRACE logging is used for debugging during development.  Also only
```

(Continued on next page)

(Continued from previous page)

```
turn on
# TRACE if needed.
log.fatal=true
log.warning=true
log.notice=true
log.diagnostic=true
log.trace=true
log.time.zone=US/Eastern


# Specify whether logging for a particular level should be performed
# synchronously or asynchronously.
log.fatal.synchronous=true
log.warning.synchronous=true
log.notice.synchronous=true
log.diagnostic.synchronous=true
log.trace.synchronous=true


# Specify the format of the log output. Can be modified according to the
format
# specifications at:
# http://logging.apache.org/log4j/docs/api/org/apache/log4j/
PatternLayout.html
# NOTE: Because of the nature of asynchronous logging, detailed
information
# (class name, line number, etc.) cannot be obtained when logging
# asynchronously.  Therefore, if this information is desired (i.e.
specified
# below), the above synchronous properties must be set accordingly (for
the
# levels for which this detailed information is desired). Also note that
this
# type of detailed information can only be obtained for Java code.
log.format=%d [%t] %p %m%n
# Specify the full path and filename of the message library.
log.message.library=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/mantas_cfg/etc/mantas_database_message_lib_en.dat
```

(Continued from previous page)

```
# Specify the full path to the categories.cfg file
log.categories.file.path=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/mantas_cfg/


# Specify where a message should get logged for a category for which
there is
# no location property listed above.
# This is also the logging location of the default MANTAS category
unless
# otherwise specified above.
# Note that if this property is not specified, logging will go to the
console.
log.default.location=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/logs/Utilities.log
# Specify the location (directory path) of the mantaslog, if the
mantaslog
# was chosen as the log output location anywhere above.
# Logging will go to the console if mantaslog was selected and this
property is
# not given a value.
log.mantaslog.location=/scratch/ofsaadb/BD802_Final/BD802FL/database/
db_tools/logs/mantaslog.log


# Specify the hostname of syslog if syslog was chosen as the log output
location
# anywhere above.
# Logging will go to the console if syslog was selected and this
property is
# not given a value.
log.syslog.hostname=


# Specify the hostname of the SMTP server if an e-mail address was
chosen as
# the log output location anywhere above.
# Logging will go to the console if an e-mail address was selected and
this
# property is not given a value.
```

(Continued on next page)

(Continued from previous page)

```
log.smtp.hostname=


# Specify the maxfile size of a logfile before the log messages get
rolled to
# a new file (measured in MBs).
# If this property is not specified, the default of 10 MB will be used.
log.max.size=


#NOTE: The values for the following variables need not be changed
# Specify the ID range for wrapper datasets
dataset.wrapper.range.min=113000001
dataset.wrapper.range.max=114000000
product.id.range.min=113000000
product.id.range.max=200000000
```

Figure 37: Sample `install.cfg` File

6.2.1.1 Log4j2.xml Configuration

In the `<OFSAAI Installed Directory>/database/db_tools/log4j2.xml files` file, you can modify the default location to where you want to direct logging output for each utility. The entries that you make require a specific format; the file contains instructions and examples of correct formatting. Figure 38 provides a sample Log4j2.xml file.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

<Appenders>

<RollingFile name="CALENDAR_MANAGER" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/calendar_manager.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/calendar_manager.log</FileN
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [CALENDER_MANAGER] [%5p] - %
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>

<RollingFile name="PURGE_UTIL" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/purge.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/purge.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [PURGE_UTIL] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>

<RollingFile name="BATCH_CONTROL" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/batch_control.log">
```

(Continued on next page)

(Continued from previous page)

```
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/batch_control.log</
FileName>
      <PatternLayout>
<Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [BATCH_CONTROL] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
<SizeBasedTriggeringPolicy size="10000kb"/>
</Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="DATA_RETENTION_MANAGER" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/DRM_Utility.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/DRM_Utility.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [DATA_RETENTION_MANAGER]
[%5p] - %m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="TRUNCATE_MANAGER" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/truncate_manager.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/truncate_manager.log</
FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [TRUNCATE_MANAGER] [%5p] -
%m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
```

(Continued from previous page)

```
      </Policies>
     <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="COMMON_UTILITIES" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/common_utilities.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/common_utilities.log</
FileName>
      <PatternLayout>
<Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [COMMON_UTILITIES] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
```

(Continued on next page)

```
<SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="EXTRACT" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/extract.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/extract.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [EXTRACT] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="LOAD" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/load.log">
```

(Continued on next page)

(Continued from previous page)

```
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/load.log</FileName>
      <PatternLayout>
      <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [LOAD] [%5p] - %m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>

<RollingFile name="REFRESH_TEMP_TABLE" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/refresh_temp_table.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/refresh_temp_table.log</
FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [REFRESH_TEMP_TABLE] [%5p]
-
%m%n</Pattern>
      </PatternLayout>
      <Policies>
<SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>

<RollingFile name="RUN_STORED_PROCEDURE" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/run_stored_procedure.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/run_stored_procedure.log</
FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [RUN_STORED_PROCEDURE]
[%5p] - %m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
```

(Continued on next page)

(Continued from previous page)

```
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="GET_DATASET_QUERY" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/get_dataset_query.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/get_dataset_query.log</
FileName>
      <PatternLayout>
       <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [GET_DATASET_QUERY] [%5p] -
%m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="DATA_ANALYSIS_TOOL" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/data_analysis_tool.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/data_analysis_tool.log</
FileName>
      <PatternLayout>
<Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [DATA_ANALYSIS_TOOL] [%5p] -
%m%n</Pattern>
      </PatternLayout>
      <Policies>
<SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="DB_BUILDER" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/db_builder.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/db_builder.log</FileName>
```

(Continued on next page)

(Continued from previous page)

```
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [DB_BUILDER] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>
<RollingFile name="ARCHIVE_PDF" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/pdf_archive.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/pdf_archive.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [ARCHIVE_PDF] [%5p] -
%m%n</Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>



<RollingFile name="HIGHLIGHT_GENERATOR" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/highlight_generator.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/highlight_generator.log</
FileName>
      <PatternLayout>
       <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [HIGHLIGHT_GENERATOR] [%5p]
-
%m%n</Pattern>
      </PatternLayout>
      <Policies>
<SizeBasedTriggeringPolicy size="10000kb"/>
```

(Continued on next page)

(Continued from previous page)

```
        </Policies>
  <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="HDC" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/hdc.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/hdc.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [HDC] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<RollingFile name="REPORT" append="true"
filePattern="@ORION_DB_DBTOOLS_PATH@/logs/report.log">
      <FileName>@ORION_DB_DBTOOLS_PATH@/logs/report.log</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [REPORT] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>


<Console name="stdout" target="SYSTEM_OUT">
            <PatternLayout>
                <pattern>
                    [%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} -
%msg%n
```

(Continued on next page)

(Continued from previous page)

```
                </pattern>>
</PatternLayout>
        </Console>
</Appenders>


 <Loggers>
           <Logger name="CALENDAR_MANAGER" level="info"
additivity="false">
   <AppenderRef ref="CALENDAR_MANAGER" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>


<Logger name="PURGE_UTIL" level="info" additivity="false">
   <AppenderRef ref="PURGE_UTIL" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

   <Logger name="BATCH_CONTROL" level="info" additivity="false">
   <AppenderRef ref="BATCH_CONTROL" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

   <Logger name="HDC" level="info" additivity="false">
   <AppenderRef ref="HDC" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>



   <Logger name="HIGHLIGHT_GENERATOR" level="info" additivity="false">
   <AppenderRef ref="HIGHLIGHT_GENERATOR" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>
```

(Continued on next page)

(Continued from previous page)

```
<Logger name="DATA_RETENTION_MANAGER" level="info" additivity="false">
   <AppenderRef ref="DATA_RETENTION_MANAGER" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

   <Logger name="DB_BUILDER" level="info" additivity="false">
   <AppenderRef ref="DB_BUILDER" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
</Logger>

    <Logger name="DB_BUILDER_SQL" level="info" additivity="false">
   <AppenderRef ref="DB_BUILDER" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

     <Logger name="EXTRACT" level="info" additivity="false">
   <AppenderRef ref="EXTRACT" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

     <Logger name="CORRRULEMIGRATIONUTIL_EXTRACT" level="info"
additivity="false">
   <AppenderRef ref="EXTRACT" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

    <Logger name="CONFIGURATIONMIGRATIONUTIL_EXTRACT" level="info"
additivity="false">
   <AppenderRef ref="EXTRACT" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>
```

(Continued on next page)

(Continued from previous page)

```
    <Logger name="LOAD" level="info" additivity="false">
   <AppenderRef ref="LOAD" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

   <Logger name="CORRRULEMIGRATIONUTIL_LOAD" level="info"
additivity="false">
   <AppenderRef ref="LOAD" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

   <Logger name="CONFIGURATIONMIGRATIONUTIL_LOAD" level="info"
additivity="false">
   <AppenderRef ref="LOAD" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

    <Logger name="REFRESH_TEMP_TABLE" level="info" additivity="false">
   <AppenderRef ref="REFRESH_TEMP_TABLE" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

    <Logger name="RUN_STORED_PROCEDURE" level="info"
additivity="false">
   <AppenderRef ref="RUN_STORED_PROCEDURE" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

    <Logger name="GET_DATASET_QUERY" level="info" additivity="false">
   <AppenderRef ref="GET_DATASET_QUERY" level="trace"/>
   <AppenderRef ref="stdout" level="error"/>
           </Logger>

    <Logger name="REPORT" level="info" additivity="false">
```

(Continued from previous page)

```
    <AppenderRef ref="REPORT" level="trace"/>
    <AppenderRef ref="stdout" level="error"/>
            </Logger>

    <Logger name="DATA_ANALYSIS_TOOL" level="info" additivity="false">
    <AppenderRef ref="DATA_ANALYSIS_TOOL" level="trace"/>
    <AppenderRef ref="stdout" level="error"/>
            </Logger>

         <Root level="error">
             <AppenderRef ref="stdout"/>
         </Root>
     </Loggers>
<!-- <root>
<priority value="##PRIORITY##"></priority>
</root> -->
</log4j:configuration>
```

Figure 38: Sample Logging Information in the Log4j2.xml File

6.3 Managing Annual Activities

Oracle requires that you perform certain calendar management tasks at least annually: loading holidays and weekly off-days from an Oracle client. This ensures that the application has the necessary information for populating its own business calendars.

This section covers the following topics:

- Loading Holidays
- Loading Non-business Days

6.3.1 Loading Holidays

On an annual basis, you must populate holidays for the upcoming calendar year into the Behavior Detection `KDD_CAL_HOLIDAY` database table. This ensures that the table contains holidays for at least the next year. Figure 39 provides an example of a SQL script for loading the table.

```
INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '01/01/
2017',
'MM/DD/YYYY'), 'New Year''s Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '01/16/
2017',
'MM/DD/YYYY'), 'Martin Luther King Jr.''s Birthday
- 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '02/20/
2017',
'MM/DD/YYYY'), 'President''s Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '04/14/
2017',
'MM/DD/YYYY'), 'Good Friday - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '05/29/
2017',
'MM/DD/YYYY'), 'Memorial Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '07/04/
2017',
'MM/DD/YYYY'), 'Independence Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
```

```
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '09/04/
2017',
'MM/DD/YYYY'), 'Labor Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '11/22/
2017',
'MM/DD/YYYY'), 'Thanksgiving Day - 2017', 'C');


INSERT INTO KDD_CAL_HOLIDAY ( CLNDR_NM, CLNDR_DT,
HLDY_NM,
HLDY_TYPE_CD ) VALUES ( 'SYSCAL', TO_DATE( '12/25/
2017',
'MM/DD/YYYY'), 'Christmas Day - 2017', 'C');


COMMIT;
```

Figure 39: Sample KDD_CAL_HOLIDAY Table Loading Script

The following table describes the contents of the KDD_CAL_HOLIDAY table.

Table 18: KDD_CAL_HOLIDAY

Column Name	Description
CLNDR_NM	Specific calendar name.
CLNDR_DT	Date that is a holiday.
HLDY_NM	Holiday name , such as Thanksgiving or Christmas.
HLDY_TYPE_CD	Indicates whether the business is Closed (C) or Shortened (S).
SESSN_OPN_TM	Indicates the opening time of the trading session for a shortened day. The format is HHMM.
SESSN_CLS_TM	Indicates the closing time of the trading session for a shortened day. The format is HHMM.
SESSN_TM_OFFSET_TX	Indicates the timezone offset for SESSN_OPN_TM and SESSN_CLS_TM.

When the system runs the set_mantas_date.sh script, it queries the KDD_CAL_HOLIDAY table for the maximum date for each calendar in the table.

> **NOTE** If the maximum date is less than 90 days ahead of the provided date, the process logs a warning message that the specific calendar's future holidays need updating. If any calendars have no holiday records, the system logs a Warning message that the specific calendar has no recorded holidays for the appropriate date range.

6.3.2 Loading Non-business Days

After obtaining non-business days (or weekly off-days; typically Saturday and Sunday) from an Oracle client, load this information for the upcoming calendar year into the `KDD_CAL_WKLY_OFF` table.

The following text provides an example of an SQL script for loading the table.:

```
INSERT INTO KDD_CAL_WKLY_OFF (CLNDR_NM, DAY_OF_WK) VAL-
UES (
 'SYSCAL', 1);


INSERT INTO KDD_CAL_WKLY_OFF (CLNDR_NM, DAY_OF_WK) VAL-
UES (
 'SYSCAL', 7);


    COMMIT;
```

Figure 40: Sample KDD_CAL_WKLY_OFF Table Loading Script

> **NOTE** By default, the system identifies Saturdays and Sundays as non-business days in the system calendar (SYSCAL).

The following table describes the contents of the `KDD_CAL_WKLY_OFF` table.

Table 19: KDD_CAL_WKLY_OFF

Column Name	Description
CLNDR_NM	Specific calendar name.
DAY_OF_WK	Value that represents the day of the week: Sunday=1, Monday=2, Tuesday=3,Wednesday=4, Thursday=5, Friday=6, Saturday=7.

> **NOTE** If the table does not contain records for any calendar in the list, the system logs a Warning message that the specific calendar contains no weekly off-days.

6.4 Managing Alert Purge Utility

The ingestion of certain data can result in the creation of false matches, alerts, and activities. While correction and data re-ingestion is possible, the system does not remove these erroneously generated matches, alerts, and activities automatically.

There may also be cases when the alerts have been residing in the database due to the retention policies imposed by the regulatory bodies, or the internal policies of the respective organization.

The Alert Purge Utility enables you to identify and remove such matches, alerts and cases, and activities selectively, based on a number of parameters (like the Job ID, Scenario ID, Scenario Class, or a date range with optional alert status codes). Additional parameters enable you to simulate a purge run to determine all found matches, alerts, and activities using the input parameters. You can also limit the alerts in the purge process only to those that contain false matches.

The utility consists of a UNIX shell script, Java executables, a XML File and a configuration file in which you define the process parameters to use in the purge processing. The system directs output to a configurable log file; processing appends this log with information about subsequent executions of the scripts.

This section covers the following topics:

- Directory Structure
- Logs
- Precautions
- Using the Alert Purge Utility
- Sample Alert Purge Processes

6.4.1 Directory Structure

The following table describes the directory structure for the Alert Purge Utility.

Table 20: Alert Purge Utility Directory Structure

Directory	Description
bin/	Contains executable files, including the run_alert_purge.sh shell script.
lib/	Contains required class files in .jar format.
mantas_cfg/	Contains configuration files , such as `install.cfg` and `categories.cfg`, in which you can configure properties and logging attributes.
logs/	Keeps the `<OFSAAI Installed Directory>/database/db_tools/logs/purge.log file` that the utility generates during execution.
data/	Keeps .sql files for execution.
.xml	Contains the Purge Rules Configuration File (PurgeRules.xml), which is used for configuring the Alert Purge rules.

6.4.2 Logs

As the Alert Purge Utility performs alert detection activities, it generates a log that it enters in the `<OFSAAI Installed Directory>/database/db_tools/logs/purge.log` file (the logging process time-stamps all entries). The log file contains relevant information such as status of the purge processing, log-relevant information, and error records.

You can modify the current logging configuration for the Alert Purge Utility in the `<OFSAAI Installed Directory>/database/db_tools/log4j2.xml files`. For more information about logging in these configuration files, refer to Managing Common Resources for Batch Processing Utilities and Appendix A, Logging for more information.

6.4.3 Precautions

You use the utility to rid the system of falsely-generated matches and alerts or cases. Other than recorded information in the `<OFSAAI Installed Directory>/database/db_tools/logs/purge.log` file, the system does not capture audit information for this process. The utility does not update other alerts' prior counts as a result of purging alerts.

> **NOTE** The utility also purges any alert or case which is used to trigger Auto Suppression or establish Trusted Parties. However, this would not affect the Suppression Rule or the Trusted Pair except that the `kdd_auto_suppr_alert.trgr_alert_id`, `kdd_trusted_pair.trgr_alert_id`, or `kdd_trusted_pair.trgr_case_id` columns are set to a null value
>
> Run the Alert Purge Utility one process at a time. Multiple, simultaneous executions of the utility may lead to unexpected results and compromise the relational integrity of match, alert, and action data.When no users are editing or viewing any of the alerts, actions, or associated information (including matches derived from the alerts and actions specified, alerts derived from the specified actions, and actions derived from the specified alerts). However, you can run the utility during editing or viewing of other alerts and related information. You can also run the utility during alert post-processing, subject to time constraints.

6.4.4 Using the Alert Purge Utility

The Alert Purge Utility is not part of an automated batch process. You run this manual process only when necessary (refer to Figure 36). The following sections describe configuring and executing the utility, as well as the utility's process flow:

- Configuring the Alert Purge Utility
- Executing the Alert Purge Utility
- Processing for Purging

6.4.4.1 Configuring the Alert Purge Utility

To configure the Alert Purge Utility, follow these steps:

3. Navigate to the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg`.
4. Edit the parameters in the install.cfg file to the desired settings. This file contains common configuration information that the Alert Purge Utility and other utilities require for processing (refer to Figure 37). The following is a sample section from the `install.cfg` file for configuration information specific to this utility:

```
################ ALERT PURGE CONFIGURATION ##########################
# Set the Alert Purge input variables here.
# (use the word "null" as the value of any parameters that are not
#  to be used)
#

# Specify whether or not to consider Matches
limit_matches=N

# Specify whether or not to purge the data
purge=Y

# Specify batch size for which commit should perform
batch_size=5000

job=null
scenario=null
# enter dates, with quotes in the following format:
#   'DD-MON-YYYY HH24:MI:SS'
start_date=null
end_date=null
alert_status=NW

# Specify purge db user
purge.database.user=f802_fccm

# Specify purge db user password.
purge.database.password=

# Specify whether alerts has to be purged or not.
purge_alert_flag=Y

# Specify whether fatca cases/assessments has to be purged or not.
```

(Continued on next page)

(Continued from previous page)

```
purge_fatca_flag=Y

# Specify whether case has to be purged or not.
purge_case_flag=Y

# Specify defualt rule set.
purge_default_rule_set=

# Specify total number of threads should be used for the process.
purge_threads_no=10

# Specify report directory for report on process performed.
purge_report_directory=

# Specify product version
purge_product_version=



#Base Working Directory required to put the temporary log from
Database Server
ap.storedproc.logdir=/tmp

#The common Path required to put the SQL files to execute
commonSQLFilePath=/scratch/ofsaadb/BD804_Final/BD804FL/database/
db_tools/data
```

Figure 41: Configuration Information

> **NOTE** Not specifying a value of *null* , such as leaving a value blank, in this section of the `install.cfg` file causes undesirable results.

The following table describes required and optional parameters for this utility.

Table 21: Alert Purge Utility Parameters

Parameter	Description
purge	Determines how the utility performs processing, depending on the specified value: • N (default): Performs all processing up to the point of the purge. The utility identifies resulting matches, alerts, and actions, but performs no purging. • Y: Performs the above in addition to purging matches, alerts, and actions.
limit_matches	Identifies restrictions on the matches to delete: • Y (default): If a match that you want to delete is part of an alert that contains matches that you do not want to delete, do not delete this match either (applies to multi-match alerts). • N: Deletes all selected matches for purging based on the input criteria. The utility deletes only alerts and associated actions that exclusively contain matches to be purged. **Note:** The system purges matches that do not relate to alerts, regardless of the value of `limit_matches`.
batch_size	*Optional:* Sets the batch size of purge actions to minimize log space use. Specifying a non-positive value or specifying no value uses the default of 5,000 rows.
purge_alert_flag	Determines whether or not the utility would purge alerts, depending on the specified value: • N: Does not purge the alerts irrespective of whether or not they identified according to the purge rule being used. This may be used when purging only the cases. • Y (default): Purges the alerts as identified by the purge rule used to perform the purge operation.
purge_case_flag	Determines whether or not the utility would purge cases, depending on the specified value: • N: Does not purge the cases irrespective of whether or not they identified according to the purge rule being used. This may be used when purging only the cases. • Y (default): Purges the cases as identified by the purge rule used to perform the purge operation.
purge_default_rule_set	*(Optional)* Indicates the default set of rules to be used for purging alerts/cases. You may either specify the purge rules to be used against this parameter, or pass the name of the specific purge rules) as command line parameters. You may specify a single purge rule, or a comma separated list of purge rules to be used as default when no other purge rule is provided from the command line.
purge_threads_no	*(Optional)* Identifies the number of concurrent threads to create for purging the alerts to optimize the performance. Specifying a non-positive value or specifying no value uses the default of 10 threads.
purge_report_directory	Identifies the absolute path to the directory where the purge activity report should be generated. The report file name has a name similar to Purge_<YYYYMMDD.HH.MM.SS>.txt. Here <YYYYMMDD.HH.MM.SS> represents current timestamp when the utility was executed.

Table 21: Alert Purge Utility Parameters (Continued)

Parameter	Description
purge_product_version	Identifies the OFSBD Product Version installed by the client.

The `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/etc/xml/PurgeRules.xml` file contains purge rules configuration information that the Alert Purge Utility requires for processing. The following sample section from the `PurgeRules.xml` file provides configuration information for this utility.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:RuleSet xmlns:xs="http://namespaces.mantas.com/RuleSet">
  <Alert>
    <Rule id="1">
    <IdentifierList>286,4565,4537</IdentifierList>
 <ScenarioIdList>114697002</ScenarioIdList>
      <ScenarioClassList>CR</ScenarioClassList>
      <CreateDate>
        <StartDate>2011-05-25</StartDate>
        <EndDate>2011-05-25</EndDate>
      </CreateDate>
      <DomainCode>MTS</DomainCode>
      <BatchId>2</BatchId>
      <ThresholdSetIds>118745206,118710066</ThresholdSetIds>
      <LastActionDate>
        <StartDate>2016-05-25</StartDate>
        <EndDate>2016-05-25</EndDate>
      </LastActionDate>
      <Status>CL</Status>
      <JobIds>102202</JobIds>
    </Rule>
   </Alert>
  <Case>
    <Rule id="2">
      <IdentifierList>CA51300004,CA3773,CA3757,CA3766</
IdentifierList>
      <CaseTypeList>FR_EE,FR_ON</CaseTypeList>
      <CreateDate>
        <Age>1Y</Age>
```

```
        </CreateDate>
        <LastActionDate>
          <StartDate>2016-06-22</StartDate>
          <EndDate>2016-06-22</EndDate>
</LastActionDate>
    </Rule>
    </Case>
</xs:RuleSet>
```

Figure 42: Configuration Information

The following table describes the Purge Rules Configuration Parameters.

Table 22: Alert Purge Utility Parameters

Parameter	Description
Alert/Case	Identifies and encapsulates the purge rules for Alerts/Cases. You may define any number of purge rules for both alerts and cases.
Rule	Identifies a set of rules to be used for purging Alert/Case Information. All Alert Purge rules defined in this file must be provided a unique positive integer ID (as specified against the ID attribute). The value provided against the ID attribute is used by the utility to identify the rules to be used for carrying out the purge operations. **Note:** Not specifying a unique value for the ID attribute may lead to undesirable results.
IdentifierList	Identifies a list of Alert and Case IDs to be purged. You may specify more than one alert or case ID by separating them by comma.
ScenarioIdList	Identifies a list of Scenario IDs for which the alerts are to be purged. You may specify more than one Scenario ID by separating them by comma. **Note:** This property is specific to alerts only. This should not be specified for cases
ScenarioClassList	Identifies a list of Scenario Class for which the alerts are to be purged. You may specify more than one Scenario Class by separating them by comma. **Note:** This property is specific to alerts only. This should not be specified for cases

Table 22: Alert Purge Utility Parameters (Continued)

Parameter	Description
CreateDate	Identifies the dates to be considered for purging the alerts or cases by their creation date. The date range may be provided in terms of Start Date or End Date, or the Age of the Alert or Case calculated from the current day/month/year. • **StartDate**: Identifies the date from when the alerts/cases are to be considered for purging. The date should be provided in the format YYYY-MM-DD. • **EndDate**: Identifies the date up to which the alerts are to be purged. The date should be provided in the format YYYY-MM-DD • **Age**: Identifies the age of the Alert/Case to be purged relative to the current date/month/year. Acceptable values for this parameter constitutes a non-negative number followed by D (Days), M (Months) or Y (Years). If we specify age of a record is 1 Day means it should complete 1 day in the database. That is from current day to yesterday. The following example gives more details: (Assume Current date: 21 NOV 2012) 1. Case1: a. If age = 1Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2011 (includes both days) b. If age = 5Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2007 (includes both days) 1. Case2: a. If age = 1M: Date range would be considered: 21 NOV 2012 to 21 OCT 2012 (includes both days) b. If age = 5M: Date range would be considered: 21 NOV 2012 to 21 JUN 2012 (includes both days) 1. Case3: a. If age = 1D: Date range would be considered: 21 NOV 2012 to 20 NOV 2012 (includes both days) b. If age = 5D: Date range would be considered: 21 NOV 2012 to 16 NOV 2012 (includes both days) c. If age = 0D: Date range would be considered: 21 NOV 2012 to 21 NOV 2012 (that is, current date only) **Note:** If only EndDate is specified, utility would consider it as on or before that date, in case of only StartDate being provided, utility would consider it as on or after that date. In-case both dates are specified utility would consider both the dates and the dates in between them.
BatchId	Identifies the list of Batch IDs for which the alerts should be purged. **Note:** This property is specific to alerts only. This should not be specified for cases.

Table 22: Alert Purge Utility Parameters (Continued)

Parameter	Description
DomainCode	Identifies the list of domains for which the alerts should be purged. Acceptable values include: • MTS • TST • PFM • NVZ **Note:** This property is specific to alerts only. This should not be specified for cases.
LastActionDate	Identifies the dates to be considered for purging the alerts and cases by he date on which last action was taken on them. The date range may be provided in terms of Start Date or End Date, or the Age of the Alert or Case calculated from the current day/month/year. • **StartDate**: Identifies the date from when the alerts/cases are to be considered for purging.The date should be provided in the format YYYY-MM-DD • **EndDate**: Identifies the date up to which the alerts are to be purged. The date should be provided in the format YYYY-MM-DD • **Age**: Identifies the age of the Alert or Case to be purged relative to the current date/month/year. Acceptable values for this parameter constitutes a non-negative number followed by D (Days), M (Months) or Y (Years). If we specify age of a record is 1 Day means it should complete 1 day in the database. That is from current day to yesterday. The following example gives more details: (Assume Current date: 21 NOV 2012) 1. Case1: a. If age = 1Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2011 (includes both days) b. If age = 5Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2007 (includes both days) 2. Case2: a. If age = 1M: Date range would be considered: 21 NOV 2012 to 21 OCT 2012 (includes both days) b. If age = 5M: Date range would be considered: 21 NOV 2012 to 21 JUN 2012 (includes both days) 3. Case3: a. If age = 1D: Date range would be considered: 21 NOV 2012 to 20 NOV 2012 (includes both days) b. If age = 5D: Date range would be considered: 21 NOV 2012 to 16 NOV 2012 (includes both days) c. If age = 0D: Date range would be considered: 21 NOV 2012 to 21 NOV 2012 (that is, current date only) **Note:** If only EndDate is specified, utility would consider it as on or before that date, in case of only StartDate being provided, utility would consider it as on or after that date. If both dates are specified utility would consider both the dates and the dates in between them.

Table 22: Alert Purge Utility Parameters (Continued)

Parameter	Description
Status	Identifies a list of Status Codes against which the Alert or Case should be purged. You may specify more than one Status Code by separating them by comma.
JobIds	Identifies the list of Job IDs for which the alerts should be purged. You may specify more than one Job ID by separating them by comma. **Note:** This property is specific to alerts only. This should not be specified for cases.
ThresholdSetIds	Identifies the list of Threshold Set IDs for which the alerts should be purged. You may specify more than one Threshold Set ID by separating them by comma. **Note:** This property is specific to alerts only. This should not be specified for cases.

6.0.0.1 Executing the Alert Purge Utility

To execute the Alert Purge Utility, follow these steps:

1. Verify that the TBAML database is operational:

   ```
   tnsping <database instance name>
   ```

2. Verify that the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` configuration file contains the correct source database connection and logging information.

3. Access the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

4. Start the Alert Purge shell script:

   ```
   run_alert_purge.sh -purge
   ```

   Executing this command sets the environment classpath and starts the utility. You may also pass command line arguments to the utility, and execute the utility in any of the following ways:

   - You may pass a list of purge rules (as configured in `PurgeRules.xml` file) separated by a comma (,) following the convention of `alert_rule_<i0>` for alert-related rules and `case_rule_<i0>` for case-related rules; here `i0` is an integer representing the corresponding rule number in the `purgeRules.xml` file.

     ```
     ./run_alert_purge.sh  -purge
     alert_rule_<i0>,alert_rule_<i1>,case_rule_<i2>….
     ```

   - You may instruct the utility not to purge any alerts, but only cases, and vice-versa. If the value passed is 'alert=N' the utility considers this as no to purge alerts

     ```
     ./run_alert_purge.sh  -purge  alert=N
     ```

     If the value passed is `'case=N'` the utility considers this as no to purge cases

     ```
     ./run_alert_purge.sh  -purge  case=N
     ```

   - You may instruct the utility only to simulate the purge process and not purge the alerts and cases by passing a command line parameter '`test=Y`'. In this case, the utility considers this as running in test mode and generates the report of alerts and cases that would have purged.

     ```
     ./run_alert_purge.sh  -purge  test=Y
     ```

   - You can provide all these parameters or a combination of these parameters irrespective of order, once at a time, to the utility as shown in the example below:

```
./run_alert_purge.sh   -purge  case=N alert_rule_<i0>,alert_rule<i1>
test=Y
```

> **NOTE** If the utility is executed without any command line arguments, the utility considers purging the alerts and cases as configured in the `install.cfg` file.

6.0.0.2 Processing for Purging

The process for purging is as follows:

1. Once you execute the `run_alert_purge.sh` script, the Alert Purge Utility generates a listing of actions, matches, and alerts or cases that it must purge according to the rules specified at the command line, or the default rule set configured in the `install.cfg`file.
2. After the script is executed, the actions, alerts, and cases are recorded in the `<OFSAAI Installed Directory>/database/db_tools/logs/purge.log` file.

> **NOTE** The utility presumes that you have determined the input parameters to specify what matches, alerts, and actions to purge. The utility does not check against the data to verify what it should purge.
>
> To capture the SQL statements naming, set `log.diagnostic=true` in the `install.cfg`.

3. The utility then purges actions, then matches, then alerts, according to the contents of the `KDD_AP_ACTION, KDD_AP_MATCH`, and `KDD_AP_ALERT` tables.
4. The utility captures purging results and any errors in the `purge.log` and a report (having the naming convention `Purge_<YYYYMMDD.HH.MM.SS>.txt`) files.

> **NOTE** The Alert Purge Utility purges data from archive tables for erroneous alerts. Also, the system does not update score and previous match count values associated with generated matches and alerts since creation of the erroneous matches.

6.0.0.2.1 Automatic Restart Capability

The Alert Purge Utility has an automatic restart capability in that any interruption in the purge processing resumes at that point, regardless of the input parameters. The system documents log information about the interruption in the `<OFSAAI Installed Directory>/database/db_tools/logs/purge.log` file. Otherwise, any restart that has not progressed to the purge component behaves as a new processing run.

The restart capability allows interrupted purges to resume at a convenient point, but is unable to execute all desired input parameters.

6.0.1 Sample Alert Purge Processes

This section includes examples of the Purge Alerts process based on input parameters. These example patterns are also applicable for filtering cases.

6.0.1.1 Example 1

The user specifies only one rule 'xyz' for purging alerts and assumes it as follows:

```
<Alert>
……..
    <Rule id="xyz">
      <IdentifierList>3775,3731,3669,3663</IdentifierList>
<Status>CL</Status>
</Rule>
……..
</Alert>
```

The utility filters in the existing alerts for IDs 3775,3731,3669,3663 and* status having Closed (CL).

Here and* specifies the logical and operation specified by sql.

In this case, the alert has closed status among the existing alert IDs of (3775, 3731, 3669, and 3663).

```
<Alert>
……..
<Rule id="xyz">
<IdentifierList>3775,3731,3669,3663</IdentifierList>
<Status>CL</Status>
<ScenarioIdList>114697002, 114690106</ScenarioIdList>
<JobIds>456789</JobIds>
</Rule>
……..
</Alert>
```

The utility filters in the existing alerts for IDs 3775,3731,3669,3663 and* having status Closed (CL) and* having Scenario IDs 114697002,114690106 and having Job Id 456789.

6.0.1.2 Example 2

The user specifies multiple rules for purging:

```
<Alert>
……..
<Rule id="pqr">
<IdentifierList>3775, 3731,3669,3663</IdentifierList>
<Status>CL</Status>
<JobIds>456789</JobIds>
</Rule>
<Rule id="xyz">
<ScenarioIdList>114697002,114690106</ScenarioIdList>
<CreateDate>
<StartDate>2011-05-25</StartDate>
```

```
<EndDate>2011-05-29</EndDate>
</CreateDate>
</Rule>
……..
</Alert>
```

The utility prepares a query to filter alerts so that rule 'pqr' (fetches alerts as per the single rule described above) or* rule 'xyz' (fetches alerts as per the single rule described above) or*... That is, union of the alerts from all the rules would be filtered.

Here or* specifies the logical or operation specified by sql.

6.1 Managing Batch Control Utility

The Batch Control Utility enables you to manage and record the beginning and ending of a batch process. It also enables you to access the currently running batch. You control the process through a job scheduling tool such as Maestro or Unicenter Autosys.

This utility consists of a Java file that resides in the directory `<OFSAAI Installed Directory>/database/db_tools/lib` and UNIX script files that reside in `<OFSAAI Installed Directory>/database/db_tools/bin`:

- `start_mantas_batch.sh` starts the batch process.
- `end_mantas_batch.sh` ends the batch process.
- `get_mantas_batch.sh` obtains the name of the currently running batch.

The utility also uses common parameters in the configuration file `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` (refer to Install Configuration for more information).

This section covers the following topics:

- Batches in TBAML
- Directory Structure
- Logs
- Using the Batch Control Utility

> **NOTE** To calculate the age in business days versus calendar days, verify that the `age.events.useBusinessDays` setting in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file has a value of Y (yes).

6.1.1 Batches in TBAML

Except for the behavior detection subsystem, batches govern all other activity in the TBAML system. A batch provides a method of identifying a set of processing. This includes all activities associated with data management and Behavior Detection.

Deployment of a system can be with a single batch or with multiple batches. You can use multiple batches to permit intra-day processing to generate results several times per day, or to separate processing based on servicing multiple time zones.

TBAML provides two types of batches:

- **End-of-day**: Represent processing at the completion of a business day for a set of data. Some processes are only appropriate for end-of-day batches. For example, daily activity summary derivations and calculating event ages are activities that occur only in end-of-day batches. Multiple end-of-day batches per day can run if the Behavior Detection installation supports multiple time zones , such as New York and Singapore.
- **Intra-day**: Used when loading data between end-of-day batches to obtain more frequent detection results. For example, running a batch of trading-compliance scenarios at 10:00 A.M. can identify behaviors relevant to the opening of the market without waiting for the end of the day to be able to act.

6.1.2 Directory Structure

Table 23 provides the directory structure for the Batch Control Utility, in `<OFSAAI Installed Directory>/database/db_tools/`:

Table 23: Batch Control Utility Directory Structure

Directory	Contents
bin/	Executable files, including the `start_mantas_batch.sh`, `end_mantas_batch.sh`, and `get_mantas_batch.sh` shell scripts.
lib/	Required class files in .jar format.
mantas_cfg/	Configuration files , such as `install.cfg` and `categories.cfg`, in which you can configure properties and logging attributes.
logs/	File `batch_control.log` that the utility generates during execution.

6.1.3 Logs

As the Batch Control Utility manages batch processing, it generates a date-stamped log in the `<OFSAAI Installed Directory>/database/db_tools/logs/batch_control.log` file. The log file contains relevant information such as status of various batch control processes, results, and error records.

You can modify the current logging configuration for the Alert Purge Utility in the `<OFSAAI Installed Directory>/database/db_tools/log4j2.xml files`. For more information about logging in these configuration files, refer to Managing Common Resources for Batch Processing Utilities, and Appendix A, Logging, for more information.

6.1.4 Using the Batch Control Utility

The Batch Control Utility typically runs as part of automated processing that a job scheduling tool such as Maestro or Unicenter AutoSys controls. The utility starts and terminates through a shell script, using values in parameters that particular configuration files contain.

You can use the Batch Control Utility to run the following types of batches:

- **End-of-day**: Represent processing at the completion of a business day for a set of data. Some processes are only appropriate for end-of-day batches. For example, daily activity summary derivations and calculating event ages are activities that occur only in end-of-day batches. Multiple end-of-day batches per day can run if the Behavior Detection installation supports multiple time zones , such as New York and Singapore.
- **Intra-day**: Used when loading data between end-of-day batches to obtain more frequent detection results. For example, running a batch of trading-compliance scenarios at 10:00 A.M.

can identify behaviors relevant to the opening of the market without waiting for the end of the day to be able to act.

The following sections describe this process, including tasks that you can perform when configuring the utility or running it manually (that is, starting, stopping, or obtaining a batch name).

- Configuring the Batch Control Utility
- Setting Up Batches
- Starting a Batch Process Manually
- Processing for Batch Start
- Ending a Batch Process
- Processing for End Batch
- Identifying a Running Batch Process
- Obtaining a Batch Name

6.1.4.1 Configuring the Batch Control Utility

To configure the batch control utility, follow these steps:

1. Navigate to the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file.This file contains common configuration information that Batch Control and other utilities require for processing (see Figure 37).
2. Use the following sample section from the `install.cfg` file to input configuration information specific to this utility, including the single parameter that batch control requires.

```
############### BATCH CONTROL CONFIGURATION
####################


# When ending the batch, age events in calendar
or business days.

age.events.useBusinessDays=Y
```

Figure 43: Configuring Batch Control Utility

The value of the `age.events.useBusinessDays` parameter indicates that at completion of an end-of-day batch process, the Behavior Detection application calculates the age of active events by number of calendar days (N) or business days (Y). The value of this parameter resides in the `KDD_CAL` table (refer to Table 32, for more information).

The utility connects to the database employing the user that the `utils.database.username` property specifies in the `install.cfg` file.

6.1.4.2 Setting Up Batches

TBAML delivers with a default batch called DLY. The `KDD_PRCSNG_BATCH` table includes this batch and must contain all batches in the system. When a batch starts as part of an automated process, it uses the batch names and other start-up information in this table. The DLY processing batch with ALL as the source origin is reserved for instances where one batch load is required, ignoring source systems. If you wish to associate specific source systems to DLY, then the DLY/ALL record must be deleted from the `KDD_PRCSNG_BATCH_SRC` table.

The following table provides the contents of the KDD_PRCSNG_BATCH table.

Table 24: KDD_PRCSNG_BATCH Table Contents

Column Name	Description
PRCSNG_BATCH_NM	Name of the batch , such as DLY.
PRCSNG_BATCH_DSPLY_NM	Readable name for the batch, such as Daily.
PRCSNG_ORDER	Relative order of a batch run within processing.
EOD_BATCH_NM	Name of the batch that is this batch's end-of-day. This name is the same as the name for PRCSNG_BATCH_NM if the row represents an end-of-day batch.
PRCSNG_BATCH_NM	Description of this processing batch.

Each row in the KDD_PRCSNG_BATCH table represents a batch. Each batch identifies the batch that is the corresponding end-of day batch. The following examples illustrate this concept:

- Single Batch
- Single Site Intra-day Processing
- Multiple Countries

6.1.4.2.1 Single Batch

In this example, the KDD_PRCSNG_BATCH table contains a single batch per day. This is typical of deployment of a single geography for which a solution set does not require detection more than once daily. The KDD_PRCSNG_BATCH table may look similar to the example in Table 25.

Table 25: Sample KDD_PRCSNG_BATCH Table with Single Batch

PRCSNG_BATCH_NM	PRCSNG_BATCH_DSPLY_NM	PRCSNG_ORDER	EOD_BATCH_NM
DLY	Daily Batch	1	DLY

6.1.4.2.2 Single Site Intra-day Processing

In this intra-day batch example, the system is servicing a single time zone but runs an additional batch during the day to identify behaviors related to overnight trading, as Table 26 describes.

Table 26: Sample KDD_PRCSNG_BATCH Table with Intra-day Processing

PRCSNG_BATCH_NM	PRCSNG_BATCH_DSPLY_NM	PRCSNG_ORDER	EOD_BATCH_NM
MAIN	Main Evening Batch	2	MAIN
MORN	Morning Batch	1	MORN

In this configuration, run the Calendar Manager Utility only during the MORN batch. Refer to Managing Calendar Manager Utility., for more information. You can run the Data Retention Manager either in the MORN or MAIN batch. If you run it in the MAIN batch, define at least one *buffer* partition so that the MORN batch does not fail due to inadequate partitions.

Refer to Managing Data Retention Manager, for more information.

6.1.4.2.3 Multiple Countries

As an Oracle client loading data through CSA, the system groups various source systems into one processing batch, so that it can call upon a specific batch and load data from specific source systems within that batch. This allows the handling of different batch loads from different countries running on the same staging instance. The association of the source systems to processing batch are captured in the KDD_PRCSNG_BATCH_SRC FCDM table. The following columns are available in this table:

Table 27: KDD_PRCSNG_BATCH_SRC FCDM Columns

Column	Data Type	Null	Primary Key	Default Value
PRCSNG_BATCH_NM	VARCHAR2(20)	Not Null	Yes	DLY To load only the US source for a batch, for example, Batch1, another record, Batch1, needs to be added.
SRC_ORIGIN	VARCHAR2(3)	Not Null	Yes	ALL To load only the US source for a batch, for example, Batch1, another record, US, needs to be added.
SRC_DESC	VARCHAR2(255)	Null	No	Productized Daily Processing Batch for all Source Systems

If you want to load only the US source for a batch, for example, Batch1, then another record, US Source System Load, needs to be added.

A single deployment supports detection against data from New York, London, and Hong Kong. In this case, three batches are all end-of-day batches, as Table 28 describes.

Table 28: Sample `KDD_PRCSNG_BATCH` Table with Multiple Country Processing

PRCSNG_BATCH_NM	PRCSNG_BATCH_DSPLY_NM	PRCSNG_ORDER	EOD_BATCH_NM
HK	Hong Kong	1	HK
LND	London	2	LND
NY	New York	3	NY

Since Hong Kong's markets open first, this is the first batch. You should run the Calendar Manager and Data Retention Manager at the start of the HK batch.

Upon setup of the batches, Behavior Detection processing begins with the `start_mantas_batch.sh` shell script. The final step in a batch is calling the `end_mantas_batch.sh` shell script.

6.1.4.3 Starting a Batch Process Manually

To start a batch manually, follow these steps:

1. Verify that the TBAML database is operational:

   ```
   tnsping <database instance name>
   ```

2. Verify that the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` configuration file contains the correct source database connection information.

3. Access the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

4. Run the batch control shell script:

   `start_mantas_batch.sh <batch name>`

   where `<batch name>` is the name of the batch. This parameter is case-sensitive.

> **NOTE** If you enter an invalid batch name, the utility terminates and logs a message that describes the error. The error message appears on the console only if you have output to the console enabled in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/categories.cfg` file.

6.1.4.4 Processing for Batch Start

After establishing the required Java environment and initiating various Java processing activities, the Batch Control Utility does the following:

1. The utility verifies that the provided batch name contains only the characters A-Z, a-z, and 0-9 by querying the `KDD_PRCSNG_BATCH` table (Table 28).
2. The utility determines whether a batch is running by querying the `KDD_PRCSNG_BATCH_CONTROL` table. The following table describes the `KDD_PRCSNG_BATCH_CONTROL` table.

Table 29: `KDD_PRCSNG_BATCH_CONTROL` Table Contents

Column Name	Description
PRCSNG_BATCH_ID	Current batch process ID.
PRCSNG_BATCH_NM	Name of the current batch process.
DATA_DUMP_DT	Current business day. The Calendar Manager Utility places this information in the table.
EOD_PRCSNG_BATCH_FL	Flag that indicates whether the batch is an end-of-day process (Y or N).

3. The utility records information about the batch in the `KDD_PRCSNG_BATCH_HIST` table. This table contains a history of all batches that appear by start date and end date.

   The following table describes the `KDD_PRCSNG_BATCH_HIST` table.

Table 30: `KDD_PRCSNG_BATCH_HIST` Table Contents

Column Name	Description
PRCSNG_BATCH_ID	Current batch process ID.
PRCSNG_BATCH_NM	Name of the current batch process.
DATA_DUMP_DT	Business day on which the batch ran.
START_TS	Time that the batch started.
END_TS	Time that the batch ended (if applicable).
STATUS_CD	Status code that indicates whether the batch is currently running (*RUN*) or has finished (*FIN*).

4. The Batch Control Utility logs a message in the `<OFSAAI Installed Directory>/ database/db_tools/logs/batch_control.log` file, stating that the batch process has begun.

Querying the KDD_PRCSNG_BATCH_HIST table for confirmation that the batch has started displays information similar to that in Figure 44. In the last entry, note the appearance of RUN for STATUS_CD and lack of end time in END_TS.

PRCSNG_BATCH_ID	PRCSNG_BATCH_NM	DATA_DUMP_DT	START_TS	END_TS	STATUS_CD
1	DLY	10-Nov-06	11-Nov-06 6:45:32 AM	11-Nov-06 7:32:56 AM	FIN
2	DLY	11-Nov-06	12-Nov-06 7:54:45 AM	12-Nov-06 8:23:12 AM	FIN
3	DLY	12-Nov-06	13-Nov-06 6:12:32 AM	13-Nov-06 7:23:20 AM	FIN
4	DLY	13-Nov-06	14-Nov-06 6:23:49 AM	14-Nov-06 7:10:45 AM	FIN
5	DLY	14-Nov-06	15-Nov-06 6:25:32 AM	15-Nov-06 7:12:56 AM	FIN
6	DLY	15-Nov-06	16-Nov-06 6:34:37 AM	16-Nov-06 7:56:32 AM	FIN
7	DLY	16-Nov-06	17-Nov-06 6:21:34 AM	17-Nov-06 7:48:26 AM	FIN
8	DLY	17-Nov-06	18-Nov-06 6:11:23 AM	18-Nov-06 7:13:56 AM	FIN
9	DLY	18-Nov-06	19-Nov-06 6:34:36 AM	19-Nov-06 7:45:56 AM	FIN
10	DLY	19-Nov-06	20-Nov-06 6:39:35 AM	20-Nov-06 7:32:56 AM	FIN
11	DLY	20-Nov-06	21-Nov-06 6:35:32 AM		RUN

Figure 44: Sample KDD_PRCSNG_BATCH_HIST Table—Batch Start Status

6.1.4.5 Ending a Batch Process

When a batch ends as part of an automated process, the utility retrieves the batch name and other information from the KDD_PRCSNG_BATCH table (refer to Table 24).To stop a batch process manually, follow these steps:

1. Verify that the TBAML database is operational.

   ```
   tnsping <database instance name>
   ```

2. Verify that the <OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg configuration file contains the correct source database connection information.

3. Access the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

4. Start the batch shell script:

   ```
   end_mantas_batch.sh
   ```

   If you enter an invalid batch name, the utility terminates and logs a message that describes the error. The error message appears on the console only if you have output to the console enabled in the <OFSAAI Installed Directory>/database/db_tools/mantas_cfg/categories.cfg configuration file.

6.1.4.6 Processing for End Batch

After establishing the required Java environment and initiating various Java processing activities, the Batch Control Utility does the following:

1. Determines whether a batch is running by querying the KDD_PRCSNG_BATCH_CONTROL table (refer to Table 29).

2. Records information about the batch in the KDD_PRCSNG_BATCH_ HIST table (refer to Table 30). This table contains a history of all batches that appear by start date and end date. Figure 45 illustrates a sample table query; an end time-stamp in END_TS and status of FIN in STATUS_CD for the bolded entry indicates that the batch has ended.

PRCSNG_BATCH_ID	PRCSNG_BATCH_NM	DATA_DUMP_DT	START_TS		END_TS		STATUS_CD
1	DLY	10-Nov-06	11-Nov-06	6:45:32 AM	11-Nov-06	7:32:56 AM	FIN
2	DLY	11-Nov-06	12-Nov-06	7:54:45 AM	12-Nov-06	8:23:12 AM	FIN
3	DLY	12-Nov-06	13-Nov-06	6:12:32 AM	13-Nov-06	7:23:20 AM	FIN
4	DLY	13-Nov-06	14-Nov-06	6:23:49 AM	14-Nov-06	7:10:45 AM	FIN
5	DLY	14-Nov-06	15-Nov-06	6:25:32 AM	15-Nov-06	7:12:56 AM	FIN
6	DLY	15-Nov-06	16-Nov-06	6:34:37 AM	16-Nov-06	7:56:32 AM	FIN
7	DLY	16-Nov-06	17-Nov-06	6:21:34 AM	17-Nov-06	7:48:26 AM	FIN
8	DLY	17-Nov-06	18-Nov-06	6:11:23 AM	18-Nov-06	7:13:56 AM	FIN
9	DLY	18-Nov-06	19-Nov-06	6:34:36 AM	19-Nov-06	7:45:56 AM	FIN
10	DLY	19-Nov-06	20-Nov-06	6:39:35 AM	20-Nov-06	7:32:56 AM	FIN
11	DLY	20-Nov-06	21-Nov-06	6:35:32 AM	21-Nov-06	7:39:32 AM	FIN

Figure 45: Sample KDD_PRCSNG_BATCH_HIST Table—Batch End Status

3. Calculates the age of all open events and writes it to `KDD_REVIEW.AGE` if the `EOD_BATCH_FL` is `Y` in the `KDD_PRCSNG_BATCH_CONTROL` table.
4. Updates the `KDD_REVIEW` table for all events from the current batch to set the Processing Complete flag to `Y`. This makes the events available for event management.
5. Deletes any records in the `KDD_DOC` table that the system marks as temporary and are older than 24 hours.
6. Logs a message in the `<OFSAAI Installed Directory>/database/db_tools/logs/batch_control.log` file, stating that the end batch process has begun.

6.1.4.7 Identifying a Running Batch Process

ATTENTION At times, you may must know the name of a currently running batch, or verify that a batch is active. For example, during intra-day detection processing, many batches may be running simultaneously and you must identify one or more by name. If you set the batch control logging to display at the console, be aware that log messages are mixed with the output of the shell script; the output can be difficult to read.

6.1.4.7.1 To Obtain a Batch Name

To identify a running batch process, follow these steps:

1. Access the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

2. Start the batch shell script:

   ```
   get_mantas_batch.sh
   ```

   The name of the currently running batch is written to standard output.

6.1.4.8 Obtaining a Batch Name

After establishing the required Java environment and initiating various Java processing activities, the Batch Control Utility retrieves the name of the currently running batch from the `KDD_PRCSNG_BATCH_CONTROL` table (refer to Table 29).

The utility returns the batch name to standard output.

6.2 Managing Calendar Manager Utility.

After loading holidays into the KDD_CAL_HOLIDAY table and weekly off-days into the KDD_CAL_WKLY_OFF table, you can use the Calendar Manager Utility to update and manage Oracle system calendars. The <OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg configuration file contains modifiable inputs that you use to run the utility (refer to Install Configuration for more information).

This section contains the following topics:

- Directory Structure
- Logs
- Calendar Information
- Using the Calendar Manager Utility

6.2.1 Directory Structure

The following table provides the directory structure for the Calendar Manager Utility in <OFSAAI Installed Directory>/database/db_tools/.

Table 31: Calendar Manager Utility Directory Structure

Directory	Description
bin/	Contains executable files, including the shell script set_mantas_date.sh.
lib/	Includes required class files in .jar format.
mantas_cfg/	Contains configuration files, such as install.cfg and categories.cfg, in which you can configure properties and logging attributes.
logs/	Keeps the calendar_manager.log log file that the utility generates during execution.

6.2.2 Logs

As the utility updates the calendars in the TBAML system, it generates a log that it enters in the <OFSAAI Installed Directory>/database/db_tools/logs/calendar_manager.log file (the logging process time-stamps all entries). The log file contains relevant information such as status of the various Calendar Manager processes, results, and error records.

You can modify the current logging configuration for the Alert Purge Utility in the <OFSAAI Installed Directory>/database/db_tools/log4j2.xml files. For more information about logging in these configuration files, refer to Managing Common Resources for Batch Processing Utilities, and Appendix A, Logging, for more information.

6.2.3 Calendar Information

The Calendar Manager Utility obtains all holidays and weekly off-days for loading into the OFSBD calendars by retrieving information from the KDD_CAL_HOLIDAY and KDD_CAL_WKLY_OFF tables (refer to Table 18 and Table 19). These tables contain calendar information that an Oracle client has provided regarding observed holidays and non-business days.

6.2.4 Using the Calendar Manager Utility

The Calendar Manager Utility runs as part of automated processing that a job scheduling tool such as Maestro or Unicenter AutoSys controls. The utility runs through a shell script, using values in parameters that the `install.cfg` file contains. The utility then populates the `KDD_CAL` database table with relevant OFSBD business calendar information.

The following sections describe this process, including tasks that you can perform when configuring the utility or running it manually.

- Configuring the Calendar Manager Utility
- Executing the Calendar Manager Utility
- Updating the `KDD_CAL` Table

6.2.4.1 Configuring the Calendar Manager Utility

The `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file contains common configuration information that Calendar Manager and other utilities require for processing (refer to Figure 37). The following sample section from the `install.cfg` file provides configuration information specific to this utility, including default numerical values in the utility's two required parameters.

```
################ CALENDAR MANAGER CONFIGURATION
##################


# The look back and look forward days of the
provided date.

# These values are required to update the KDD_CAL
table. The

# maximum look back or forward is 999 days.

calendar.lookBack=365

calendar.lookForward=10
```

- `calendar.lookBack`: Determines how many days to iterate backward from the provided date during a calendar update.
- `calendar.lookForward`: Determines how many days to iterate forward from the provided date during a calendar update.

The maximum value that you can specify for either of these parameters is 999 days.

> **NOTE** The lookback period should be at least 90 days and as long as any events are likely to be open. The lookforward period does not must be more than 10 days. This is used when calculating projected settlement dates during data management.

WARNING When you have configured the system to calculate event and case age in Business Days, the calendar date of the current system date and the calendar date of the event or case creation must be included in the calendar. As such, if you are running with a business date that is substantially behind the current system date, you should set the `lookForward` parameter for the calendar manager sufficiently high to ensure that the system date is included on the calendar. Additionally, if you have events that are open for a very long period, you should set the `lookBack` parameter sufficiently high to include the dates of your oldest open events. If the business calendar does not cover either of these dates, the processing reverts to calculating age in Calendar days.

The utility connects to the database employing the user that the `utils.database.username` property specifies in the `install.cfg` file.

6.2.4.2 Executing the Calendar Manager Utility

You can manage the Calendar Manager Utility as part of automated processing. You can run the utility either inside a batch process (that is, after calling the `start_mantas_batch.sh` script) or outside a batch.

6.2.4.2.1 Starting the Utility Manually

To start the Calendar Manager Utility, follow these steps:

1. Verify that the TBAML database is operational:

   ```
   tnsping <database instance name>
   ```

2. Verify that the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` configuration file contains the correct source database connection information.

3. Go to the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

4. Start the calendar manager shell script:

   ```
   set_mantas_date.sh YYYYMMDD
   ```

   where `YYYYMMDD` is the date on which you want to base the calendar , such as *`20161130`* for November 30, 2016. The utility then verifies that the entered date is valid and appears in the correct format.

   If you do not enter a date or enter it incorrectly, the utility terminates and logs a message that describes the error. The error message displays on the console only if you have output to the console enabled in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/categories.cfg` configuration file.

6.2.4.3 Updating the KDD_CAL Table

The Calendar Manager Utility retrieves information that it needs for updating business calendars from the `KDD_CAL_HOLIDAY` and `KDD_CAL_WKLY_OFF` database tables. It then populates the `KDD_CAL` table accordingly. That is, for each calendar name found in the `KDD_CAL_WKLY_OFF` and `KDD_CAL_HOLIDAY` tables, the utility creates entries in `KDD_CAL`.

The following table provides the contents of the KDD_CAL table.

Table 32: KDD_CAL Table Contents

Column Name	Description
CLNDR_NM	Specific calendar name.
CLNDR_DT	Date in the range between the lookback and lookforward periods.
CLNDR_DAY_AGE	Number of calendar days ahead or behind the provided date. The provided date has age 0, the day before is 1, the day after is –1. For example, if a specified date is 20061129, the CLNDR_DAY_AGE of 20061128 = 1, and 20061130 = –1.
BUS_DAY_FL	Flag that indicates whether the specified date is a valid business day (set the flag to Y). Set this flag to N if the DAY_OF_WK column contains an entry that appears as a valid non-business day in the KDD_CAL_WKLY_OFF table, or a valid holiday in KDD_CAL_HOLIDAY.
BUS_DAY_AGE	Number of business days ahead or behind the provided date. If BUS_DAY_FL is N, BUS_DAY_AGE receives the value of the previous day's BUS_DAY_AGE.
DAY_OF_WK	Value that represents the day of the week: Sunday=1, Monday=2, Tuesday=3, ... Saturday=7.
WK_BNDRY_CD	Week's start day (SD) and end day (ED). ● If this is the last business day for this calendar name for the week (that is, next business day has a lower DAY_OF_WK value), set to ED<x>, where <x> is a numeric counter with the start/end of the week that the provided date is in = 0. ● If it is the first business day for this calendar name for this week (that is, previous business day has a higher DAY_OF_WK value), set to SD<x>. Weeks before the provided date increment the counter, and weeks after the provided date decrement the counter. Therefore, "ED0" is always on the provided date or in the future, and "SD0" is always on the provided date or in the past.
MNTH_BNDRY_CD	Month's start day (SD) and end day (ED). ● If this is the last business day for this calendar name for the month (that is, next business day in a different month), set to ED<y>, where *y* is a numeric counter with the start/end of the month that the provided date is in = 0. ● If it is the first business day for this calendar for this month (that is, previous business day in a different month), set to SD<y>. Months before the provided date increment the counter, and months after the provided date decrement the counter. Therefore, "ED0" is always on the provided date or in the future, and "SD0" is always on the provided date or in the past.

Table 32: KDD_CAL Table Contents (Continued)

Column Name	Description
BUS_DAY_TYPE_ CD	Indicates the type of business day: • N = Normal • C = Closed • S = Shortened
SESSN_OPN_TM	Indicates the opening time of the trading session for a shortened day. The format is HHMM.
SESSN_CLS_TM	Indicates the closing time of the trading session for a shortened day. The format is HHMM.
SESSN_TM_OFFST_TX	Indicates the timezone offset for SESSN_OPN_TM and SESSN_CLS_TM. The format is HH:MM.
QRTR_BNDRY_CD	Quarter's start day (SD) and end day (ED). • If this is the last business day for this calendar name for the quarter (that is, next business day in a different quarter), set ED to <y>, where y is a numeric counter with the start/end of the quarter that the provided date is in = 0. • If it is the first business day for this calendar name for this quarter (that is, previous business day is in a different quarter), set SD to <y>. Quarters before the provided date increment the counter, and quarters after the provided date decrement the counter. Therefore, "ED0" is always on the provided date or in the future, and "SD0" is always on the provided date or in the past.

If a batch is running, the system uses the date provided in the call to start the `set_mantas_date.sh` script. This script updates the `KDD_PRCSNG_BATCH_CONTROL.DATA_DUMP_DT` field.

6.2.4.4 Configuring Case Age

Case age can be calculated based on Business Days or Calendar Days by updating the configurable parameter set in the Installation Parameter table, from the Manage Parameters screen. (Refer to the Configuration Guide for more information).

To execute the parameter, use the following command:

```
run_caseage_calc.sh
```

This will update the `KDD_CASES.age` column with age of the case, calculated in business days or calendar days based on the configuration made in the Installation Parameter table.

6.3 Managing Data Retention Manager

TBAML relies on Oracle partitioning for maintaining data for a desired retention period, providing performance benefits, and purging older data from the database. The data retention period for business and market data is configurable. Range partitioning of the tables is by date.

The Data Retention Manager enables you to manage Oracle database partitions and indexes on a daily, weekly, and/or monthly basis (refer to Figure 36). This utility allows special processing for trade-related database tables to maintain open order, execution, and trade data prior to dropping old partitions. As administrator, you can customize these tables.

The utility accommodates daily, weekly, and monthly partitioning schemes. It also processes specially configured Mixed Date partitioned tables. The Mixed Date tables include partitions for Current Day, Previous Day, Last Day of Week for weeks between Current Day and Last Day of Previous Month, and Last Business Day of Previous Two Months.

The Data Retention Manager can:

- Perform any necessary database maintenance activities, such as rebuilding global indexes.
- Add and drop partitions, or both, to or from the date-partitioned tables.

Data Retention Manager provides a set of SQL procedures and process tables in the Behavior Detection database. A shell script and a configuration file that contain the various inputs set the environment that the utility uses.

This section covers the following topics:

6.3.1 Directory Structure

The following table provides the directory structure for the Data Retention Manager.

Table 33: Data Retention Manager Directory Structure

Directory	Contents
bin/	Executable files, including the `run_drm_utility.sh` shell script.
lib/	Required class files in `.jar` format.
mantas_cfg/	Configuration files , such as `install.cfg` and `categories.cfg`, in which you can configure properties and logging attributes.
logs/	File `<OFSAAI Installed Directory>/database/db_tools/logs/DRM_Utility.log` that the utility generates during execution.

6.3.2 Logs

Oracle stored procedures implement Data Retention Manager and conducts some logging on the database server. A configuration parameter in the `install.cfg` file controls the path to which you store the logs on the database server.

As the Data Retention Manager performs partitioning and indexing activities, it generates a log that it enters in the `<OFSAAI Installed Directory>/database/db_tools/logs/DRM_Utility.log` file (the logging process time-stamps all entries). The log file contains relevant information such as status of the various processes, results, and error records.

You can modify the current logging configuration for the Alert Purge Utility in the `<OFSAAI Installed Directory>/database/db_tools/log4j2.xml files`. For more information about

logging in these configuration files, refer to Managing Common Resources for Batch Processing Utilities, and Appendix A, Logging, for more information.

6.3.3 Processing Flow

Figure 46 illustrates the Data Retention Manager's process flow for daily, weekly, and monthly partitioning. Based on a table's retention period, the utility drops the oldest partition and then adds a new partition.

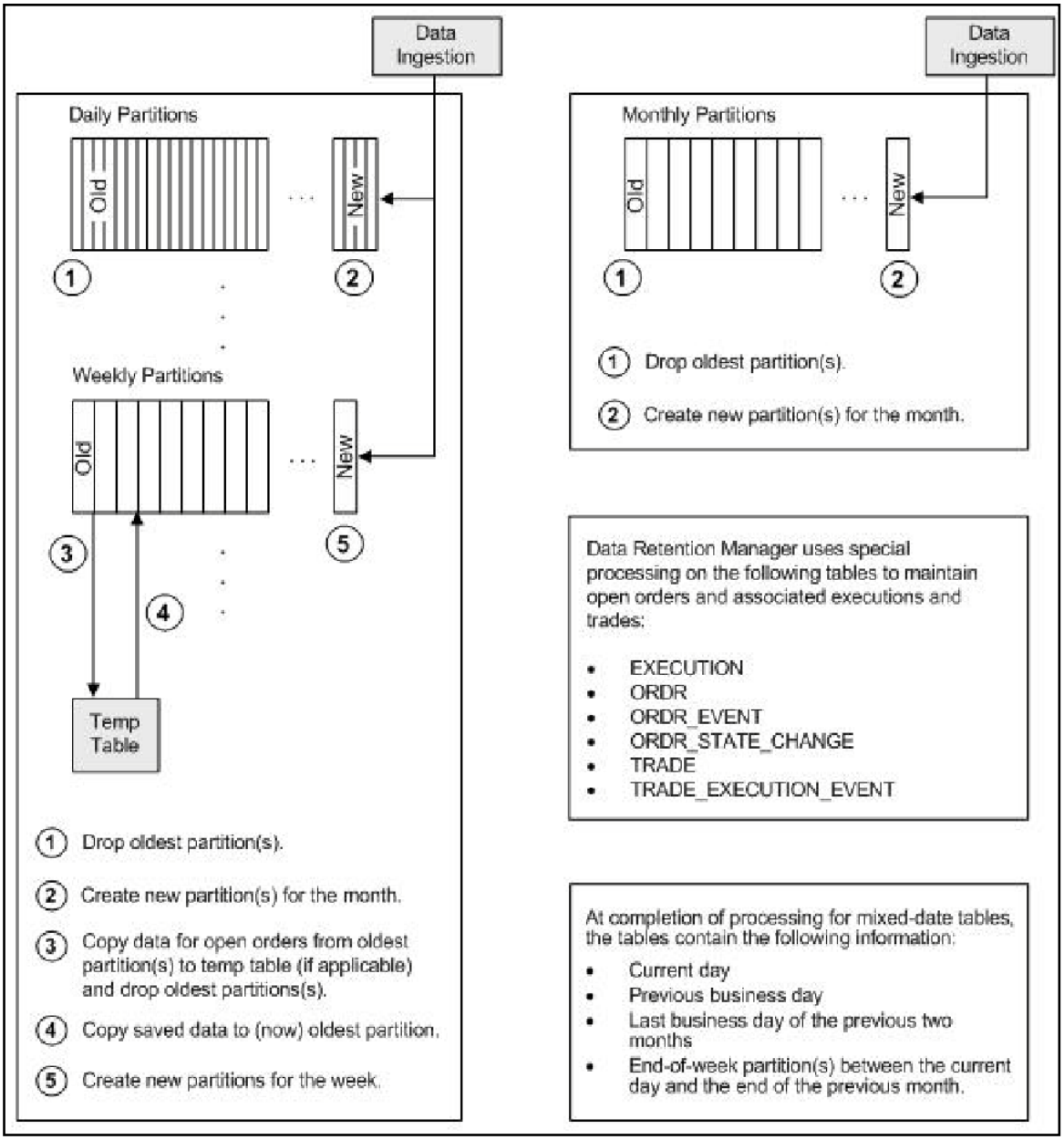


Figure 46: Database Partitioning Process

6.3.4 Using the Data Retention Manager

The Data Retention Manager typically runs as part of automated processing that a job scheduling tool such as Maestro or Unicenter AutoSys controls. However, you can run Data Retention Manager manually on a daily, weekly, or monthly basis to manage database tables.

The following sections describe how to configure and execute the utility and maintain database partitions and indexes.

- Configuring the Data Retention Manager
- Executing the Data Retention Manager
- Creating Partitions
- Maintaining Partitions
- Maintaining Indexes

6.3.4.1 Configuring the Data Retention Manager

To configure the Data Retention Manager, follow these steps:

1. Navigate to the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file.This file contains common configuration information that Data Retention Manager and other utilities require for processing
2. Use the sample install.cfg file in Figure 37 to do a configuration.

> **NOTE** The configuration parameters in the `install.cfg` are only used if command line parameters are not provided. It is strongly recommended that you provide command line parameters instead of using the `install.cfg` parameters.

The Data Retention Manager automatically performs system checks for any activity that may result in an error , such as insufficient space in the tablespace. If it discovers any such activity, it logs a Warning message that identifies the potential problem. If Data Retention Manager fails to run successfully, you can configure the utility so that the ingestion process for the following day still proceeds.

The following sample section from the `install.cfg` file provides other configuration information specific to this utility, including required and optional parameters.

```
######### DATA RETENTION MANAGER CONFIGURATION
##################

# Set the Data Retention Manager input variables
here.

##

drm_operation=P

drm_partition_type=A

drm_owner=${schema.mantas.owner}

drm_object_name=A

drm_weekly_proc_fl=Y
```

Figure 47: install.cfg Data Retention Manager Configuration

This example shows default values that the system uses only when calling the utility with no command line parameters. The following table describes these parameters.

Table 34: Data Retention Manager Processing Parameters

Parameter	Description
drm_operation	Operation type: • P-Partition • AM-Add Monthly Partition • DM -Drop Monthly Partition • RI - Rebuild Indexes • RV - Recompile Views • T-Truncate Current Partition
drm_partition_type	Partition type: • D-Daily • W-Weekly • M- Monthly • X- Mixed-Date • A- All Partitions (Daily, Weekly, Monthly)
drm_owner	Owner of the object (Atomic schema owner).
drm_object_name	Object name. If performing an operation on all objects, the object name is A.
drm_weekly_proc_fl	Flag that determines whether partitioning occurs weekly (Y and N).

NOTE The system processes Daily partitioned tables (`drm_partition_type=D`) and Mixed-date partitioned tables (`drm_partition_type=X`) simultaneously. Therefore, you need only specify D or X to process these tables.

An example for the Mixed-date partition, for the present date 20050711, is:

```
P20050711 (Current Day)
P20050708 (Previous Day and End of week #1)
P20050701 (End of previous week #2)
P20050630 (End of previous Month #1)
P20050624 (End of previous week #3)
P20050617 (End of previous week #4)
P20050531 (End of previous Month #2)
```

6.3.4.2 Executing the Data Retention Manager

Before you execute the Data Retention Manager, ensure that users are not working on the system. To avoid conflicts, Oracle recommends that you use this utility as part of the end-of-day activities.

The Data Retention Manager should be executed nightly for Daily partitioned and Mixed-date partitioned tables, after the calendar has been set for the next business day. For weekly and monthly partitioned tables, the Data Retention Manager should be executed prior to the end of the current processing period.

> **TIP** Oracle recommends running the Data Retention Manager on Thursday or Friday for weekly partitioned tables and on or about the 23rd of each month for monthly partitioned tables.
>
> Set the system date with the Calendar Manager Utility prior to running the Data Retention Manager (refer to Managing Calendar Manager Utility. for more information).

6.3.4.2.1 Running the Data Retention Manager

To run the Data Retention Manager manually, follow these steps:

3. Verify that the TBAML database is operational:

   ```
   tnsping <database instance name>
   ```

4. Verify that the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` configuration file contains the correct source database connection information.
5. Access the directory where the shell script resides:

   ```
   cd <OFSAAI Installed Directory>/database/db_tools/bin
   ```

6. Start the batch shell script with the parameters in Table 34:

   run_drm_utility.sh <drm_operation> <drm_partition_type> <drm_owner> <drm_object_name> <drm_weekly_proc_fl>

The following are examples of running the script:

- To run the utility for all daily tables in the ATOMIC schema, execute the script:

  run_drm_utility.sh P D BUSINESS A N

- To run the utility to drop a monthly partition of the `BUSINESS` table `ACCT_SMRY_MNTH`, execute the script as follows (using the same parameters as in the previous example):

  ```
  run_drm_utility.sh DM M BUSINESS ACCT_SMRY_MNTH N
  ```

6.3.4.3 Creating Partitions

To create partition names, use the formats in the following table:

Table 35: Partition Name Formats

Partition Type	Format and Description
Monthly	PYYYYMM where YYYY is the four-digit year and MM is the two-digit month for the data in the partition. For example: Data for November 2006 resides in partition P200611. **Note:** The Data Retention Manager uses information in the `KDD_CAL` table to determine end-of-week and end-of-month boundary dates.

Table 35: Partition Name Formats

Partition Type	Format and Description
Weekly or Daily	PYYYYMMDD where YYYY is the four-digit year, MM is the two-digit month, and DD is either the date of the data (daily) or the date of the following Friday (weekly) for the data in the partition. For example: Data for November 30, 2006 resides in partition P20061130. Data for the week of November 19 - November 23, 2006 resides in partition P20061123. **Note:** The Data Retention Manager uses information in the `KDD_CAL` table to determine end-of-week and end-of-month boundary dates.

NOTE Data Retention Manager assesses the current status of partitions on the specified table to determine the requested partition. If the system previously fulfilled the request, it logs a warning message.

The Data Retention Manager does not support multiple partition types on a single table. If an Oracle client wants to alter the partitioning scheme on a table, that client must rebuild the table using the new partitioning scheme prior to utilizing the Data Retention Manager. Then you can update the values in the Data Retention Manager tables to reflect the new partitioning scheme.

6.3.4.4 Maintaining Partitions

Partition maintenance procedures remove old data from the database so that the database does not continue to grow until space is insufficient. Daily, weekly, or monthly maintenance is necessary for tables that have daily, weekly, and monthly partitions, respectively.

To maintain Partitions, follow these steps:

1. Copy information related to open orders from the oldest partitions to temp tables `(EXECUTION, ORDR, ORDR_EVENT, ORDR_STATE_CHANGE TRADE and TRADE_EXECUTION_EVENT)`
2. Drop the oldest partitions for all partition types.
3. Insert the saved data into what is now the oldest partition (applicable to tables with open orders).
4. Create new partitions.
5. Recompile the views that scenarios use.

6.3.4.4.1 Managing Daily Partitioning Alternative

The Data Retention Manager also enables you to build five daily partitions on a weekly basis. To build partitions, follow these steps:

1. Execute the `run_drm_utility.sh` shell script
2. Set the `drm_weekly_proc_flg` parameter to Y.For more information, refer to Table 34.

This procedure eliminates the must perform frequent index maintenance; Oracle recommends doing this for large market tables.

This approach builds the daily partitions for the next week. When creating the five daily partitions on a weekly basis, the Data Retention Manager should be executed prior to the end of the current week, to create partitions for the next week.

> **NOTE** You must set the `WEEKLY_ADD_FL` parameter in the `KDD_DR_MAINT_OPRTN` table to `Y` so that the procedure works correctly. For more information about this parameter, refer to Table 36, for more information.

6.3.4.4.2 Partition Structures

The structures of business data partitions and market data partitions differ in the following ways:

- Business data partitions are pre-defined so that weekdays (Monday through Friday) are business days, and Saturday and Sunday are *weekly off-days*. Business data tables use all partitioning types.

  You can use the Calendar Manager Utility to configure a business calendar as desired. For more information about this utility, refer to Managing Calendar Manager Utility., for more information.

- Market data partitions hold a single day of data. The partitions use the `PYYYYMMDD` convention, where `YYYYMMDD` is the date of the partition.

6.3.4.4.3 Recommended Partition Maintenance

You should run partition maintenance as appropriate for your solution set. Oracle recommends that you run partition maintenance for AML on a daily basis (after setting the business date through the Calendar Manager Utility, and prior to the daily execution of batch processing), and Trading Compliance at least once a week.

Oracle recommends that you use the P (Partition) option when running the Data Retention Manager, as it drops older partitions and adds appropriate partitions in a single run of the utility.

When performing monthly maintenance, you can add or drop a partition independently, as the following procedures describe.

> **NOTE** If you ingest data belonging to a date less than the current date, you should run the DRM utility till current date. This avoids the error *Partition Not Found* while accessing trade records in Trade Blotter UI.

6.3.4.4.4 Managing Alternative Monthly Partition

As part of an alternative method of monthly partition maintenance, you can either add or drop a monthly database partition. as described in the following section:

6.3.4.4.4.1 Adding a Monthly Database Partition

To add a monthly partition, run the utility's shell script as follows (refer to Table 34 for parameters):

```
run_drm_utility.sh AM M BUSINESS <object> N
```

where `AM` is the `drm_operation` parameter that implies adding a monthly partition.

6.3.4.4.4.2 Dropping a Monthly Database Partition

To drop a monthly partition, run the utility's shell script as follows (refer to Table 34 for parameters):

run_drm_utility.sh DM M BUSINESS <object> N

where, `DM` is the `drm_operation` parameter that implies dropping a partition.

6.3.4.5 Maintaining Indexes

As part of processing, the Data Retention Manager automatically rebuilds the database index and index partitions that become unusable. You do not need to maintain the indexes separately.

The utility enables you to rebuild global indexes by executing the following command:

```
run_drm_utility.sh RI M BUSINESS <object> N
```

where RI is the `drm_operation` parameter that implies rebuilding indexes.

6.3.5 Utility Work Tables

The Data Retention Manager uses the following work tables during database partitioning:

- KDD_DR_MAINT_OPRTN Table
- KDD_DR_JOB Table
- KDD_DR_RUN Table

6.3.5.1 KDD_DR_MAINT_OPRTN Table

The `KDD_DR_MAINT_OPRTN` table contains the processing information that manages Data Retention Manager activities. The following table provides these details.

Table 36: BUSINESS.KDD_DR_MAINT_OPRTN Table Contents

Column Name	Description
PROC_ID	Identifies the sequence ID for the operation to perform.
ACTN_TYPE_CD	Indicates the activity that the utility is to perform on the table: • A: Analyze • RI: Rebuild Indexes • P: Partition • RV: Recompile Views
OWNER	Identifies an owner or user of the utility.
TABLE_NM	Identifies a database table.
PARTN_TYPE_CD	Indicates the partition type: • D: Daily • W: Weekly • M: Monthly • X: Mixed Date
TOTAL_PARTN_CT	Specifies the total number of partitions to be created, including the current partition. For example, for a daily partitioning scheme of four previous days and the current day, the value of this field is five (5).

Table 36: BUSINESS.KDD_DR_MAINT_OPRTN Table Contents (Continued)

Column Name	Description
BUFFER_PARTN_CT	Specifies the number of buffer partitions the utility is to maintain, excluding the current partition. For example, a two-day buffer has a value of two (2).
CNSTR_ACTN_FL	Determines whether to enable or disable constraints on the table during processing.
WEEKLY_ADD_FL	Indicates whether daily partitions are added for a week at a time. If set to Y, creates Daily Partitions for the next week. For example, if run on a Thursday, the DRM creates the five (5) partitions for the next week beginning with Monday.
NEXT_PARTN_DATE	Indicates starting date of the next partition that may get created, based on the current partitioned date.

ATTENTION For weekly partitioned tables, do not set the value to Y.

6.3.5.2 KDD_DR_JOB Table

The `KDD_DR_JOB` table stores the start and end date and time and the status of each process that the Data Retention Manager calls. The following table provides these details.

Table 37: BUSINESS.KDD_DR_JOB Table Contents

Column Name	Description
JOB_ID	Unique sequence ID.
START_DT	Start date of the process.
END_DT	End date of the process.
STATUS_CD	Status of the process: • RUN: Running • FIN: Finished successfully • ERR: An error occurred • WRN: Finished with a warning

6.3.5.3 KDD_DR_RUN Table

The `KDD_DR_RUN` table stores the start and end date and time and status of individual process runs that are associated with a table. The following table provides these details.

Table 38: BUSINESS.KDD_DR_RUN Table Contents

Column Name	Description
JOB_ID	Unique sequence ID.
PROC_ID	Process ID.
START_DT	Start date of the process.
END_DT	End date of the process.

Table 38: BUSINESS.KDD_DR_RUN Table Contents (Continued)

Column Name	Description
RSULT_CD	Result of the process: • FIN: Finished successfully • ERR: An error occurred • WRN: Finished with a warning
ERROR_DESC_TX	Description of a resulting error or warning.

The system also uses the KDD_CAL table to obtain information such as the dates of the last-day-of-previous-month and end-of-weeks. Refer to Table 32 for contents of the KDD_CAL table.

6.4 Database Statistics Management

The system uses a script to manage Oracle database statistics. These statistics determine the appropriate execution path for each database query.

6.4.1 Logs

The log.category.RUN_STORED_PROCEDURE property controls logging for the process.location entry in the <OFSAAI Installed Directory>/database/db_tools/mantas_cfg/categories.cfg file.

6.4.2 Using Database Statistics Management

The system calls the script as part of nightly processing at the appropriate time and with the appropriate parameters:

- analyze_mantas.sh <analysis_type> [TABLE_NAME]

The <analysis_type> parameter can have one of the following values:

- DLY_POST_LOAD: Use this value to update statistics on tables that the system just loaded (for BUSINESS and MARKET related tables).
- ALL: Use this once per week on all schemas.
- DLY_POST_HDC: Use this value to update statistics of the event-related archived data (in _ARC tables) that the Behavior Detection UI uses to display events. It is recommended that you do not modify this table. The Behavior Detection Historical Data Copy procedures uses this table to archive event-related data.
- DLY_PRE_HDC: Use this value to update statistics of the Mantas related tables that contain the event-related information. It is recommended that you do not modify this table. The Historical Data Copy procedures uses this table to archive event-related data.
- DLY_POST_LINK: Use this value to update statistics of the Mantas related tables that contain network analysis information. Run this option at the conclusion of the network analysis batch process.

The [TABLE_NAME] parameter optionally enables you to analyze one table at a time. This allows scheduling of the batch at a more granular level, analyzing each table as processing completes instead of waiting for all tables to complete before running the analysis process.

The metadata in the KDD_ANALYZE_PARAM table drive these processes. For each table this table provides information about the method of updating the statistics that you should use for each analysis type. Valid methods include:

- EST_STATS: Performs a standard statistics estimate on the table.
- EST_PART_STATS: Estimates statistics on only the newest partition in the table.

For the EST_STATS and EST_PART_STATS parameters, the default sample size that the analyze procedure uses is now based on DBMS_STATS.AUTO_SAMPLE_SIZE.

- IMP_STATS: Imports statistics that were previously calculated. When running an ALL analysis, the system exports statistics for the tables for later use.

Failure to run the statistics estimates can result in significant database performance degradation.

These scripts connect to the database using the user that the utils.database.username property specifies, in the <OFSAAI Installed Directory>/ database/db_tools/mantas_cfg/install.cfg file. The install.cfg file also contains the following properties:

- schema.mantas.owner

The system derives schema name from this property.

For the ATOMIC Schema, there is no separate script for managing Oracle database statistics. But for improved query performance, we have to manage the Oracle database statistics periodically. Following are the sample commands.

To analyze table wise use, use the following commands:

```
ANALYZE table <Table name> compute statistics;
```

Example: `ANALYZE table KDD_CASES compute statistics;`

We can also perform whole schema analyze periodically.

6.5 Managing ETL Process for Threshold Analyzer Utility

For inserting and updating records into the KDD_TA_ML_DATA, KDD_TA_BC_DATA, and KDD_TA_TC_DATA tables, there are two shell scripts that are used to call the database procedures. These are:

- run_insert_ta_utility.sh – This script calls the P_TA_ML_INSERT_BREAKS, P_TA_BC_INSERT_BREAKS, and P_TA_TC_INSERT_BREAKS procedures, which insert data into the KDD_TA_ML_DATA, KDD_TA_BC_DATA, and KDD_TA_TC_DATA tables, respectively, based on the CREAT_TS of the events in relation to the LAST_RUN_DT from KDD_TA_LAST_RUN (values for RUN_TYPE_CD are ML_I, BC_I, and TC_I).
- run_update_ta_utility.sh – This script calls the P_TA_ML_UPDATE, P_TA_BC_UPDATE, and P_TA_TC_UPDATE procedures, which update QLTY_RTNG_CD in the KDD_TA_ML_DATA, KDD_TA_BC_DATA, and KDD_TA_TC_DATA tables, respectively, for any *Review* closed since the last run based on LAST_RUN_DT from KDD_TA_LAST_RUN (values for RUN_TYPE_CD are ML_U, BC_U, and TC_U). The CLS_CLASS_CD value from KDD_REVIEW is used as the new QLTY_RTNG_CD.

The log for these scripts is written in the run_stored_procedure.log file under the <OFSAAI Installed Directory>/database/db_tools/logs directory.

The LAST_RUN_DT column in the KDD_TA_LAST_RUN table is only updated for *inserts* and *updates* if at least one or more records were inserted or updated. The LAST_RUN_DT column is not updated for significant errors that resulted in no records being updated. These scripts are a part of the database tools and reside in the <OFSAAI Installed Directory>/database/db_tools/bin directory.

You can run this utility anytime, that is, it is not necessary to run this utility during specific processing activities.

6.5.1 Running Threshold Analyzer

To run the threshold analyzer, follow these steps:

1. Go to the ATOMIC schema and execute the following query:

```
select distinct (creat_ts)
  from kdd_review t
 where t.review_type_cd = 'AL'
   and SCNRO_DISPL_NM <> 'User Defined'
   and PRCSNG_BATCH_NM = 'DLY';
```

2. Set date as per dates returned from above SQL. Say CREATE_TS is 05/21/2013 in kdd_review table than we will set a date 05/17/2013 (Friday of last week) from the $FICHOME/database/db_tools/bin folder.
3. Execute the following command:

```
start_mantas_batch.sh DLY
set_mantas_date.sh 20130517 --(Friday of last week)
```

4. Execute DRM utility to create partitions, refer to Table 34 on page 114 for parameter values:

```
run_drm_utility.sh <Partition> <Weekly> <schema> <Table name>
<drm_weekly_proc_fl>
```

There should be different variations for each Oracle product. For example:

```
run_drm_utility.sh P W ATOMIC KDD_TA_ML_DATA N
run_drm_utility.sh P W ATOMIC KDD_TA_BC_DATA N
run_drm_utility.sh P W ATOMIC KDD_TA_TC_DATA N
```

5. Execute the following Insert and Update Threshold Analyzer scripts from $FICHOME/database/db_tools/bin folder:

```
run_insert_ta_utility.sh
run_update_ta_utility.sh
```

6. Repeat the above process if you have more than one date returned from the query in Step1.

6.6 Managing Truncate Manager

The data management subsystem calls the `run_truncate_manager.sh` script to truncate tables that require complete replacement of their data.

6.6.1 Logs

The `log.category.TRUNCATE_MANAGER.location` property in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/categories.cfg` file controls logging for this utility. The system writes log information for this process to the following location:

<OFSAAI Installed Directory>/database/db_tools/logs/truncate_manager.log

6.6.2 Using the Truncate Manager

For the `run_truncate_manager.sh` script to take the table name as an argument, the table must exist in the ATOMIC schema. The script logs into the database using the user that the `truncate.database.username` property specifies in the `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` file.

The script has the following calling signature:

```
run_truncate_manager.sh <table_name>
```

> **NOTE** This process is not intended to be called independently; only the Ingestion Manager subsystem should use it.

7 Managing Administrative Utilities

Oracle provides utilities that enable you to set up or modify a selection of database processes. This chapter focuses on the following topics:

- About Administrative Utilities
- Managing Scenario Migration Utility
- Managing the Threshold Editor
- Configuring Administration Tools

7.1 About Administrative Utilities

Several database utilities that configure and perform system pre-processing and post-processing activities are not tied to the batch process cycle:

- Managing Scenario Migration Utility**:** Extracts scenarios, datasets, networks, and associated metadata from a database to flat files and loads them into another environment.
- Managing the Threshold Editor**:** Allows you to run the same scenario multiple times against a variety of sources (for example, exchanges, currencies, or jurisdictions) with separate threshold values for each source.

7.1.1 Common Resources for Administrative Utilities

Configuration files enable the utilities to share common resources such as database configuration, directing output files, and setting up logging activities.

7.2 Managing Scenario Migration Utility

Use the Scenario Migration Utility to migrate scenarios, datasets, networks, and associated metadata from the development environment to the production environment.

To provide a list of scenarios, datasets, or networks, you edit the `scnros.cfg`, `dataset.cfg`, or the `network.cfg` files prior to scenario extraction or loading.

The Scenario Migration Utility creates and migrates the following metadata files:

- **Scenarios:** The `<scenario catalog identifier>.<scenario id>.xml` file contains scenario metadata for core Behavior Detection tables. It also may contain scenario metadata for optional tables.
- **Datasets:** The `<dataset idDS>.xml` file contains dataset metadata for core Behavior Detection tables.
- **Networks:** The `<network>NW.xml` file contains network metadata for core Behavior Detection tables.

> **NOTE** When the Scenario Migration Utility extracts these files, you can version-control them or store them in the Oracle client's archival system.

To help avoid accidental loading of a scenario into the incorrect environment, the Scenario Migration utility enables you to *name* your source and target environments. On extract, you can specify the environment name to which you plan to load the scenario. If you attempt to load it to a different environment, the system displays a warning prompt.

This section covers the following topics:

- Logs
- Using the Scenario Migration Utility
- Scenario Migration Best Practices

7.2.1 Logs

The Scenario Migration Utility produces two log files (Figure 48): `load.log` and `extract.log`. These files reside in the following location:

<OFSAAI Installed Directory>/database/db_tools/logs

7.2.2 Using the Scenario Migration Utility

This section covers the following topics, which describe configuring and executing the Scenario Migration Utility, including extracting and loading metadata:

- Configuring the Scenario Migration Utility
- Extracting Scenario Metadata
- Loading Scenario Metadata

7.2.2.1 Configuring the Scenario Migration Utility

To configure the Scenario Migration Utility, follow these steps:

Navigate to `OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg`. Theinstall.cfg file contains common configuration information that Scenario Migration and other utilities require for processing. Figure 48 provides sample information from the `install.cfg` file that is specific to this utility.

```
################ SCENARIO MIGRATION CONFIGURATION
######################

#### GENERAL SCENARIO MIGRATION SETTINGS


#Specify the flags for whether scoring rules and wrapper datasets must
be extracted or loaded

score.include=N

wrapper.include=N


#Specify the Use Code for the scenario. Possible values are 'BRK' or
'EXP'

load.scnro.use=BRK


#If custom patterns exist for a product scenario, set to 'Y' when
loading a scenario hotfix.

#This should normally be set to 'N'.
```

(Continued on next page)

```
load.ignore.custom.patterns=N

#Specify the full path of depfile and name of fixfile used for
extraction and loading

#Note : fixfile need not be specified in case of loading

sm.depfile=/scratch/ofsaaapp/OFSBD 8.0.2/OFSBD 8.0.2_B06/BDP62_B06/
database/db_tools/mantas_cfg/dep.cfg



sm.release=5.7.1


#### EXTRACT


# Specify the database details for extraction

extract.database.username=${utils.database.username}

extract.database.password=${utils.database.password}


# Specify the case schema name for both extraction and load .

caseschema.schema.owner=ATOMIC


# Specify the jdbc driver details for connecting to the source database

extract.conn.driver=${database.driverName}

extract.conn.url=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1O11
L56

#Source System Id

extract.system.id=

# Specify the schema names for Extract

extract.schema.mantas=${schema.mantas.owner}

extract.schema.case=ATOMIC

extract.schema.business=${schema.business.owner}

extract.schema.market=${schema.market.owner}

extract.user.miner=${load.user.miner}

extract.miner.password=${utils.miner.password}


# File Paths for Extract
```

(Continued on next page)

(Continued from previous page)

```
#Specify the full path in which to place extracted scenarios

extract.dirname=/scratch/ofsaaapp/OFSBD 8.0.2/OFSBD 8.0.2_B06/
BDP62_B06/database/db_tools/data


#Specify the full path of the directory where the backups for the
extracted scripts would be maintained

extract.backup.dir=/scratch/ofsaaapp/OFSBD 8.0.2/OFSBD 8.0.2_B06/
BDP62_B06/database/db_tools/data/temp


#Controls whether jobs and thresholds are constrained to IDs in the
product range (product.id.range.min)

# through product.id.range.max). Values are Y and N. If the range is not
restricted, you can use range.check

# to fail the extract if there are values outside the product range.

extract.product.range.only=N

extract.product.range.check=N


#### LOAD


# Specify the jdbc driver details for connecting to the target database

load.conn.driver=${database.driverName}

load.conn.url=${utils.database.urlName}


#Target System ID

load.system.id=Ti1O11L56


# Specify the schema names for Load

load.schema.mantas=${schema.mantas.owner}

load.schema.case=ATOMIC

load.schema.business=${schema.business.owner}

load.schema.market=${schema.market.owner}

load.user.miner=${utils.miner.user}

load.miner.password=${utils.miner.password}
```

(Continued on next page)

(Continued from previous page)

```
#Directory where scenario migration files reside for loading
load.dirname=/scratch/ofsaaapp/OFSBD 8.0.2/OFSBD 8.0.2_B06/BDP62_B06/
database/db_tools/data

# Specify whether threshold can be updated
load.threshold.update=Y

# Specify whether or not to verify the target environment on load
verify.target.system=N
```

Figure 48: Sample `install.cfg` File for Scenario Migration

> **NOTE** In the `install.cfg` file, entries are in the form `Property1=${Property2}`. That is, the value for Property1 is the value that processing assigns to Property2. As such, if you change Property2's value, Property1's value also changes.

7.2.2.1.1 Configuring the Environment

To configure the environment for scenario migration, modify the parameters that the sample `<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/install.cfg` shows. The tables in the following sections describe the parameters specific to the Scenario Migration Utility.

7.2.2.1.2 Configuring General Scenario Migration

The following table describes general scenario migration parameters.

Table 39: General Scenario Migration Parameters

Parameter	Description
score.include	Flag that indicates whether scenario migration includes scenario scoring metadata; value is "Y" or "N" (the default).
wrapper.include	Flag that indicates whether scenario migration includes wrapper metadata; value is " Y" or "N" (the default).
sm.depfile	Location of the scenario migration dependencies file, <OFSAAI Installed Directory>/database/db_tools/mantas_cfg/dep.cfg.
sm.release	Version of the Scenario Migration Utility.

WARNING Oracle strongly recommends that you maintain scores and threshold values in a single environment. Maintaining these attributes in multiple environments and migrating the scenarios between the environments can cause the loss of threshold set-specific scoring rules.

7.2.2.1.3 Configuring Scenario Extraction

The following table describes scenario extraction parameters.

Table 40: Scenario Extraction Parameters

Parameter	Description
extract.database.username	User used to connect to the database when extracting scenarios (ATOMIC).
extract.database.password	Password for the above user.
extract.conn.driver	Database connection driver that the utility is to use (oracle.jdbc.driver.OracleDriver).
extract.conn.url	Database connection string that the Scenario Migration Utility is to use.
extract.system.id	System from which the scenario was extracted.
extract.schema.mantas	ATOMIC schema owner in the database into which extraction of the scenarios occurs (ATOMIC).
extract.schema.business	ATOMIC schema owner in the database into which extraction of the scenarios occurs (ATOMIC).
extract.schema.market	ATOMIC schema owner in the database into which extraction of the scenarios occurs (ATOMIC).
extract.user.miner	ATOMIC schema owner in the database into which extraction of the scenarios occurs (ATOMIC).
extract.miner.password	Password for the above user.
extract.dirname	Full path to the target directory where the utility writes extracted metadata <OFSAAI Installed Directory>/database/ db_tools/ data).
extract.backup.dir	Full path to the target directory where the utility writes backups of the extracted metadata (<OFSAAI Installed Directory>/ database/ db_tools/data/temp).
extract.product.range.only	Indicator (Y or N) of whether to extract custom patterns, jobs, thresholds, threshold sets, and scoring rules when extracting a scenario. Set to Y to prevent extraction of these entities.
extract.product.range.check	(For internal use only.) Indicator (Y or N) of whether to fail the extraction of a scenario if any metadata has sequence IDs outside the product range. Set to Y to fail the extraction.

7.2.2.1.3.1 Configuring Scenario Load

The following table describes scenario load parameters.

Table 41: Scenario Load Parameters

Parameter	Description
load.conn.driver	Database connection driver that the utility is to use (oracle.jdbc.driver.OracleDriver).
load.conn.url	Database connection string that the Scenario Migration Utility is to use.
load.ignore.custom.patterns=N	When set to N, custom patterns will not be ignored. This mode should be used when migrating scenarios between environments within the client's environment. If a custom pattern is not in the loaded XML file, then it will be deactivated. When set to Y, any custom patterns will be ignored by the load process, and should continue to operate.
load.schema.mantas	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
load.schema.business	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
load.schema.market	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
load.user.miner	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
load.miner.password	Password for the above user.
load.threshold.update	Threshold values from the incoming scenario. • Selecting N retains the threshold values from the target environment. • Selecting Y updates thresholds in the target environment to values from the incoming file.
load.system.id	Name that is assigned to the system into which this instance of Scenario Migration loads metadata. The system compares the value for this setting to the target system in the metadata file.
load.dirname	Directory from which the system loads scenario, network, and dataset XML files.

Table 41: Scenario Load Parameters (Continued)

Parameter	Description
verify.target.system	Check target name upon loading metadata files. • Setting to N prevents Scenario Migration from checking the load.system.id against the target system specified when the scenario, network or dataset was extracted. • Setting to Y enables this check. If the target in the XML file does not match the setting for load.system.id or the target is present in XML file but the load.system.id is blank then the system prompts you for an appropriate action. You can then continue with load or abandon the load, and you can apply the same answer to all other files in the session of Scenario Migration or allow the utility to continue prompting on each XML file that has a mismatch.

7.2.2.2 Extracting Scenario Metadata

Scenario metadata includes XML files that contain the table data for scenario, dataset, and network logic. The `sm_extract.sh` script invokes a Java tool, which creates these files. You start this script as follows:

```
sm_extract.sh <mode> -notarget | -target <name>
```

where:

- `mode` (mandatory) is the scenario, network, or dataset.
- `-notarget`, if included, implies that the system does not save the target environment to the generated XML files.
- `-target <name>` identifies the same target (in `<name>`) for all extracted XML files.

If you do not specify `-notarget` or `-target <name>` on the command line, the system prompts you to supply a target environment on each extracted file.

To extract scenario, dataset, and network metadata, follow these steps:

7. Navigate to the

   ```
   cd <OFSAAI Installed Directory>/db_tools directory
   ```

8. Edit the metadata configuration files with identifying information for the scenarios, datasets, or networks for extraction:
   - `<scnro_ctlg_id>` in the `scnros.cfg` file

     and/or

   - `<scnro_ctlg_id>.<scnro_id>` in the `scnros.cfg` file

   NOTE Providing both `<scnro_ctlg_id>` and `<scnro_id>` in the `scnros.cfg` file allows finer granularity when extracting scenarios. If you provide both a scenario catalog ID and a scenario ID on a line, you must separate them with a period.

   - `<data_set_id>` in the `dataset.cfg` file
   - `<network_id>` in the `network.cfg` file

9. Execute the `sm_extract.sh` script in this order:
   a. Enter `sm_extract.sh dataset` to extract dataset metadata.
   b. Enter `sm_extract.sh scenario` to extract scenario metadata.
   c. Enter `sm_extract.sh network` to extract network metadata.

7.2.2.3 Loading Scenario Metadata

The `sm_load.sh` script loads translated XML table data files into the target database.

To avoid corrupting the Behavior Detection process, never load scenarios while the process is running.

To load scenario, dataset, and network metadata, follow these steps:

1. Navigate to the following directory:

   ```
   cd <OFSAAI Installed Directory>/db_tools
   ```

   Optional: Edit the metadata configuration files (that is, `scnros.cfg`, `dataset.cfg`, and `network.cfg`) with identifying information for the scenarios, datasets, or networks that you want to load:

   - `<scnro_ctlg_id>` in the `scnros.cfg` file

     and/or

   - `<scnro_id>` in the `scnros.cfg` file

   NOTE Providing both `<scnro_ctlg_id>` and `<scnro_id>` in the `scnros.cfg` file allows finer granularity when loading scenarios. You must separate values with a period per line.

   - `<data_set_id>` in the `dataset.cfg` file
   - `<network_id>` in the `network.cfg` file

2. Copy the XML files you plan to load into the directory that the load.dirname specifies in the `install.cfg` file.
3. Execute the `sm_load.sh` script:
   a. Enter `sm_load.sh dataset` to load dataset metadata.
   b. Enter `sm_load.sh scenario` to load scenario metadata.
   c. Enter `sm_load.sh network` to load network metadata.

7.2.3 Scenario Migration Best Practices

Migrating scenarios from one environment to another requires a unified process in order to prevent conflicts and errors. This section describes the recommended best practices for scenario migration for any existing OFSBD system.

ATTENTION Not following the recommended best practices while loading scenarios to the targeted system may cause one or more sequence ID conflicts to occur, and your scenario will not be loaded. Once a conflict occurs, the metadata in the target environment must be corrected before the scenario can be successfully loaded.

To execute the recommended best practices, you should have an intermediate level knowledge of the scenario metadata, and be familiar with scenario patterns, thresholds, threshold sets, and so on. Basic SQL are required, as well as access privileges to the ATOMIC schema. You must also be able to update records through SQLPLUS or a similar DB utility.

7.2.3.1 Process Overview

Scenario metadata is stored in many tables, with each table using a unique sequence ID for each of its records. If scenarios, thresholds, and scoring rules are modified in multiple environments using the same sequence ID range, then conflicts may occur when you migrate scenarios to these environments. To prevent conflict, you must set different sequence ID ranges in each of the environments.

The recommended best practices contain two basic points:

- Make changes in only one environment
- Separate the sequence ID ranges

7.2.3.2 Best Practices

Prepare to implement the recommended best practices before installing OFSBD. Once the application is installed you should execute these steps to avoid scenario migration problems.

7.2.3.2.0.1 Making Changes in Only One Environment

1. Only make changes to scenarios, thresholds, threshold sets, and scoring rules in the source environment.
2. Test and confirm your changes in the source environment.
3. Extract scenarios from the source environment and migrate them to all of your target environments..

7.2.3.2.0.2 Separating Sequence ID Ranges

Conflicting sequence IDs are often the cause errors when you migrate a scenario, so it is important to separate the sequence ID range.

1. Review the `ATOMIC.KDD_COUNTER` table, which contains all sequence ID ranges and current values.
2. Start your sequence ID ranger at 10,000,000 and separate each environment by 10,000,000. The OFSBD product sequence ID range is >100,000,000.

7.2.3.3 Sequences to Modify

You should set these sequences before doing any work on scenarios, thresholds, or scoring rules.

Table 42 lists sequences involved and sample values for the Development environment.

Table 42: Environment 1 (Development)

TABLE_NM	SEQUENCE_NAME	CURRENT_VALUE	MIN_VALUE	MAX_VALUE
KDD_ATTR	ATTR_ID_SEQUENCE	10000000	10000000	19999999
KDD_AUGMENTATION	AGMNT_INSTN_ID_SEQ	10000000	10000000	19999999
KDD_DATASET	DATASET_ID_SEQUENCE	10000000	10000000	19999999
KDD_JOB	JOB_ID_SEQ	200000000	10000000	19999999

Table 42: Environment 1 (Development)

KDD_LINK_ANLYS_NTWRK_DEFN	NTWRK_DEFN_ID_SEQ	10000000	10000000	19999999
KDD_LINK_ANLYS_TYPE_CD	TYPE_ID_SEQ	10000000	10000000	19999999
KDD_NTWRK	NTWRK_ID_SEQ	10000000	10000000	19999999
KDD_PARAM_SET	PARAM_SET_ID_SEQ	200000000	10000000	19999999
KDD_PTTRN	PTTRN_ID_SEQ	10000000	10000000	19999999
KDD_RULE	RULE_ID_SEQ	10000000	10000000	19999999
KDD_SCNRO	SCNRO_ID_SEQ	10000000	10000000	19999999
KDD_SCORE	SCORE_ID_SEQ	10000000	10000000	19999999
KDD_SCORE_HIST	SCORE_HIST_SEQ_ID_SEQ	10000000	10000000	19999999
KDD_TSHLD	TSHLD_ID_SEQ	10000000	10000000	19999999
KDD_TSHLD_HIST	HIST_SEQ_ID_SEQ	10000000	10000000	19999999
KDD_TSHLD_SET	TSHLD_SET_ID_SEQ	10000000	10000000	19999999

Table 43 lists sequences involved and sample values for the Test/UAT environment.

Table 43: Environment 2 (Test/UAT)

TABLE_NM	SEQUENCE_NAME	CURRENT_VALUE	MIN_VALUE	MAX_VALUE
KDD_ATTR	ATTR_ID_SEQUENCE	20000000	20000000	29999999
KDD_AUGMENTATION	AGMNT_INSTN_ID_SEQ	20000000	20000000	29999999
KDD_DATASET	DATASET_ID_SEQUENCE	20000000	20000000	29999999
KDD_JOB	JOB_ID_SEQ	20000000	20000000	29999999
KDD_LINK_ANLYS_NTWRK_DEFN	NTWRK_DEFN_ID_SEQ	20000000	20000000	29999999
KDD_LINK_ANLYS_TYPE_CD	TYPE_ID_SEQ	20000000	20000000	29999999
KDD_NTWRK	NTWRK_ID_SEQ	20000000	20000000	29999999
KDD_PARAM_SET	PARAM_SET_ID_SEQ	20000000	20000000	29999999
KDD_PTTRN	PTTRN_ID_SEQ	20000000	20000000	29999999
KDD_RULE	RULE_ID_SEQ	20000000	20000000	29999999
KDD_SCNRO	SCNRO_ID_SEQ	20000000	20000000	29999999
KDD_SCORE	SCORE_ID_SEQ	20000000	20000000	29999999
KDD_SCORE_HIST	SCORE_HIST_SEQ_ID_SEQ	20000000	20000000	29999999
KDD_TSHLD	TSHLD_ID_SEQ	20000000	20000000	29999999

Table 43: Environment 2 (Test/UAT) (Continued)

KDD_TSHLD_HIST	HIST_SEQ_ID_SEQ	20000000	20000000	29999999
KDD_TSHLD_SET	TSHLD_SET_ID_SEQ	20000000	20000000	29999999

,Table 44 lists sequences involved and sample values for the Production environment.

Table 44: Environment 3 (PROD)

TABLE_NM	SEQUENCE_NAME	CURRENT_VALUE	MIN_VALUE	MAX_VALUE
KDD_ATTR	ATTR_ID_SEQUENCE	30000000	30000000	39999999
KDD_AUGMENTATION	AGMNT_INSTN_ID_SEQ	30000000	30000000	39999999
KDD_DATASET	DATASET_ID_SEQUENCE	30000000	30000000	39999999
KDD_JOB	JOB_ID_SEQ	30000000	30000000	39999999
KDD_LINK_ANLYS_NTWRK_DEFN	NTWRK_DEFN_ID_SEQ	30000000	30000000	39999999
KDD_LINK_ANLYS_TYPE_CD	TYPE_ID_SEQ	30000000	30000000	39999999
KDD_NTWRK	NTWRK_ID_SEQ	20000000	20000000	29999999
KDD_PARAM_SET	PARAM_SET_ID_SEQ	30000000	30000000	39999999
KDD_PTTRN	PTTRN_ID_SEQ	30000000	30000000	39999999
KDD_RULE	RULE_ID_SEQ	30000000	30000000	39999999
KDD_SCNRO	SCNRO_ID_SEQ	30000000	30000000	39999999
KDD_SCORE	SCORE_ID_SEQ	30000000	30000000	39999999
KDD_SCORE_HIST	SCORE_HIST_SEQ_ID_SEQ	30000000	30000000	39999999
KDD_TSHLD	TSHLD_ID_SEQ	30000000	30000000	39999999
KDD_TSHLD_HIST	HIST_SEQ_ID_SEQ	30000000	30000000	39999999
KDD_TSHLD_SET	TSHLD_SET_ID_SEQ	30000000	30000000	39999999

In order to update your database tables with recommended values, use SQLPLUS or a similar tool.

A sample SQL statement to update a set of sequence is:

```
UPDATE KDD_COUNTER
set min_value = 10000000,
    max_value = 19999999,
    current_value = 10000000
where sequence_name in
('DATASET_ID_SEQUENCE',
 'ATTR_ID_SEQUENCE',
 'PARAM_SET_ID_SEQ',
```

```
'PTTRN_ID_SEQ',
'RULE_ID_SEQ',
'SCNRO_ID_SEQ',
'JOB_ID_SEQ',
'TSHLD_ID_SEQ',
'NTWRK_DEFN_ID_SEQ',
'TYPE_ID_SEQ',
'TAB_ID_SEQ',
'TSHLD_SET_ID_SEQ',
'HIST_SEQ_ID_SEQ',
'AGMNT_INSTN_ID_SEQ',
'SCORE_ID_SEQ',
'SCORE_HIST_SEQ_ID_SEQ');
Commit;
```

Repeat for each environment, remembering to change the values for min, max, and current.

7.3 Managing the Threshold Editor

When scenarios are created, thresholds are established that enable you to modify the values of these thresholds in a production environment. Once the application is in the production environment, any user assigned the Data Miner role can use the Threshold Editor to modify threshold values of any installed scenario, and threshold sets to fine-tune how that scenario finds matches. Using this tool, you can enter a new value for a threshold (within a defined range) or reset the thresholds to their sample values.

The Threshold Editor page can be used for modifying the scenario thresholds and test run the scenario to know the number of matches that are generated through the test run. It can also be used to create a new threshold set based on the already available threshold set to modify the threshold and test the scenario.

A scenario is installed using the sample list of thresholds and values. This sample list of thresholds is referred to as the *base threshold set.* During deployment, you can create additional threshold sets to support specific business needs using the Oracle Financial Services Scenario Manager application. For more information about the Scenario Manager application, see the Oracle Financial Services Scenario Manager User Guide.

> **NOTE** Changing scenario threshold values can generate significantly more or fewer events, depending upon the modifications made.

The following subsections discuss features you encounter while using the Threshold Editor:

- Threshold Sets
- Inactive Thresholds

For more information about scenarios, see the *Trade-Based Anti Money Laundering Technical Scenario Description.*

7.3.1 Threshold Sets

Threshold sets allow you to run the same scenario multiple times against a variety of sources (for example, exchanges, currencies, or jurisdictions) with separate threshold values for each source.

For example, you may have a scenario with the base threshold set and two additional threshold sets that were created during deployment. You decide that you need this scenario to detect matches in transactions with a minimum value in US currency, European currency, and Japanese currency. Rather than changing the base threshold set for each situation, you can set the value of the base threshold set to detect US currency (for example, USD 100,000), the second threshold set to detect European currency (for example, EUR 150,000), and the third threshold set to detect Japanese currency (for example, JPY 125,000).

Since threshold sets two and three have only a few fields that differ from the base threshold set, you can check the Inherit Base Value check box feature for those fields that are exactly the same as the base threshold set. This feature associates the threshold values in the threshold set you are modifying with the corresponding values in the base threshold set. This association copies the corresponding base threshold set values to the set you are modifying and automatically updates them if the base value changes (refer to <Scenario–Threshold Set> Area for more information).

You do not have to run all three jobs all the time. Each threshold set has a unique ID, so you can tell the system which set to run and how often to run it. Refer to your scheduling tool's (for example, Control-M) documentation to sequence these jobs.

> **NOTE** Use the Threshold Editor to modify the values of existing threshold sets. Threshold sets can be created either through the Add New Threshold Set button or through the Scenario Manager.

7.3.2 Inactive Thresholds

For scenarios to work properly, thresholds that are not being used by a scenario must have their values set to Inactive. The following groups of thresholds can have values set to Inactive:

- Mutually Exclusive Thresholds
- Additional Scenario Thresholds

7.3.2.1 Mutually Exclusive Thresholds

In some situations, scenarios apply the value of one threshold only when the value of another threshold is set to *N* for no. These types of thresholds are referred to as a *mutually exclusive* thresholds.

For example, the use of the *Included Jurisdiction Codes* threshold is contingent upon the value of the *All Jurisdictions* threshold.

Table 45 shows how mutually exclusive thresholds work in two different situations.

Table 45: Mutually Exclusive Thresholds

Threshold	Situation 1	Situation 2
All Jurisdictions	Y	N
Included Jurisdiction Codes	Inactive	North, East

If the value of the *All Jurisdictions* threshold is set to Y for yes (Situation 1), then the *Included Jurisdiction Codes* threshold values are not used and have the value set to Inactive. Conversely, if the

value of the *All Jurisdictions* threshold is set to *N* for no (Situation 2), then the scenario only uses the value specified by the *Included Jurisdiction Codes* threshold (that is, North, East).

7.3.2.2 Additional Scenario Thresholds

Your deployment may not need to utilize all the thresholds established within a particular scenario. The mutually exclusive thresholds not used by the scenario are set to Inactive.

7.3.3 About the Threshold Editor Screen Elements

The following screen elements display in the Threshold Editor:

- Search Bar
- <Scenario–Threshold Set> Area

7.3.3.1 Search Bar

The search bar allows you to search for threshold values by selecting a specific scenario and threshold set (Figure 49).

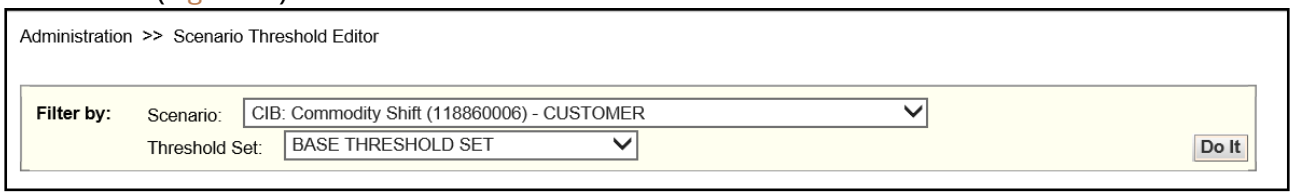


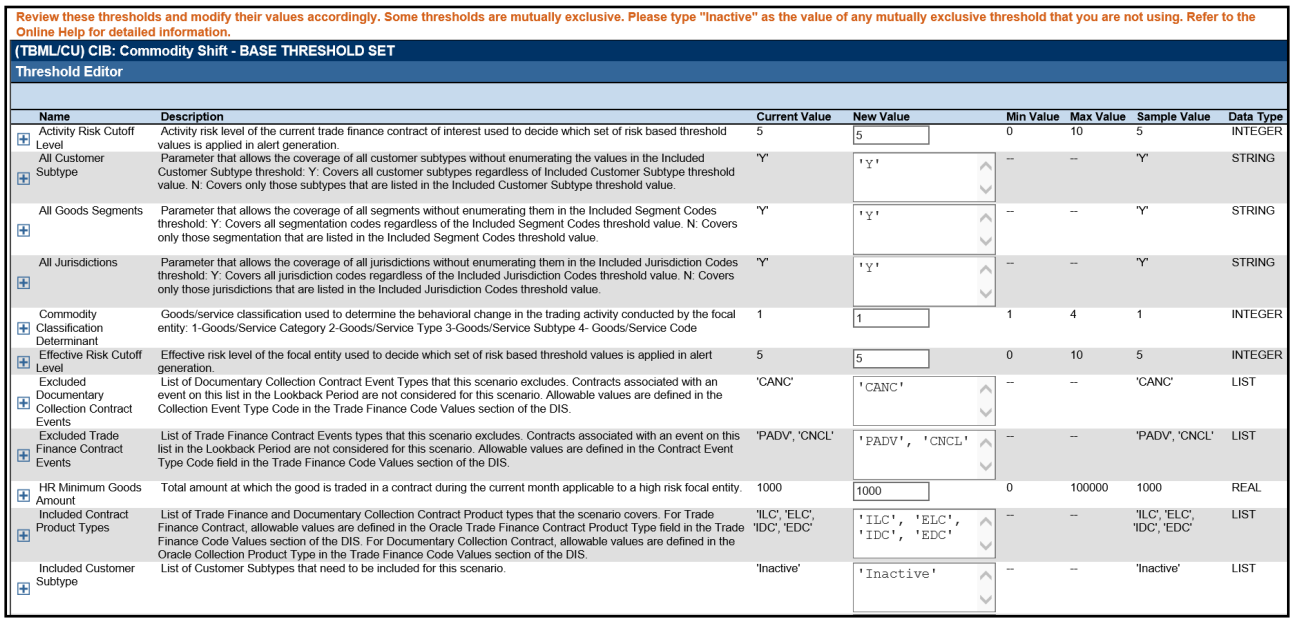
Figure 49: Search Bar

The components of the search bar includes the following:

- **Filter by: Scenario** drop-down list: Provides a list of scenarios displayed by the scenario's short name, ID number, and focus type (for example, CIB: Commodity Shift(118860006) – CUSTOMER).
- **Filter by: Threshold Set** drop-down list: Provides a list of Threshold Sets associated with the scenario displayed in the Scenario drop-down list. The base threshold set displays first, followed by additional threshold sets listed in ascending alphabetical order.
- **Do It** button: When clicked, displays the threshold values for the scenario and threshold set selected in the search bar.

7.3.3.2 <Scenario–Threshold Set> Area

The <Scenario-Threshold Set> Area displays the list of threshold values for a selected scenario and threshold set (Figure 50). This list displays after you select a scenario and threshold set in the search bar and click **Do It**.



Review these thresholds and modify their values accordingly. Some thresholds are mutually exclusive. Please type "Inactive" as the value of any mutually exclusive threshold that you are not using. Refer to the Online Help for detailed information.

(TBML/CU) CIB: Commodity Shift - BASE THRESHOLD SET

Threshold Editor

Name	Description	Current Value	New Value	Min Value	Max Value	Sample Value	Data Type
Activity Risk Cutoff Level	Activity risk level of the current trade finance contract of interest used to decide which set of risk based threshold values is applied in alert generation.	5	5	0	10	5	INTEGER
All Customer Subtype	Parameter that allows the coverage of all customer subtypes without enumerating the values in the Included Customer Subtype threshold: Y: Covers all customer subtypes regardless of Included Customer Subtype threshold value. N: Covers only those subtypes that are listed in the Included Customer Subtype threshold value.	'Y'	'Y'	--	--	'Y'	STRING
All Goods Segments	Parameter that allows the coverage of all segments without enumerating them in the Included Segment Codes threshold: Y: Covers all segmentation codes regardless of the Included Segment Codes threshold value. N: Covers only those segmentation that are listed in the Included Segment Codes threshold value.	'Y'	'Y'	--	--	'Y'	STRING
All Jurisdictions	Parameter that allows the coverage of all jurisdictions without enumerating them in the Included Jurisdiction Codes threshold: Y: Covers all jurisdiction codes regardless of the Included Jurisdiction Codes threshold value. N: Covers only those jurisdictions that are listed in the Included Jurisdiction Codes threshold value.	'Y'	'Y'	--	--	'Y'	STRING
Commodity Classification Determinant	Goods/service classification used to determine the behavioral change in the trading activity conducted by the focal entity: 1-Goods/Service Category 2-Goods/Service Type 3-Goods/Service Subtype 4- Goods/Service Code	1	1	1	4	1	INTEGER
Effective Risk Cutoff Level	Effective risk level of the focal entity used to decide which set of risk based threshold values is applied in alert generation.	5	5	0	10	5	INTEGER
Excluded Documentary Collection Contract Events	List of Documentary Collection Contract Event Types that this scenario excludes. Contracts associated with an event on this list in the Lookback Period are not considered for this scenario. Allowable values are defined in the Collection Event Type Code in the Trade Finance Code Values section of the DIS.	'CANC'	'CANC'	--	--	'CANC'	LIST
Excluded Trade Finance Contract Events	List of Trade Finance Contract Events types that this scenario excludes. Contracts associated with an event on this list in the Lookback Period are not considered for this scenario. Allowable values are defined in the Contract Event Type Code field in the Trade Finance Code Values section of the DIS.	'PADV', 'CNCL'	'PADV', 'CNCL'	--	--	'PADV', 'CNCL'	LIST
HR Minimum Goods Amount	Total amount at which the good is traded in a contract during the current month applicable to a high risk focal entity.	1000	1000	0	100000	1000	REAL
Included Contract Product Types	List of Trade Finance and Documentary Collection Contract Product types that the scenario covers. For Trade Finance Contract, allowable values are defined in the Oracle Trade Finance Contract Product Type field in the Trade Finance Code Values section of the DIS. For Documentary Collection Contract, allowable values are defined in the Oracle Collection Product Type in the Trade Finance Code Values section of the DIS.	'ILC', 'ELC', 'IDC', 'EDC'	'ILC', 'ELC', 'IDC', 'EDC'	--	--	'ILC', 'ELC', 'IDC', 'EDC'	LIST
Included Customer Subtype	List of Customer Subtypes that need to be included for this scenario.	'Inactive'	'Inactive'	--	--	'Inactive'	LIST

Figure 50: <Scenario-Threshold Set> Area

The <Scenario-Threshold Set> Area includes the following components and contents:

- Long name of the scenario and the name of the threshold set in the title of the <Scenario-Threshold Set> bar.
- List of scenario thresholds by threshold name, sorted in ascending alphabetical order.
- Threshold information as follows:
  - **Threshold History Icon**: Expands or contracts the Threshold History inset that displays a history of all modifications to the selected threshold value in reverse chronological order by creation date. Information displayed includes the creation date, user name, threshold value, and any comment associated with the threshold value change.

    If comments are displayed and the comment text consists of more than 100 characters, the Threshold Editor displays the first 100 characters followed by an ellipsis (...) indicating that more text is available. When you click the ellipsis, the entire comment displays in the Expanded Comments dialog box for ease of viewing.
  - **Name**: Displays the name of the threshold.
  - **Description**: Displays the description of the threshold.
  - **Current Value**: Displays the current value of the threshold. If the data type of the threshold is *LIST*, multiple values are displayed in a comma-delimited list, with each value contained in single quotes (' ').Thresholds with an *Inactive* current value are not being used by the scenario (refer to Inactive Thresholds for more information).
  - **Inherit Base Value**: Enables you to select the check box to apply the corresponding threshold values from the base threshold set to the threshold set displayed. Selecting the check box disables the New Value text box. This option does not display for the base threshold set.
  - **New Value**: Displays the current value of the threshold in the editable New Value text box if the Inherit Base Value check box is not selected. If the data type for the threshold is *LIST*, multiple values are displayed in a comma-delimited list, with each value contained in single quotes (' ').

- **Min Value**: The minimum value of the threshold.
- **Max Value**: The maximum value of the threshold.
- **Sample Value**: The sample value of the threshold.
- **Data Type**: The type of data that is utilized by a threshold in a scenario. There are five data types: Integer, Boolean, Real, String, and List. Place your cursor over this value to display the threshold unit of measure (for example, days, percentage, or distance).
- **Add A Comment**: Provides a place to type comments. When you type a comment and click **Save**, the same comment is applied to each modified threshold.

- **Restore Samples Values:** Restores all thresholds within the selected scenario threshold set to the sample values
- **Save**: Saves all modifications to the database.
- **Cancel**: Redisplays the Threshold Editor without the <Scenario-Threshold Set> Area and does not save your changes.
- Test: When the Test button is clicked, *Scenario Test Execution* pop-up window is displayed, showing the following fields:

Table 46: Scenario Test Execution components

Field	Description
Scenario Name	This field is non-editable and displays the scenario that has been selected in the drop-down list from the threshold editor page.
Threshold Set	This is a non-editable text box which displays the threshold set name that has been selected for test run.
Pattern	Select the pattern from the drop-down list that are part of the selected scenario. **Note:** Since the scenario job runs based on the pattern, you cannot run multiple patterns of the scenario at the same time.
Processing Batch Date	Select the date based on which the scenario patterns will run.
Processing Batch Name	Select the batch name from the drop-down list. **Note:** If a Batch with the selected Processing Batch Name and Date is already running, then the following error message is displayed: *A Batch with the selected Processing Batch Name and Date is already running. Please wait till the Batch completes*.

- **Update Product Threshold Set**: Enables you to update the test threshold set to product threshold set. This button is enabled only when the threshold set selected is newly created threshold.

7.3.4 Using the Threshold Editor

The Threshold Editor configures scenario threshold values by:

- Providing threshold values for a specific scenario and threshold set
- Accepting and validating user-entered threshold values
- Saving the modified threshold values to the database

This section explains the following functions of the Threshold Editor:

- Adding a New Threshold Set
- Changing a Scenario Threshold
- Resetting a Scenario Threshold to the Sample Values
- Viewing a Scenario Threshold's History
- Viewing Expanded Comments

7.3.4.1 Adding a New Threshold Set

To add a new scenario threshold set, follow these steps:

1. Select the desired scenario from the **Filter by: Scenario** drop-down list.
2. Click **Add New Threshold**. The Add New Threshold Set pop-up window is displayed.
3. Enter the required details in the following fields:

Table 47: Add New Threshold Set Components

Field	Description
Scenario	This field is non editable and displays the scenario that has been selected in the Filter by: Scenario drop-down list.
Available Threshold Sets	This drop-down list displays all the available threshold sets in the system for the selected scenario. This is required to acquire the thresholds for the new threshold set that is being created. • If the user does not select a value from the "Available Threshold sets" drop-down list, the following error message is displayed: *Please select a threshold set from the available threshold sets drop-down to create a new threshold set.* • If the user has selected a threshold set which doesn't have an associated job, then the following error message is message: The selected threshold set doesn't have the required Job and Job Dataset for running the scenario test execution. Please select any other threshold set and take action.
Test Threshold Set	Select this checkbox if the threshold set created is a test threshold set or not. The threshold set name is available threshold set name + _TST_datetimestamp.
New Threshold Set Name	By default, this field is kept blank. You can enter the threshold set name only when the Create Test Threshold Set checkbox is not selected. When the user has selected the Test Threshold Set checkbox, then this field is pre-populated with a value.

4. Click **Save**. The Threshold Set is added.

7.3.4.2 Changing a Scenario Threshold

To change a scenario threshold value, follow these steps:

1. Select the desired scenario from the **Filter by: Scenario** drop-down list.
2. Select the desired threshold set from the **Filter by: Threshold Set** drop-down list.
3. Click **Do It**.

   The system displays the threshold values for the scenario and threshold set selected.
4. Type a new value in the **New Value** box for each threshold that you wish to update.

If you are not updating a base threshold set, you can inherit corresponding values from the base threshold set by checking the **Inherit Base Value** check box.

Optional: Enter any comments in the **Add A Comment** text box.

5. Click **Save**.

   The new threshold values display in the Threshold List for <Scenario-Threshold Set>.

7.3.4.3 Resetting a Scenario Threshold to the Sample Values

To reset a scenario's threshold sample values, follow these steps:

1. Select the desired scenario from the **Filter by: Scenario** drop-down list.
2. Select the desired threshold set from the **Filter by: Threshold** Set drop-down list.
3. Click **Do It**.

   The system displays the threshold values for the scenario and threshold set selected.

4. Click **Restore Sample Values** button.

   The Confirmation dialog box displays the following message: *Are you sure you want to restore the threshold values of the displayed threshold set to their sample values?*

   To restore thresholds that have the Inherit Base Value check box selected, you must clear the check box. Click **OK** to return to the Threshold Editor with the sample values displayed, then click **Save.** Click **Cancel** to retain the current values.

5. Click **OK**.

   The dialog box closes and the sample values display in the [Scenario-Threshold Set] Area.

6. Click **Save**.

   The database is updated to reflect the changes.

7.3.4.4 Viewing a Scenario Threshold's History

To view the modification history for a specific threshold, follow these steps:

1. Click **Expand** next to the desired threshold.

   The Threshold History inset displays with the history for the threshold selected.

2. Click **Contract** next to the threshold to hide the Threshold History inset.

7.3.4.5 Viewing Expanded Comments

To view an expanded comment in the Scenario Threshold inset, follow these steps:

1. Click the **ellipsis (...)** at the end of the comment in the Scenario Threshold inset.

   The entire comment, up to 4,000 characters, displays in the Expanded Comments dialog box (Figure 51).

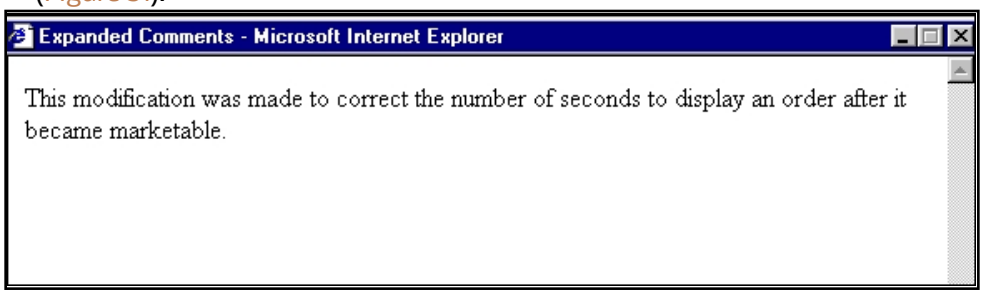


Figure 51: Example Expanded Comment Dialog Box

2. Click **Close (X)** on the top right corner to close the dialog box.

7.4 Configuring Administration Tools

Follow these steps for Administration Tools configuration:

If the administration tool is deployed on a separate web application server, then perform these steps:

1. Log in as an Administrator User. The Home page displays.
2. Click **Manage Configuration** from the LHS menu.
3. Select the **Manage Common Parameters**.
4. In the Parameter Category drop-down, select **Used for Design**.
5. In the Parameter Name drop-down, select **Admin Tools**.
6. Set the Attribute 2 Value as follows: <PROTOCOL>://<AdminTools_WEB_SERVER_NAME>:<PORT>
   - <PROTOCOL> is web page access PROTOCOL (http or https).
   - <AdminTools_WEB_SERVER_NAME> is the FQDN of the web application server hosting Administrative Tools.
   - <PORT> is the web application server port hosting Administration Tools.

A Logging

This appendix describes the mechanism that TBAML uses when logging system messages.

A.1 About System Log Messages

The Common Logging component provides a centralized mechanism for logging TBAML messages, in which the system places all log messages in a single message library file.

In the event that a log file becomes very large (one gigabyte or more), the system creates a new log file. The naming convention is to add *.x* to the log file's name , such as `mantas.log`, `mantas.log.1`, `mantas.log.2`.

> **NOTE** The log file size is a configurable property. The default value for this property is 10 MB. The maximum file size should not exceed two gigabytes (2000000000 bytes).

A.2 Message Template Repository

The message template repository resides in a flat text file and contains messages in the format `<message id 1> <message text>`. The following is an example of a message repository's contents:

```
111 Dataset id {0} is invalid

112 Run id {0} running Pattern {1} failed

113 Checkpoint false, deleting match
```

111, 112, and 113 represent message IDs; whitespace and message text follow. The {0}s and {1}s represent placeholders for code variable values.

Each subsystem has its own repository.

The naming convention for each message library file is:
`mantas_<subsystem>_message_lib_<language-code>.dat`

where

`<subsystem>` is the name of the subsystem and

`<language-code>` is the two-character Java (ISO 639) language code.

For example, the English version of the Algorithms message library is `mantas_algorithms_message_lib_en.dat`.

The `log.message.library` property that the subsystem's base `install.cfg` file contains the full path to a subsystem's message library file.

A.3 Logging Levels

Table 48 outlines the logging levels that the Common Logging component supports.

Table 48: Logging Levels

Severity (Log Level)	Usage
Fatal	Irrecoverable program, process, and thread errors that cause the application to terminate.
Warning	Recoverable errors that may still enable the application to continue running but should be investigated , such as failed user sessions or missing data fields).
Notice (default)	High-level, informational messaging that highlights progress of an application , such as startup and shutdown of a process or session, or user login and logout.
Diagnostic	Fine-grained diagnostic errors—used for viewing processing status, performance statistics, SQL statements, etc.
Trace	Diagnostic errors—use only for debugging purposes as this level enables all logging levels and may impact performance.

The configuration file specifies enabling of priorities in a hierarchical fashion. That is, if Diagnostic is active, the system enables the Notice, Warning, and Fatal levels.

A.4 Logging Message Libraries

Some Oracle subsystems produce log output files in default locations. The following sections describe these subsystems.

A.4.1 Verifying the Schema Creator Log Files

The log files can be found at the following paths:

For batch logs: `FTPSHARE/logs`

For Application logs: `FIC_HOME/logs`

A.4.2 Administration Tools

The following file is the message library for the Administration Tools application:

```
$FIC_WEB_HOME/AM/admin_tools/WEB-INF/classes/conf/mantas_cfg/etc/
mantas_admin_tools_message_lib_en.dat
```

All message numbers that this log contains must be within the range of 50,000 - 89,999.

A.4.3 Database

The following file is the message library for the Database:

```
<OFSAAI Installed Directory>/database/db_tools/mantas_cfg/etc/
mantas_database_message_lib_en.dat
```

All message numbers that this file contains must be within the range of 250,000 - 289,999.

A.4.4 Scenario Manager

The following file is the message library for the Scenario Manager:

<OFSAAI Installed Directory>/behavior_detection/toolkit/mantas_cfg/etc/
mantas_toolkit_message_lib_en.dat

All message numbers that this section contains must be within the range of 130,000 - 169,999.

A.4.5 Services

The following file is the message library for the Services:

```
<OFSAAI Installed Directory>/services/server/webapps/mantas/WEB-INF/classes/
conf/
mantas_cfg/etc/mantas_event_management_message_lib_en.dat
```

All message numbers that this section contains must be within the range of 210,000 - 249,999.

A.5 Alert Management

The following logs contain the message library for the Alert Management application:

- Web Server Logs
- Application Server Logs
- Database Objects Logs
- Ingestion Manager

A.5.1 Web Server Logs

The following file is the message library for the Web server logs:

$FIC_WEB_HOME/logs/UMMService.log

A.5.2 Application Server Logs

The following file is the message library for the Application Server logs:

$FIC_APP_HOME/common/ficserver/logs/RevAppserver.log

A.5.3 Database Objects Logs

DB objects logs used in the application are maintained in the table `KDD_LOGS_MSGS`. An entry in this table represents the timestamp, stage, error code and module.

A.5.4 Ingestion Manager

The following file is the message library for the Ingestion Manager:

```
<OFSAAI Installed Directory>/ingestion_manager/config/message.dat
```

A.6 Logging Configuration File

You can configure common logging through the following files depending on the subsystem you want to modify. The following table lists the subsystems and their log files:

Table 49: Logging Configuration Files

Subsytem	File
Database	`<OFSAAI Installed Directory> / database/db_tools/log4j2.xml`
Scenario Manager	`<OFSAAI Installed Directory>/ behavior_detection/toolkit/ mantas_cfg/install.cfg`
Behavior Detection	`<OFSAAI Installed Directory>/ behavior_detection/algorithms/MTS/ mantas_cfg/install.cfg`

Administration Tools Web Server logs	`$FIC_WEB_HOME/conf/RevLog4jConfig.xml`
	`<root>`
	The following logger levels are available:
	• DEBUG
	• INFO
	• WARN
	• SEVERE
	• FATAL

Administration Tools Application Server logs	`$FIC_WEB_HOME/conf/RevLog4jConfig.xml`
	`<root>`
	`<priority value ="debug" />`
	`<appender-ref ref="ConsoleAppender1"/ >`
	`</root>`
	The following logger levels are available:
	• DEBUG
	• INFO
	• WARN
	• SEVERE
	• FATAL

Services	`<OFSAAI Installed Directory> / services/server/webapps/mantas/WEB-INF/log4j2.xml`
Ingestion Manager	`<OFSAAI Installed Directory> /ingestion_manager/config/ log4j2_common.xml`

The configuration file specifies enabling of priorities in a hierarchical fashion. For example, if Diagnostic priority is enabled, Notice, Warning, and Fatal are also enabled, but Trace is not.

In the configuration file, you can specify the following:

- Locations of recorded log messages
- Logging to the console, files, UNIX syslog, e-mail addresses, and the Microsoft Windows Event Viewer
- Routing based on severity and/or category
- Message library location
- Maximum log file size

A.6.1 Sample Configuration File

The following is a sample logging configuration file. Make special note of the comments in the following sample as they contain constraints that relate to properties and logging.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE log4j:configuration SYSTEM "log4j.dtd">
<log4j:configuration xmlns:log4j="http://jakarta.apache.org/log4j/">

<Appenders>



               <RollingFile name="@@CATAGORY@@" append="true" filePat-
tern="@@PATH@@">
      <FileName>@@PATH@@</FileName>
      <PatternLayout>
        <Pattern>[%d{E  dd/M/yyyy hh:mm:ss}] [@@CATAGORY@@] [%5p] - %m%n</
Pattern>
      </PatternLayout>
      <Policies>
        <SizeBasedTriggeringPolicy size="10000kb"/>
      </Policies>
       <DefaultRolloverStrategy max="20"/>
    </RollingFile>

                                            <Console name="stdout" tar-
get="SYSTEM_OUT">
            <PatternLayout>
                <pattern>
                  [%-5level] %d{yyyy-MM-dd HH:mm:ss.SSS} [%t] %c{1} - %msg%n
                </pattern>>
```

```
                </PatternLayout>
            </Console>
                  </Appenders>

                  <Loggers>
               <Logger name="@@CATAGORY@@" level="info" additivity="false">
                                                  <AppenderRef ref="@@CATAGORY@@"
level="trace"/>
                                                       <AppenderRef ref="stdout"
level="error"/>
               </Logger>



            <Root level="error">
                <AppenderRef ref="stdout"/>
            </Root>
        </Loggers>
<!--         <root>
<priority value="##PRIORITY##"></priority>
                  </root> -->
</log4j:configuration>
```

Figure 52: Sample Logging Configuration File

A.6.2 Configurable Logging Properties

Table 50 identifies the configurable properties for logging in an Oracle client's environment.

Table 50: Configurable Parameters for Common Logging

Property	Sample Value	Description
log.format	<Pattern>[%d{E dd/M/yyyy hh:mm:ss}] [@@CATAGORY@@] [%5p] - %m%n</Pattern>	Identifies the log formatting string. Refer to Apache Software's *Short Introduction to log4j* guide (http://logging.apache.org/log4j/docs/manual.html) for more details about the log message format.
log.message.library	To be specified at installation.	Identifies the full path and filename of the message library.

Table 50: Configurable Parameters for Common Logging (Continued)

Property	Sample Value	Description
log.max.size	<Policies> <SizeBasedTriggeringPolicy size=""10000kb""/> </Policies>	Determines the maximum size (in kilobytes) of a log file before the system creates a new log file.
log.category.<catgory_name>.location		Contains routing information for message libraries for this category.
log.categories.file.path	To be specified at installation.	Identifies the full path to the categories.cfg file.
log.<category_name>.<severity>.location		Contains routing information for message libraries with the given severity for the given category.
log4j.config.file	To be specified at installation.	Specifies the full path to the external log4j configuration file.
log.default.location		Contains routing information for message libraries for this category for which there is no location previously specified.
log.mantaslog.location		Contains routing information for message libraries for this category for which there is no location previously specified.
log.smtp.hostname		Identifies the hostname of the SMTP server if e-mail address is specified as log output.
log.fatal	true	Indicates that fatal logging is enabled; *false* indicates that fatal logging is not enabled.
log.fatal.synchronous	false	Indicates that fatal level logging should happen asynchronously; true indicates fatal level logging should happen synchronously. **Note:** Setting value to true (synchronous) may have performance impact
log.warning	true	Indicates enabling of warning logging; *false* indicates that warning logging is not enabled.

Table 50: Configurable Parameters for Common Logging (Continued)

Property	Sample Value	Description
log.warning.synchronous	false	Indicates that warning level logging should happen asynchronously; true indicates warning level logging should happen synchronously. **Note:** Setting value to true (synchronous) may have performance impact
log.notice	true	Indicates enabling of notice logging; *false* indicates that notice logging is not enabled.
log.notice.synchronous	false	Indicates that notice level logging should happen asynchronously; true indicates notice level logging should happen synchronously. **Note:** Setting value to true (synchronous) may have performance impact
log.diagnostic	false	Indicates that diagnostic logging is not enabled; *true* indicates enabling of diagnostic logging.
log.diagnostic.synchronous	false	Indicates that diagnostic level logging should happen asynchronously; true indicates diagnostic level logging should happen synchronously. **Note:** Setting value to true (synchronous) may have performance impact
log.trace	false	Indicates that trace logging is not enabled; *true* indicates enabling of trace logging.
log.trace.synchronous	true	Indicates that trace level logging should happen asynchronously; true indicates trace level logging should happen synchronously. **Note:** Setting value to true (synchronous) may have performance impact
log.syslog.hostname	hostname	Indicates the host name of syslog for messages sent to syslog.
log.time.zone	US/Eastern	Indicates the time zone that is used when logging messages.

A.6.3 Monitoring Log Files

When using a tool to monitor a log file, use the message ID to search for a particular log message instead of text within the message itself. Under normal circumstances, the message IDs are not subject to change between Oracle releases, but the text of the message can change. If a message ID does change, you can refer to the appropriate `readme.txt` file for information about updated IDs.

B Oracle Software Updates

This appendix describes the application of software updates in Oracle Financial Services Behavior Detection Platform:

B.1 Oracle Software Updates - Hotfix

A hotfix is a package that includes one or more files that are used to address a defect or a change request. Typically, hotfixes are small patches designed to address specific issues reported by the clients.

Hotfixes can affect the following areas in TBAML:

- User Interface (UI)
- Scenarios (patterns and datasets)
- Post-Processing jobs
- Performance
- Ingestion

Each hotfix includes a `readme.txt` file, which describes the step-by-step process to install the hotfix.

Hotfixes are delivered to clients in the following ways:

- E-mail
- Secure FTP

B.2 Hotfix Effect on Customization

When a hotfix is installed it can affect your customizations on the *User Interface* and *Scenarios*.

B.2.1 User Interface

If your UI customizations are correctly isolated to the `custom` directory, then the impact should be minimal. It is possible, however, that the hotfix changes information in the base product that you have customized. In that case, you cannot see the effect of the hotfix. To minimize this, be sure to avoid copying more than necessary to the `custom` directory. For example, you should not copy the entire `BF_Business.xml` file to override a few fields, you should create a new file in the `custom` directory that only contains the fields you are overriding.

The hotfixes delivered will include installation and deployment instructions in the fix documentation.

B.2.2 Scenarios

If you have customized scenarios (changed dataset logic or changed scenario logic), then applying a hotfix to that scenario will remove those customizations. If you customized datasets by creating a dataset override file, then your custom dataset continues to be used after applying the hotfix. It is possible that your custom dataset prevents the scenario fix from being evident (if the dataset you customized was one of the items changed by the hotfix). It is also possible that the hotfix changes the

fields it expects from the dataset you customized, causing the scenario to fail. For scenarios you have customized, you should always test the scenario hotfix without your customizations in place, then re-apply them to the scenario, if necessary.

C User Administration

This appendix describes the user administration of Oracle Financial Services Behavior Detection Platform.

C.1 Managing User Groups and User Roles

User Roles are pre-defined in Oracle solutions. Sample values for User groups are included in the installer but can be modified by clients to meet their specific needs. The corresponding mappings between User Roles and sample User Groups are pre-defined but can also be modified by clients to either adjust the role to sample user group mapping or to map roles to newly defined user groups.

The User Groups for TBAML are CMSUPERVISORUG and CMMANADMNUG.

For more information on creating a new user group and mapping it to an existing role, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

> **NOTE** While creating a new User Group, you can set precedence as 5001 or greater.

C.2 Managing User Groups

The following sections describe how to manage User Groups:

C.2.1 Defining User Group Maintenance Details

For more information on defining user group maintenance details, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

C.2.2 Adding New User Group Details

For more information on adding new user group details, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

C.2.3 Mapping Users to User Groups

One user can also be used against multiple roles. If multiple roles are allocated to a single user, then the availability of actions depends on the Four Eyes approval option. If Four Eyes approval is *off*, then the user can take all actions available by the allocated roles, with no duplicates. If Four Eyes approval is *on*,

then action linked to a role that does not require Four Eyes approval takes precedence if there is a conflict.

For more information on mapping users to user group, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

C.2.4 Mapping User Group(s) to Domain(s)

To map user group or groups to domain or domains, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

Actions to Role mappings are done through Database tables. Sample action to role mappings are included in the application. For more information on changing the mapping of roles to actions, see the Working with Alert Action Settings section of the Configuration Guide.

Actions are primarily associated with a User Role, not an individual user. However, the ability to Reassign To All when taking a Reassign action is associated at the individual user level. Reassign To All means that a user is allowed to assign to users and organizations that may not be within their normal viewing privileges.

C.2.5 Mapping a User to a Single User Group

If a user has only one role then that user can be mapped to a single User Group associated with that User Role. For more information on mapping a user to a single user group, see the Identity Management section of Oracle Financial Services Analytical Applications Infrastructure User Guide.

C.2.5.1 Mapping a User to Multiple User Groups

If a user has more than one role within FCCM (that is, within both TBAML and Enterprise Case Management), then the user must be mapped to the different User Groups associated with the corresponding role. When the user logs into FCCM, the user access permissions are the union of access and permissions across all roles.

C.2.5.2 Mapping a User to an Organization

If a user is mapped to an organization indicating that it is the line organization for the user and if there exists any child organization for that line organization, then those organizations are implicitly mapped to the user as a business organization. If the same organization is already mapped as the business organization, then the child of the organizations should not be mapped to the user implicitly by the system.

If an organization is implicitly mapped to the user based on line organization association, the user can still be unmapped from that organization if there is a need to limit them from seeing the organization. The organization still shows (I) in the Organization list to show that the organization is a child of the line organization. But the fact that it is not selected will prevent the user from being mapped to it.

The following rules apply:

- Users can have only one organization as the line organization.
- A child organization can have only one parent organization

To map organizations, follow these steps:

1. Select a user from the **Select User** drop-down list.

2. Select the line organization or organizations you want to map the user to from the Line Organization drop-down list.

> **NOTE** If the user is associated with both line and business organizations, then the business organizations associated to the Line Organization must be implicitly mapped and display the organizations as well.

The system visually distinguishes the Implicit (I), which is the system determination based on line organization and Explicit (E), which was manually added by the user mapping, of business organizations. The system displays either I or E in the brackets to indicate that the grid displays two different column, one for Implicit and the other one for Explicit mapping.

3. Click **Save.**

C.2.5.3 Mapping a Function to a Role

The following list of functions must be mapped to appropriate User Roles through Function-Role Map function, which is available in the Security Management System, by logging in as the System Administrator in the OFSAAI toolkit.

The following table provides the function role mapping details.

Table 51: Function to Role Mapping Details

Function	Description
AMACCESS	All behavior detection user roles should be mapped to the function AMACCESS in order to access an FCC TBAML event. Users of roles that are not mapped to this function cannot access the details of the Alerts.
CMACCESS	All Case Management user roles should be mapped to the function CMACCESS in order to access a Case. Users of roles that are not mapped to this function cannot access the details of the Case.
RSGNTALL	This function should be mapped to Case Analyst1, Case Analyst2 and Case Supervisor Roles to assign ownership of a case without applying restriction on the Organization associated with the Case. If the ownership assignment is required to be restricted based on Organization associated with the Case for any of these user roles, then the RSGNTALL function need not be mapped to the above roles.

C.3 Defining User Access Properties and Relationships

The following types of data compose a user's security configuration:

- **Business Domain(s):** Property that enables an Oracle client to model client data along operational business lines and practices.
- **Jurisdiction(s):** Property that enables an Oracle client to model client data across such attributes as geographic location, type, or category of a business entity.
- **Organization(s):** Department or organization to which an individual user belongs.
- **Role(s):** Permissions or authorizations assigned to a user in the system (such as Behavior Detection Framework Case Administrator or Auditor).
- **Scenario Group(s):** Group of scenarios that identify a set of scenario permissions and to which a user has access rights.

The following figure shows the user authorization model.

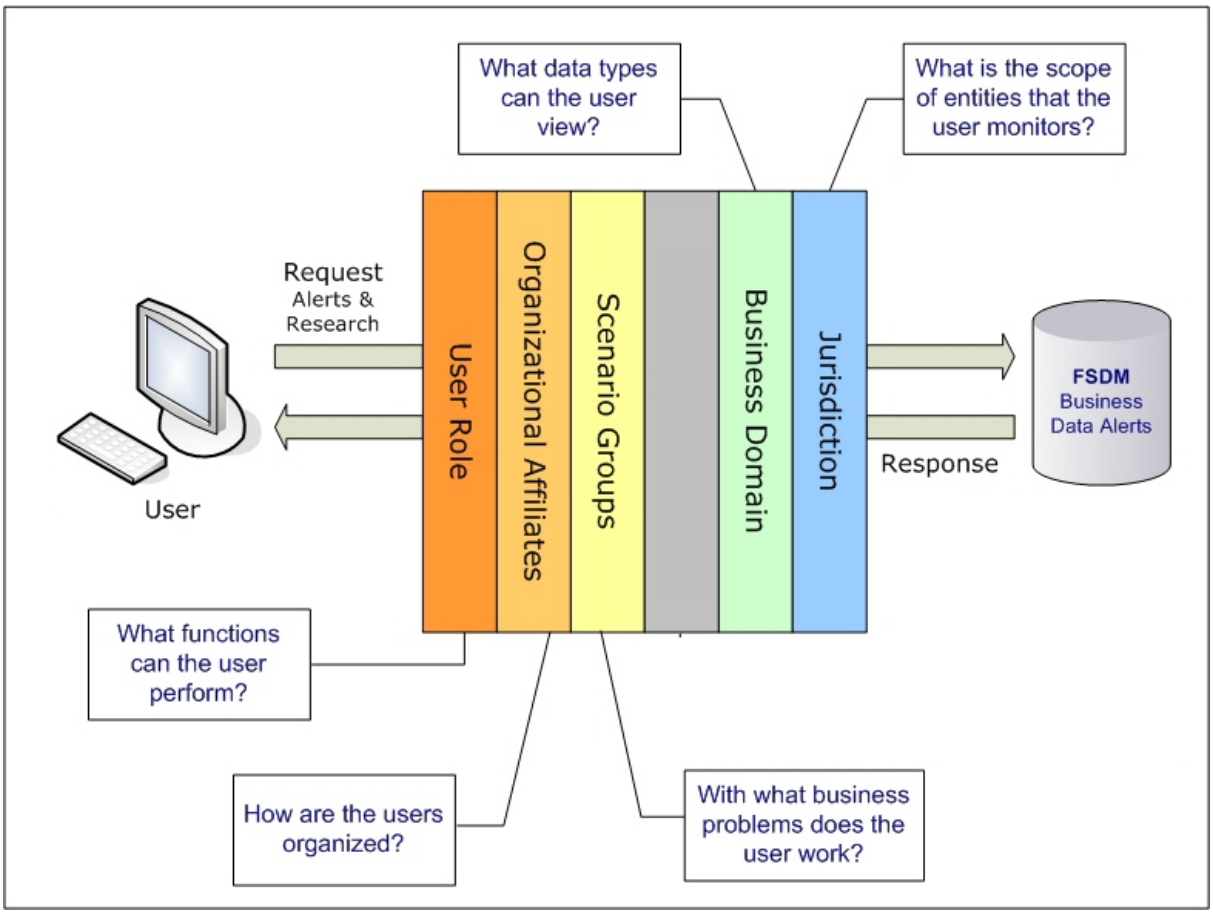


Figure 53: User Authorization Model

The following table provides the relationships between the data points that Figure 3 illustrates.

Table 52: Relationships between Data Points

Data Point	Relationship
Organization	• Root of a client's organization hierarchy • Associated with 0..n users as a line organization • Associated with 0..n users for view access to the organization • Associated with 1..n Business Domains • Associated with 1..n Scenario Groups • Associated with 1..n Case Type/Subtypes • Associated with 1..n Jurisdictions • Has no direct relationship with a Role
Role	• Associated with 0..n Users • Has no direct relationship with an Organization

Table 52: Relationships between Data Points (Continued)

Data Point	Relationship
User	• Associated with 1..n Business Domains • Associated with 1..n Jurisdictions • Associated with 1..n Roles • Associated with 1..n Scenario Groups • Associated with 1..n Case Type/Subtypes • Associated with 1..n Organizations (as members) • Associated with one Organization (as mantasLineOrgMember)
Users (Admin Tools)	• Should be mapped only to mantas Admin Role.
Scenario Group	• Associated to 0..n users • Associated with Scenarios referenced in KDD_SCNRO table.
Business Domains	• Associated to 0..n users • Business domain key must be in the KDD_BUS_DMN table
Jurisdiction	• Associated to 0..n users • Jurisdiction key must exist in the KDD_JRSDCN table

D Managing Data

This appendix covers the following topics:

D.1 CSA Ingestion

This section refers to Common Staging Area (CSA) ingestion and covers the following topics:

D.1.1 CSA Datamaps

The following list of files can be run using Common Area Staging. Files have been grouped in such a way that files in the same group can be executed in parallel to load data. However, you must execute Group 1 through Group 6 in sequence.

> **ATTENTION** You must run the Country and Customer data files before you run the other files.

Table 53: CSA Datamaps

Group	Logical Table Name	
1	Country	Customer
2	Account Phone Watch List Account Email Address Front Office Transaction	Account Customer Role Organization Customer Credit Rating Customer Identification Document
3	Account Watch List Entry	Front Office Transaction Party
4	Account To Customer Account Balance Account Address Customer To Markets Served Customer To Products Offered Customer To Customer Relationship	Customer Phone Customer Email Address Customer Country Customer Address Controlling Customer
5	Borrower Account Restriction Back Office Transaction	Investment Advisor Settlement Instruction Loan Origination Document Print Log

Table 53: CSA Datamaps

Group	Logical Table Name	
6	Trade Finance Contract Trade Finance Goods or Service Trade Finance Document Trade Finance Party Trade Finance Contract Event Acknowledgment Trade Finance Contract Amendment Status	Documentary Collection Contract Documentary Collection Contract Acknowledgment Documentary Collection Contract Acceptance Documentary Collection Discrepancy Detail Trade Finance to Account External Organization Goods or Service

D.1.2 Group Dependencies

Processing data in Group1 requires no prerequisite information (dependencies) for Pre-processing. Groups 2-5, however, rely on successful pre-processing of the previous group to satisfy any dependencies. For example, the Ingestion Manager does not run Group 4 until processing of data in Group 3 completes successfully.

Processing bases the dependencies that determine grouping on the referential relationships within the data. If the Oracle client chooses not to perform referential integrity checking, grouping is not required (except in some instances). In this case, a need still exists to process some reference data files prior to processing trading data.

D.2 Flat File Ingestion

This section refers to Behavior Detection () Ingestion Flat Files and covers the following topics:

- BDF.xml File Parameters
- TBAML Flat File Interface

D.2.1 BDF.xml File Parameters

The following table describes the parameters which must be configured in the BDF.xml file under the `<OFSAAI Installed Directory>/bdf/config` folder for processing DIS files.

Figure 54: Parameters Related to Processing DIS Files

Property Name	Description	Default
DIS.Source	Indicates the source of DIS records. Valid values are: • FILE for a DIS file • FSDW for CSA table loading • FILE-EXT for loading DIS file using an external table	FILE
DIS.ArchiveFlag	Indicates whether a DIS file should be archived after it has been processed.	true
DIS.BufferSize	Indicates the size of a byte buffer (in kilobytes) used to read in a line from a DIS file. This should be set to the maximum possible record size (in kilobytes) of a record in a DIS file.	100

Figure 54: Parameters Related to Processing DIS Files

Property Name	Description	Default
DIS.InputFileCharset	Indicates the character set of a DIS file.	UTF8
DIS.Default.Check.Requirement	Indicates whether the mandatory and conditional checks on a DIS record should be done	true
DIS.Default.Reject.Requirement	Indicates whether a mandatory or conditional check failure for a record should result in the record being rejected. If this is set to FALSE and a missing value is attempted to be inserted into a NOT NULL column, then the record will be rejected anyway.	true
DIS.Default.Check.Domain	Indicates whether the domain value checks on a DIS record should be done.	true
DIS.Default.Reject.Domain	Indicates whether a domain value check failure for a record should result in the record being rejected.	true
DIS.Default.Check.Length	Indicates whether the maximum length checks on a DIS record should be done.	true
DIS.Default.Reject.Length	Indicates whether a maximum length check failure for a record should result in the record being rejected. If this is set to FALSE, then the value will be truncated based on the maximum length of the field.	true
DIS.Default.Check.Threshold	Indicates whether the threshold checks (GREATER_THAN_ZERO, etc) on a DIS record should be done.	true
DIS.Default.Reject.Threshold	Indicates whether a threshold check failure for a record should result in the record being rejected.	true
DIS.Default.Check.Lookup	Indicates whether the reference data lookups on a DIS record should be done.	true
DIS.Default.Reject.Lookup	Indicates whether a reference data lookup failure for a record should result in the record being rejected.	true
MITrxnProducttypes	Indicates the parameter which is used to pass a list of product codes for trailing digit purpose (AUG_INSTR_NB derivation).	• CHECK • CHECK-ACH
CustProfileLookBack	Indicates the parameter which is used to look back at the days in Customer Summary Daily for Customer Summary Month recalculation. **Note:** In order to look back at a specific time period in Customer Summary Daily, you must have partitions available in Customer Summary Month.	31
CustAcctHolderType	Indicates the parameter which is used to identify customer account types to be included in customer summary.	CI

D.2.2 Ingest DIS Data Files by Group

Ingestion Manager processes data files in groups (in a specified order) from Oracle client data in the / inbox directory. The following list of files can be run using CSA. Files have been grouped in such a way that files in the same group can be executed in parallel to load data. However, you must execute Group 1 through Group 6 in sequence. The following table lists the data files by group.

Table 54: Ingest DIS Data Files By Group

Group	Data Files	
4.	Account Phone Watch List Account Email Address Front Office Transaction	Account Customer Role Organization Country
5.	Account Customer	Watch List Entry Front Office Transaction Party
6.	Account To Customer Account Balance Account Address Customer To Customer Relationship Customer Phone	Customer Email Address Customer Country Customer Address Controlling Customer
7.	Back Office Transaction	
8.	OpenOrder Order	TradeExecutionEvent

D.2.3 TBAML Flat File Interface

The following tables describe the Ingestion Flat File details for products within the TBAML Application Pack. Files have been grouped in such a way that files in the same group can be executed in parallel to load data. However, you must execute Group 1 through Group 5 in sequence. For more information, see Group Dependencies.

The Staging Representation column indicates whether this file requires a Staging source.

The following table describes the Group 1 Ingestion Flat File details.

Table 55: Group 1 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Account Phone	Datamaps	Yes
Account Email Address	Datamaps	Yes
Insurance Policy	Datamaps	Yes
Insurance Policy Balance	Datamaps	Yes
Account Customer Role	Datamaps	Yes
Insurance Policy Feature	Datamaps	Yes
Insurance Policy to Customer	Datamaps	Yes

Table 55: Group 1 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Loan	Datamaps	Yes
Loan Daily Activity	Datamaps	Yes
Online Account	Datamaps	Yes
Insurance Seller	Datamaps	Yes
Insurance Seller to License	Datamaps	Yes
Country	Datamaps	Yes
Watch List	Datamaps	Yes
Insurance Product	Datamaps	Yes
Insurance Transaction	Datamaps	Yes
Front Office Transaction	Datamaps	Yes
Organization	Datamaps	Yes
Market Center	Datamaps	Yes
Market Index Daily	Datamaps	Yes
Issuer	Datamaps	Yes
Market Index	Datamaps	Yes
Service Team Member	Datamaps	Yes
Service Team	Datamaps	Yes
CTR Transaction	runDP/runDL	No
Account Realized Profit and Loss	runDP/runDL	No
Letter of Intent	runDP/runDL	No
Collateral Value-Currency	runDP/runDL	No
Collateral Value-Product	runDP/runDL	No
Commission Product	runDP/runDL	No
Compliant Registration	runDP/runDL	No
Complaint Type Rating	runDP/runDL	No
Employee to Insurance Policy	runDP/runDL	No
Investment Guideline	runDP/runDL	No
Investment Guideline to Account	runDP/runDL	No
System Logon Type	runDP/runDL	No
Registered Representative Complaint	runDP/runDL	No
Energy And Commodity Instrument	runDP/runDL	No

The following table describes the Group 2 Ingestion Flat File details.

Table 56: Group 2 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Account to Peer Group	Datamaps	Yes
Account Group	Datamaps	Yes
Peer Group	Datamaps	Yes
Security Market Daily	Datamaps	Yes
Security Firm Daily	Datamaps	Yes
Security	Datamaps	Yes
Market Index Member Security	Datamaps	Yes
Security Market State Change	Datamaps	Yes
Matched Entity	runDP/runDL	No
Trusted Pair	Datamaps	Yes
Firm Account Position Pair	runDP/runDL	No
Natural Gas Flow	runDP/runDL	No

The following table describes the Group 3 Ingestion Flat File details.

Table 57: Group 3 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Account	Datamaps	Yes
Customer	Datamaps	Yes
Watch List Entry	Datamaps	Yes
Loan Product	Datamaps	Yes
Employee	Datamaps	Yes
Front Office Transaction Party	Datamaps	Yes
Organization Relationship	Datamaps	Yes
Restriction List	Datamaps	Yes
Automated Quote	Datamaps	No
Account Supplemental Attribute	runDP/runDL	No
Customer Supplemental Attribute	runDP/runDL	No
Market Trading Session	runDP/runDL	No
Account GroupAddress	runDP/runDL	No
Account Group Investment Objective	runDP/runDL	No
Account Group IOS Member	runDP/runDL	No

Table 57: Group 3 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Account Group Member Experience	runDP/runDL	No
Loan Origination Action	runDP/runDL	No
Mail Handling Instruction Activity	runDP/runDL	No
Banker To Officer	runDP/runDL	No
Reference Table Detail	runDP/runDL	No
General Usage List	runDP/runDL	No
Loan Origination Product	runDP/runDL	No
Organization To Mortgage Type	runDP/runDL	No
Securities License	runDP/runDL	No
Service Vendor	runDP/runDL	No
Energy and Commodity Trade	runDP/runDL	No

The following table describes the Group 4 Ingestion Flat File details.

Table 58: Group 4 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Market News Event	Datamaps	No
Managed Account	Datamaps	Yes
Account To Customer	Datamaps	Yes
Branch CTR Transaction	Datamaps	Yes
Branch CTR Conductor	Datamaps	Yes
Branch CTR Summary	Datamaps	Yes
Account Group Member	Datamaps	Yes
Account To Correspondent	Datamaps	Yes
Account Balance	Datamaps	Yes
Account Address	Datamaps	Yes
Customer Identification Document	Datamaps	Yes
Customer To Markets Served	Datamaps	Yes
Customer To Products Offered	Datamaps	Yes
Customer To Customer Relationship	Datamaps	Yes
Anticipatory Profile	Datamaps	Yes
Customer Phone	Datamaps	Yes
Customer Email Address	Datamaps	Yes

Table 58: Group 4 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Customer Country	Datamaps	Yes
Customer Address	Datamaps	Yes
Online Account to Account	Datamaps	Yes
Controlling Customer	Datamaps	Yes
Employee To Account	Datamaps	Yes
Account Position	Datamaps	Yes
Security Trading Restriction	Datamaps	Yes
Employee Trading Restriction	Datamaps	Yes
Employee Phone	Datamaps	Yes
Employee Email Address	Datamaps	Yes
Employee Address	Datamaps	Yes
Outside Business Activity	Datamaps	Yes
Private Security Transaction	Datamaps	Yes
Security Group Member	Datamaps	Yes
Security Investment Rating	Datamaps	Yes
Structured Deal	Datamaps	Yes
Account Profit and Loss	Datamaps	Yes
Account Position Pair	Datamaps	Yes
Account Investment Objective	Datamaps	Yes
Mutual Fund Breakpoint	Datamaps	Yes
Account Feature	runDP/runDL	No
Access Events	runDP/runDL	No
Customer Balance	runDP/runDL	No
Front Office Transaction Remittance Document	runDP/runDL	No
Related Front Office Transaction Information	runDP/runDL	No
Account To Organization	runDP/runDL	No
Firm Account Position	runDP/runDL	No
External Investment Account Position	runDP/runDL	No
Employee To Organization	runDP/runDL	No
Security Select List Entry	runDP/runDL	No
Account Fees	runDP/runDL	No
Account Profile Stage	runDP/runDL	No

Table 58: Group 4 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Account Qualification Agreement	runDP/runDL	No
Account Representative Position	runDP/runDL	No
Account Asset Allocation	runDP/runDL	No
Account Scheduled Event	runDP/runDL	No
Account Identifier Change History	runDP/runDL	No
Account Position Profile And Loss	runDP/runDL	No
Uncovered Option Account Position	runDP/runDL	No
Account Collateral	runDP/runDL	No
Mail Handling Instruction	runDP/runDL	No
Mutual Fund Family Letter of Intent	runDP/runDL	No
Employee Disciplinary Action	runDP/runDL	No
Employee Exam History	runDP/runDL	No
Employee Firm Transfer History	runDP/runDL	No
Employee Securities License State Registration	runDP/runDL	No
Employee Supervision List	runDP/runDL	No
Employee To Manager History	runDP/runDL	No
Employee To Securities License	runDP/runDL	No
Employment History	runDP/runDL	No
System Logon	runDP/runDL	No
Plan of Solicitation	runDP/runDL	No
Mutual Fund Family Configuration	runDP/runDL	No
Energy And Commodity Market Daily	runDP/runDL	No
Energy And Commodity Firm Daily	runDP/runDL	No
Energy And Commodity Reported Market Sale	runDP/runDL	No
Energy And Commodity Market Trading Session	runDP/runDL	No
Energy And Commodity Market Center	runDP/runDL	No
Energy And Commodity Location	runDP/runDL	No
Energy Flow Mode	runDP/runDL	No
Energy and Commodity Instrument Position	runDP/runDL	No

The following table describes the Group 5 Ingestion Flat File details.

Table 59: Group 5 Interface Ingestion Flat Files

Interface File Name	Current Ingestion	Staging Representation
Borrower	Datamaps	Yes
Back Office Transaction	Datamaps	Yes
Account Restriction	Datamaps	Yes
Investment Advisor	Datamaps	Yes
Investment Guideline Override	Datamaps	Yes
Settlement Instruction	Datamaps	Yes
Loan Origination Document Print Log	Datamaps	Yes
Change Log	runDP/runDL	No
Options Violation	runDP/runDL	No
Loan Origination Condition	runDP/runDL	No
Loan Origination Fee Detail	runDP/runDL	No
Loan Origination Note	runDP/runDL	No
Loan Origination To Service	runDP/runDL	No
Investment Guideline Override	runDP/runDL	No
Loan Origination Condition Type	runDP/runDL	No
System Logon To System Logon Type	runDP/runDL	No
System Logon To Organization	runDP/runDL	No
Registered Representative Account Commission	runDP/runDL	No
Registered Representative Account Commission Prior Year	runDP/runDL	No
Registered Representative Commission Monthly Profile	runDP/runDL	No
Registered Representative Commission Product	runDP/runDL	No
Currency Transaction	Datamaps	Yes

The following table describes the Group 6 Ingestion Flat File details.

Table 60: Group 6 Interface Ingestion for Market Data

Interface File Name	Current Ingestion	Staging Representation
Inside Quote	Datamaps	Yes
Market Center Quote	Datamaps	Yes

Table 60: Group 6 Interface Ingestion for Market Data

Interface File Name	Current Ingestion	Staging Representation
ReportedMarketSale	Datamaps	Yes
InsideQuote_Derived	Datamaps	Yes
MarketCenterQuote_Derived	Datamaps	Yes
ReportedMarketSale_Derived	Datamaps	Yes

The following table describes the Group 7 Ingestion Flat File details.

Table 61: Group 7 Interface Ingestion for Trade Finance Data

Interface File Name	Current Ingestion	Staging Representation
TradeFinanceContractEventAcknowledgement	Datamaps	Yes
TradeFinanceContractAmendmentStatus	Datamaps	Yes
TradeFinanceContract	Datamaps	Yes
TradeFinancetoAccount	Datamaps	Yes
TradeFinanceDocument	Datamaps	Yes
TradeFinanceGoodorService	Datamaps	Yes
TradeFinanceParty	Datamaps	Yes
DocCollectionContractAcknowlegementStage	Datamaps	Yes
DocumentaryCollectionContractAcceptacneStage	Datamaps	Yes
DocumentaryCollectionDiscrepancyDetail	Datamaps	Yes
DocumentaryCollectionContractEvent	Datamaps	Yes

D.3 Directory Structure

The Datamap component is organized as subdirectories below the `<OFSAAI Installed Directory>/bdf` file. The following table provides details about each subdirectory..

Table 62: Directory Structure Description

Directory Name	Description
scripts	Shell scripts for running components, setting the environment, and changing passwords
logs	Log files containing status and error messages produced by components
config	Files used to configure components
config/datamaps	XML files containing data map definitions for individual components
jars	Java Archive (JAR) files used to run components

Table 62: Directory Structure Description

Directory Name	Description
data/errors	Files containing error records produced by components
data/temp	Temporary files produced by components
inbox	Data files provided by the Oracle client in DIS format
fuzzy_match	C++ library files used for the purpose of fuzzy matching names

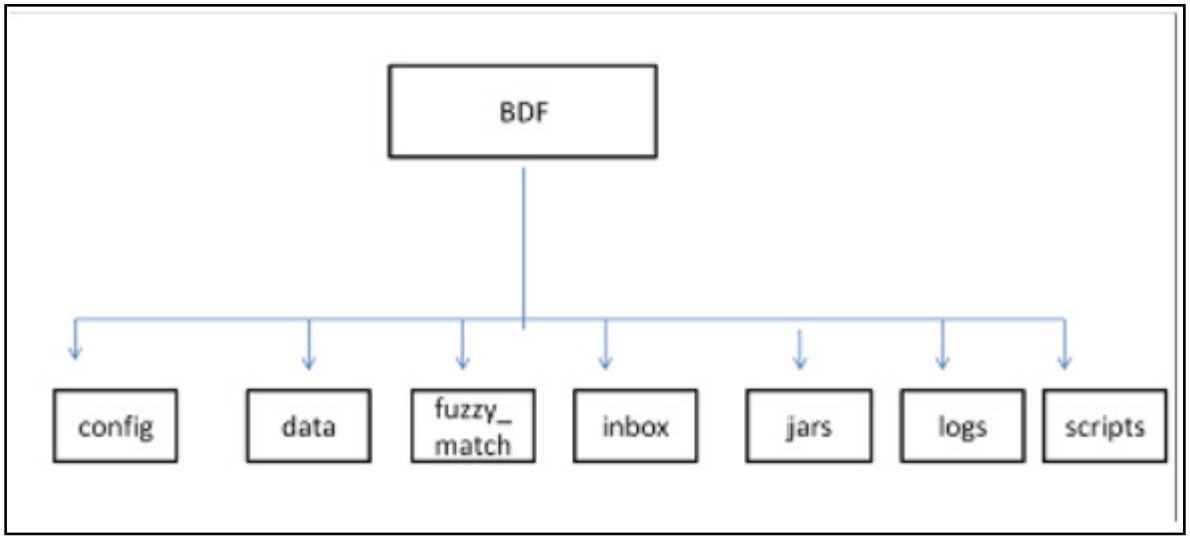


Figure 55: Subsystem Directory Structure

The following sections describe the directory structure.

D.3.0.1 Scripts

The scripts folder contains the following files:

- **changePassword.sh -** Changes passwords used during the execution of components. Refer to the *Installation Guide* for more information.
- **env.sh -** Sets ups the shell environment of components
- **execute.sh -** Executes components.

For Example:

```
<OFSAAI Installed Directory>/bdf/scripts/execute.sh <component>
```

```
<OFSAAI Installed Directory>/bdf/scripts/execute.sh CorrespondentBankProfile
```

> **NOTE** *Component* in this document means a batch process which is part of the Datamap subsystem. For the most part, these components will refer to XML data maps. For example, the AccountProfile_Balance component refers to the AccountProfile_Balance.xml data map.

Running these files in the subsystem improves performance time.

D.3.0.2 Logs

The log file has information about the warnings, errors, and status of the component. Additional information can be obtained from a component by turning on diagnostic logging. This can be done by setting the `Log.DIAGNOSTIC.Enabled` parameter to true. In a production environment, this should be left as false and only changed to true when debugging errors or performance issues.

Log files for each component are written to a log file named for the component inside a subdirectory of the logs directory named for the current processing date in YYYYMMDD format:

For example:

```
<OFSAAI Installed Directory>/bdf/logs/<processing date>/<component>.log
```

```
<OFSAAI Installed Directory>/bdf/logs/20130313/CorrespondentBankProfile.log
```

When SQL*Loader is the loading mechanism, as shown below, there are additional log files containing log output from the SQL*Loader utility named the same as the component's log file with "_N" extensions (where **N** is an integer).

For example:

```
<OFSAAI Installed Directory>/bdf/logs/20130313/CorrespondentBankProfile_0.log
```

```
<OFSAAI Installed Directory>/bdf/logs/20130313/CorrespondentBankProfile_1.log
```

When an external table is used as the DIS file loading mechanism, there are additional log files containing log output from the external table utility. The log files are named the same as the external table being loaded. The name of the external table is the name of the table being loaded with a prefix of "DIS_". For example, when loading the ACCT table, the external table log file will be:

```
<OFSAAI Installed Directory>/bdf/logs/20130313/DIS_ACCT.log
```

D.3.0.3 Parameters

Parameters in TBAML Datamaps are specified as elements in an XML file. The XSD containing a description of these elements can be found in the following directory:

```
<OFSAAI Installed Directory>/bdf/config/ParameterSet.xsd
```

The Parameter element defines a parameter and its value, and contains the following attributes:

- **name -** The name of the parameter.
- **type -** The data type of the parameter. Valid values are STRING, REAL, INTEGER, BOOLEAN, FILE, and CLASS.
- **value -** The value of the parameter, which must map the type of the parameter.
- **list -** A boolean value specifying that the value is a single value (false - the default) or a comma separated list of values (true).

For example:

```
<Parameter name="MinimumGeographyRisk" type="INTEGER" value="0"/>
```

```
<Parameter name="InternalAccountCodeList" type="STRING" value="IA,GL"
list="true"/>
```

> **NOTE** If the value of the parameter is a string containing characters which are not allowed in an XML attribute, then a CDATA element can be used as the element's text. For example:
>
> ```
> <Parameter
> name="PassThruExpressionSeparators"
> type="STRING">
> <![CDATA[~: \t/#-]]>
> </Parameter>
> ```

Parameters in the main BDF.xml file should not be modified. Instead, any customizations to parameter values should be placed in the <OFSAAI Installed Directory>/bdf/config/custom/BDF.xml file. Parameters can be overridden at the component level by placing them in the custom/<component>.xml file. Also, parameters can be overridden on the command line by passing the parameter name and value as parameters to the execute.sh script after the component name:

For example:

```
<OFSAAI Installed Directory>/bdf/scripts/execute.sh <component> [parameter
name=value]*
```

```
<OFSAAI Installed Directory>/bdf/scripts/execute.sh CorrespondentBankProfile
NumberOfThreads=4
```

When a given parameter is read by a component, the order of precedence for where the parameter value is taken from is as follows:

```
    command line
<OFSAAI Installed Directory>/bdf/config/custom/<component>.xml
<OFSAAI Installed Directory>/bdf/config/<component>.xml
<OFSAAI Installed Directory>/bdf/config/custom/BDF.xml
<OFSAAI Installed Directory>/bdf/config/BDF.xml
```

D.3.0.4 Config

The config subdirectory contains configuration files.

- `<OFSAAI Installed Directory>/bdf/config/BDF.xml` contains all default product configuration parameters. It should not be modified.
- `<OFSAAI Installed Directory>/bdf/config/install/BDF.xml` contains all configuration parameters set at installation time (refer to the *Installation Guide* for more information).
- `<OFSAAI Installed Directory>/bdf/config/custom/BDF.xml` contains any product configuration parameters that have been overridden for this installation. It is initially empty. Any changes to default product configuration parameters should be put here.

Individual components can have their own configuration file which overrides default product parameters. These files would be named using the following format:

```
<OFSAAI Installed Directory>/bdf/config/<component>.xml
```

For example:

```
<OFSAAI Installed Directory>/bdf/config/CorrespondentBankProfile.xml
```

Component configuration files in this directory are part of the product and should not be modified. If any parameters must be overridden at the individual component level, the component configuration file should be created in `<OFSAAI Installed Directory>/bdf/config/custom`.

- The datamaps subdirectory contains XML files holding the data map definitions for components.
- The derivations subdirectory contains SQL derivations for individual fields.
- The queries subdirectory contains SQL queries for individual data maps.

D.3.0.4.1 BDF.xml Configuration Parameters

The following table describes the properties configurations mentioned in the `<OFSAAI Installed Directory>/bdf/config/BDF.xml` file.

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
MISCELLANEOUS		
NumberOfThreads	The number of worker threads used by some components	4
SequenceBatchSize	The batch size when retrieving sequence IDs for new records	100000
SourceSystem	he default value for source system when one is not provided	MTS
Currency	The default value for issuing currency when one is not provided	USD
Separator	The delimiter that separates fields in data file records.	~
DB: **Parameters related to database access.**		
DB.Connection.Driver	The JDBC driver class name.	oracle.jdbc.OracleDriver
DB.Timeout	The number of seconds to wait before timing out on a database connection attempt.	10
DB.NumRetries	The maximum number of times to attempt to connect to a database before failing.	5
DB.MaxNumberOfDeadlocks	The maximum number of times a deadlock is encountered during a JDBC insert or update operation, before an error is generated.	10
Directory: **Parameters used to define directory locations.**		
Directory.Inbox	The input directory where the Oracle client will write DIS files. Date subdirectories will be created in this directory where these files will be archived	../inbox
Directory.InternalData	The directory where files generated by components will reside. This includes log files, error files, and any temporary processing files.	..
Log: **Parameters used to configure the common logging module**		
Log.Format	Identifies the log formatting string.	%d [%t] %p - %m%n

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
Log.UseDefaultLog	Specifies whether the system uses the default log file for a component. The default log file has the name of the component and resides in a date subdirectory of the logs directory (in YYYYMMDD format).	true
Log.SysLogHostName	The host name of syslog for messages sent to syslog.	hostname
Log.SMTPHostName	The host name of the SMTP server for messages that processing sends to an e-mail address.	hostname
Log.MaxSize	The maximum size (in MB) of a log file before the system creates a new log file.	2000MB
Log.MaxIndex	If a log file exceeds Log.MaxSize, this will be the maximum number of additional log files that are created (Component.log.1, Component.log.2, etc).	10
Log.TRACE.Enabled	Indicates that trace logging is not enabled; true indicates enabling of trace logging.	false
Log.TRACE.Location	Specifies additional locations to send TRACE log messages to, other than the default log file (logs/YYYYMMDD/Component.log). If the value is not provided, considers the default log location.	false
Log.TRACE.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
Log.DIAGNOSTIC.Enabled	DIAGNOSTIC logging is used to log database statements and will slow down performance. Make it true if needed.	false
Log.DIAGNOSTIC.Location	Additional locations to send DIAGNOSTIC log messages to, other than the default log file (logs/YYYYMMDD/Component.log). If the value is not provided, considers the default log location.	
Log.DIAGNOSTIC.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
Log.NOTICE.Enabled	Indicates enabling of notice logging; false indicates that notice logging is not enabled.	true
Log.NOTICE.Location	Specifies additional locations to send NOTICE log messages to, other than the default log file (logs/YYYYMMDD/Component.log). If the value is not provided, considers the default log location.	
Log.NOTICE.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
Log.WARN.Enabled	Indicates enabling of warning logging; false indicates that warning logging is not enabled.	true
Log.WARN.Location	Specifies additional locations to send WARN log messages to, other than the default log file (logs/YYYYMMDD/Component.log).	

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
Log.WARN.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
Log.FATAL.Enabled	Indicates enabling of Fatal logging; false indicates that fatal logging is not enabled.	true
Log.FATAL.Location	Specifies additional locations to send FATAL log messages to, other than the default log file (logs/YYYYMMDD/Component.log).	
Log.FATAL.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
Load: **Parameters used to configure common Loading data**		
Load.FullRefresh	For DIS files defined as Overwrite, whether to fully replace FCDM tables with the contents of the DIS file (true) or to treat the DIS file as a delta (false)	True
Load.BatchSize	The batch size when loading data.	5000
Load.Direct	Specifies whether to use direct path loading (TRUE) or conventional path loading (FALSE).	false
Load.Unrecoverable	Specifies whether a direct path load does not use redo logs (TRUE) or uses redo logs (FALSE).	false
Load.Partitioned	Specifies whether a direct path load uses the current date partition (TRUE) or any partition (FALSE).	false
Load.SkipIndexes	Specifies whether a direct path load skips index maintenance (TRUE) or maintains indexes (FALSE). If set to TRUE, rebuilding of indexes must occur after running the DataMap XML.	false
Load.DoAnalyze	Specifies whether to run a stored procedure to analyze a database table after loading data into it.	true
Load.AnalyzeType	Specifies the type of analyze statistics has to perform if DoAnalyze has a value of True.	DLY_POST_LOAD
Load.LogRecordInterval	Specifies how often to log a message saying how many records a particular thread has inserted/updated,	1000
Load.MaxErrorRate	Specifies the percentage of invalid records to allow before exiting with an error. For example, a value of 10 allows 10 percent of records to be invalid before exiting with an error. A value of 0 allows no invalid records. A value of 100 allows all invalid records.	100
Load.RecordQueueSize	Specifies the number of records the query reader thread will write to a database writer thread queue before waiting for the reader thread to catch up. Higher values will require more memory usage.	100
Load.SkipIndexesErrorCode	Specifies a database error code that occurs in the log file when skipping index maintenance.	26025

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
Load.IndexParallelLevel	Specifies the parallel level of an index rebuild (that is, number of concurrent threads for rebuilding an index).	1
Load.DataErrorCodes	Specifies a comma-separated list of database error codes that indicate data level errors , such as data type and referential integrity. This results in rejection of records with a warning instead of a fatal failure.	1,1400,1401,1407,1438,1722,1840,1841,2291,2359,1839,1847,12899
Load.ParallelLevel	Specifies the level of parallelization to apply when loading data from a set of source tables to a target table.	8
Load.WriteErrorFiles	Whether to check a DIS file for errors before loading as an external table (true) or not (false)	True
DIS: **Parameters related to processing DIS files**		
DIS.Source	The mechanism used to load DIS data. FILE: DIS files will be provided and will be loaded using SQL*Loader processes running on the application server. FILE-EXT: DIS files will be provided and will be loaded using external tables with the DIS files accessed directly by the database. FSDW: DIS data will be obtained from database tables in the FSDW.	FILE
DIS.ArchiveFlag	Whether DIS files will be archived to a date subdirectory (true) or not (false).	True
DIS.BufferSize	The size in KB of the byte buffer used to read in DIS file records.	100
DIS.InputFileCharset	The character set of the DIS files. Note that output data is always written in UTF8, this parameter just allows the DIS files to be in a different character set.	
DIS.Default.Check.Requirement	Whether to check for mandatory fields on DIS records (true) or not (false).	True
DIS.Default.Reject.Requirement	Whether to reject DIS records for failing a mandatory field check (true) or to log a warning and attempt to load the record (false).	True
DIS.Default.Check.Domain	Whether to check that a DIS field has a valid domain value (true) or not (false).	True
DIS.Default.Reject.Domain	Whether to reject DIS records that fail a domain check (true) or not (false).	True
DIS.Default.Check.Length	Whether a DIS field should be checked for a valid length (true) or not (false).	True
DIS.Default.Reject.Length	Whether to reject DIS records that fail a length check (true) or not (false)	True
DIS.Default.Check.Threshold	Whether a DIS field should be checked that it is within an acceptable threshold (i.e. greater than 0) (true) or not (false).	True

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
DIS.Default.Reject.Threshold	Whether to reject DIS records that fail a threshold check (true) or not (false).	True
DIS.Default.Check.Lookup	Not currently supported.	True
DIS.Default.Reject.Lookup -	Not currently supported	True
Parameters used by queries defined in the data maps:		
MinimumGeographyRisk	Defines what is considered High Risk For the Account Profile attributes related to High Risk Geography , such as Incoming High Risk Wire Count. Processing compares this parameter using a strict greater-than operation.	0
AccountInactivityInMonths	Specifies the number of months that processing aggregated to determine whether an account is inactive. If the sum of trades and transactions over this number of months is <= 3, the account is considered inactive. This setting can impact the Escalation in Inactive Accounts scenario. The default value is six months.	6
TransactionsReversalLookbackDays	This parameter controls how many days of transactions to look across. Verify whether the new data contains reversals of prior transactions.	7
LowPriceSecurityThreshold	Defines Low Priced in the base currency for the Account Profile attributes named Low-Priced Equity Range # Opening Trade Count. Processing compares the value of this parameter to the Trade table's Last Execution Price-Base.	5000
CommissionEquityPercentUpperLimit	Defines the upper limit for Commission Versus Average Daily Equity Percentage in Account Profile Calculation.	5
TurnOverRateUpperLimit	Defines the upper limit for Total Turnover Rate in Account Profile Calculation.	5

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
BankCodeListWithIA	Defines the List of Financial Institution Identifier Types, these are type of unique identifiers which are used to represent the financial institutions. This parameter also contains IA (Internal Account Identifier) to be used in datamaps and is mainly used in Correspondent Bank related datamap derivations. Below are the list of examples • BIC: Bank Identifier Code (BIC) • CHU: CHIPS Participant User Identifier • CO: Corporate Identifier • CHP: CHIPS Participant Identifier • FED: Federal Reserve Routing (ABA) Number • CU: Customer Identifier • GL: General Ledger Account • IA: Internal Account Identifier	BIC,FED,CHP ,CHU, DTC,CDL,EP N,KID, CBI,CSN,OTF ,BLZ,I BAN,ABLZ,B SB,CP AP, SDIC, HEBIC, BCHH, NSC, IFSC, IDIC, PNCC, RCBIC, UKDSC, Swiss BC, Swiss SIC,IA
BankCodeList	Defines the List of Financial Institution Identifier Types, these are type of unique identifiers which are used to represent the financial institutions excluding Internal Account (IA). This parameter does not contain IA (Internal Account Identifier) to be used in datamaps and is typically used to derive financial institutions. Below are the list of examples • BIC: Bank Identifier Code (BIC) • CHU: CHIPS Participant User Identifier • CO: Corporate Identifier • CHP: CHIPS Participant Identifier • FED: Federal Reserve Routing (ABA) Number • CU: Customer Identifier • GL: General Ledger Account	BIC,FED,CHP ,CHU, DTC,CDL,EP N,KID, CBI,CSN,OTF ,BLZ,I BAN,ABLZ,B SB,CP AP, SDIC, HEBIC, BCHH, NSC, IFSC, IDIC, PNCC, RCBIC, UKDSC, Swiss BC, Swiss SIC
IdRiskWinLevel	Defines the Risk level to calculate Effective Risks for internal parties (Account/ Customer). For example: Account 1234 has an Effective Risk of 5, IdRiskWinLevel can be set by the client. If the party identifier effective risk is greater than the set IdRiskWinLevel, then the party identity risk wins compared to fuzzy matcher (Party Name Risk). If not, fuzzy matcher wins.	1
InternalAccountCodeList	Codes to define types of Internal Entities with client, for example: • IA: Internal Account Identifier • GL: General Ledger Account	IA, GL

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
ExternalEntityCodeList	Codes to define types of External Entities with client, for example: • XA: External Account Identifier • CO: Corporate Identifier • DL: Driver License • IBAN: International Bank Account Number	XA,CC,CO,DL,GM, GP,LE,MC,ND,NR, PP,SS,TX,AR, OT,IB AN
TrustedPairReviewReasonText1	Defines the reason text1 for recommendation of cancelling the Trusted Pair, due to increase in Risk of parties involved in trusted pair.	Risk of <Party1> increased from <A> to <b>
TrustedPairReviewReasonText2	Defines the reason text2 for recommendation of cancelling the Trusted Pair, due to increase in Risk of parties involved in trusted pair.	Risk of <Party2> increased from <C> to <D>
CorporateActionLookBackDays	This parameter determines the how many days trades to look back from the Corporate Effective Date.	7
DealNearTermMaturityDays	Defines the maximum number of days between the End Date and Trade Date. This helps to calculate Structured Deals Initiated w/ Near-Term Exp. In Customer Profile/ Institutional Account Profile.	7
ProfitLossUpperLimit	Helps determine how much a security must move by the end of the day to be considered a win or loss. If the security moves by less than a specified percentage, processing does not count it either way. If it moves by this percentage or more, it counts as a win or a loss, depending on whether the movement was beneficial to the account that made the trade.	5
HouseholdTurnOverRateUpperLimit	Defines the upper limit for Total Turnover Rate in Household Profile Calculation.	10000
HouseholdCommissionEquityPercentUpperLimit	Defines the upper limit for Commission Versus Average Daily Equity Percentage in Account Profile Calculation.	10000
OptionTradeAmountRange1 OptionTradeAmountRange2 OptionTradeAmountRange3 OptionTradeAmountRange4 OptionTradeAmountRange5 OptionTradeAmountRange6	Define the lower bound of each range for the Account Profile attributes named Options Range # Opening Trade Count. Processing compares each parameter to the Trade table's Last Principal Amount- Base. Each range is from the lower bound entered here to the lower bound of the next range.	

Table 63: BDF.xml File Configuration Parameters

Parameter Name	Description	Example
EquityTradeAmountRange1 EquityTradeAmountRange2 EquityTradeAmountRange3 EquityTradeAmountRange4 EquityTradeAmountRange5 EquityTradeAmountRange6	Define the lower bound of each range for the Account Profile attributes named Equity Range # Opening Trade Count. Processing compares each parameter to the Trade table's Last Principal Amount- Base. Each range is from the lower bound entered here to the lower bound of the next range.	
LowPricedEquityTradeAmountRange1 LowPricedEquityTradeAmountRange2 LowPricedEquityTradeAmountRange3 LowPricedEquityTradeAmountRange4 LowPricedEquityTradeAmountRange5 LowPricedEquityTradeAmountRange6	Define the lower bound of each range for the Account Profile attributes named Low-Priced Equity Range # Opening Trade Count. Processing compares each parameter to the Trade table's Last Principal Amount-Base. Each range is from the lower bound entered here to the lower bound of the next range.	
MutualFundTradeAmountRange1 MutualFundTradeAmountRange2 MutualFundTradeAmountRange3 MutualFundTradeAmountRange4 MutualFundTradeAmountRange5 MutualFundTradeAmountRange6	Define the lower bound of each range for the Account Profile attributes named Mutual Fund Range # Opening Trade Count. Processing compares each parameter to the Trade table's Last Principal Amount-Base. Each range is from the lower bound entered here to the lower bound of the next range.	
UnrelatedWhenOffsetAccountIsNull	This parameter is used to assign unrelated party code as "J" in the BackOfficeTransaction table, If OFFST_ACCT_INTRL_ID is null and UnrelatedWhenOffsetAccountIsNull is "Y", If OFFST_ACCT_INTRL_ID is null and UnrelatedWhenOffsetAccountIsNull is "N", then unrelated party code is NULL.	Y

D.3.0.4.2 Datamap Configuration File

Oracle clients can modify the BDF.xml file under the bdf/config/custom folder to override default settings that the system provides. You can also reapply any modifications in the current BDF.xml file to the newer BDF.xml file.

Override any settings in BDF.xml by placing the modifications in BDF.xml under the bdf/config/custom folder.

During installation, the following parameters are configured by the installer:

- AccountTrustFromCustomer
- DefaultJurisdiction
- UseTaxidForUnrelatedPartyCode
- BaseCountry
- ProcessForeignFlag
- ProcessBankToBank
- ProcessTransactionXRefFlag
- TrustedPairRiskReviewFlag

These parameters are stored in the following file:

<OFSAAI Installed Directory>/bdf/config/install/BDF.xml

Parameters DefaultJurisdiction and BaseCountry are defined in the InstallConfig.xml file during Silent Installation. Refer to the *Installation Guide* for more information.

The Installer sets the default value for other parameters as follows:

- <Parameter name="AccountTrustFromCustomer" type="STRING" value="Y"/>
- <Parameter name="DefaultJurisdiction" type="STRING" value="AMEA"/>
- <Parameter name="UseTaxidForUnrelatedPartyCode" type="STRING" value="Y"/>
- <Parameter name="BaseCountry" type="STRING" value="US"/>
- <Parameter name="ProcessForeignFlag" type="STRING" value="N"/>
- <Parameter name="ProcessBankToBank" type="STRING" value="N"/>
- <Parameter name="ProcessTransactionXRefFlag" type="STRING" value="Y"/>
- <Parameter name="TrustedPairRiskReviewFlag" type="STRING" value="N"/>

To change the default value of these parameters, before running ingestion, go to <OFSAAI Installed Directory>/bdf/config/install/BDF.xml and change the value to 'Y' or 'N' as needed.

The following table describes the parameters defined in BDF.xml:

Table 64: Datamap Configuration Parameters

Property Name	Description	Example
DB.Connection.URL	Database URL for JDBC connections made by components. The content and format of this value is specific to the database vendor and the vendor database driver.	jdbc:oracle:thin:@solitaire.mantas.com:1521:D1O9L2
DB.Connection.Instance	Database instance to connect to on the database servers. Typically, the instance name matches the database name portion of the DB.Connection.URL.	D1O9L2
DB.Connection.Password	Password that Java Ingestion components use when connecting with the database. This is set by executing bdf/scripts/changepassword.sh	

Table 64: Datamap Configuration Parameters

Property Name	Description	Example
DB.Schema.MANTAS	Schema name for the Oracle ATOMIC database schema. accesses the ATOMIC schema when allocating sequence IDs to ingested records.	ATOMIC
DB.Schema.MARKET	Schema name for the ATOMIC database schema. Data Management stores market data related records in the ATOMIC schema.	ATOMIC
DB.Schema.BUSINESS	Schema name for the ATOMIC database schema. Data Management stores business data related records in the ATOMIC schema.	ATOMIC
DB.Schema.CONFIG	Name of the configuration schema owner.	REVELEUS
DB.Schema.CASE	Name of the ATOMIC schema owner.	ATOMIC
DB.Alg.Connection.User	Database user for running Behavior Detection post-processing jobs.	ATOMIC
DB.Alg.Connection.Password	Password for the DB.Alg.Connection.User.	

There are also configuration files for individual components that are delivered as part of the product:

```
<OFSAAI Installed Directory>/bdf/config/<component>.xml
```

And can also be created in the following directory:

```
<OFSAAI Installed Directory>/bdf/config/custom/<component>.xml
```

E Processing Derived Tables and Fields

This appendix covers the following topics:

E.1 Customizing Scripts

For OFSAAI to execute the shell scripts, the customized scripts have to be placed in the ficdb layer. The customized scripts should be placed under `<Installed Path>ficdb/bin`. When the customized scripts are called from OFSAAI, it appends the Batch Flag and Wait Flag parameters. This must be internally handled in the customized script to eliminate these additional parameters.

> **NOTE** The Batch Flag and Wait Flag are the default parameters expected by the AAI Batch. For more information on these parameters refer the Oracle Financial Services Analytical Applications Infrastructure User Guide.

The following paths should be set inside the scripts:

- **MANTAS_HOME**: The path where the solution is installed.

  For Example: `/scratch/ofsaaapp/FCCM806`

- **INGESTION_HOME**: The path under installed area pointing to the ingestion_manager subsystem.

  For Example: `/scratch/ofsaaapp/FCCM806/ingestion_manager`

- **DB_TOOLS_HOME**: The path under installed area pointing to database subsystem.

  For Example: `/scratch/ofsaaapp/FCCM806/database/db_tools`

- **BDF_HOME:** The path under the installed area pointing to the subsystem.

  For Example: `/scratch/ofsaaapp/FCCM806/bdf`

> **NOTE** BDF_HOME should be exported only if Ingestion has to be run through the subsystem.

After exporting the respective paths inside the script, the product script must be called from the customized script. For more information about how to create an OFSAA Batch and add a task for executing the custom script, please refer to the Oracle Financial Services Analytical Applications Infrastructure User Guide.

See the following sample customized script for execute.sh:

```
#!/bin/sh

if [[ $# == 0 || $# > 3 ]]; then
```

```
  ##echo "Usage: run_GD_dpdl.sh YYYYMMDD"
  exit -1;
fi
export MANTAS_HOME=/scratch/ofsaadb/BD_801_BUILD2/BD_801C2WL
export BDF_HOME=$MANTAS_HOME/bdf
export DB_TOOLS_HOME=$MANTAS_HOME/database/db_tools
##export DIS_FILES=$HOME/GD_Scripts/disfile.cfg
export FILE_NAME=$1
$BDF_HOME/scripts/execute.sh $FILE_NAME
      err=$?
      if [ $err -ne 0 ]
      then
        echo " BDF Execution failed"
        exit 1
      fi
```

This script is used to trigger Ingestion using execute.sh. This script expects only the file name (such as, Account) as a parameter. Since the AAI batch appends two additional default parameters (Batch Flag and Wait Flag) during batch execution, these should be handled inside the script and only the file name should be passed as a parameter. Internally this customized script calls the product script, `execute.sh`. Similarly, other scripts can also be customized.

E.2 Derivations

These utilities populate a single table in the data model. They should be executed after all the files have been loaded. A utility should not be executed until its predecessors have executed successfully.

Commands to execute:

```
<OFSAAI Installed Directory>/ingestion_manager/scripts/runUtility.sh <Utility Name>
```

```
<OFSAAI Installed Directory>/ingestion_manager/scripts/runDL.sh <Utility Name>
```

These commands should be run serially. The utility has executed successfully only after both of these commands have successfully executed.

Table 65: Utilities

Product	Utility Name	Table Name	Predecessor
ECTC	EnergyAndCommodityFirmDailyDerived	EC_FIRM_DAILY	
ECTC	EnergyAndCommodityMarketDailyDerived	EC_MARKET_DAILY	
ECTC	EnergyAndCommodityTradeDerived	EC_TRADE	
ECTC	EnergyFlow	ENERGY_FLOW	

Table 65: Utilities

Product	Utility Name	Table Name	Predecessor
BC	MutualFundFamilyAccountPosition	MUTUAL_FUND_FAM_ACCT_POSN	
BC	RegisteredRepresentativeCommissionProfile	RGSTD_REP_CMSN_SMRY	
BC	RegisteredRepresentativeCommissionProductMixProfile	RGSTD_REP_CMSN_PRDCT_SMRY	
ECTC	EnergyFlowDailyProfile	ENERGY_FLOW_SMRY_DAILY	Energy Flow

E.2.1 AccountDailySecurityProfile

The AccountDailySecurityProfile Utility is used to populate the Account Daily Security Profile table.

This Utility reads the Trade table, and processes the trade records to populate the `ACCT_SCRTY_SMRY_DAILY` table.

Execute the following commands:

```
runUtility.sh <Utility Name>
```

```
runDL.sh <Utility Name>
```

While executing these commands, replace `<Utility Name>` with `AccountDailySecurityProfile`

Example:

```
runUtility.sh AccountDailySecurityProfile
```

```
runDL.sh AccountDailySecurityProfile
```

E.3 Ingestion Timeline - Intra-Day Ingestion Processing

The following figure provides a high-level flow of the intra-day ingestion process of extracting, transforming, and loading data.

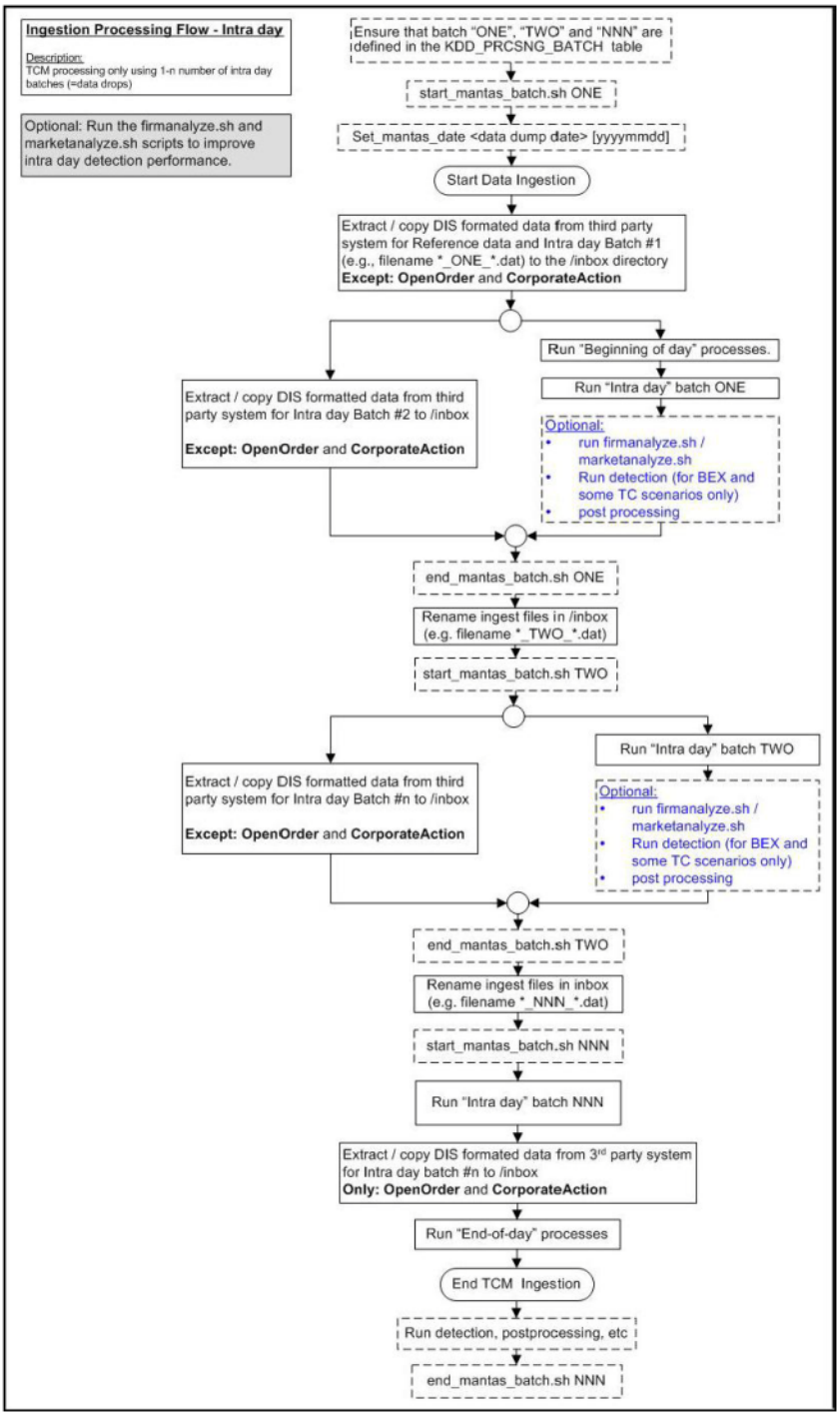


Figure 56: Intra-Day Data Management Processing

Intra-day processing references different processing groups as Figure 56 illustrates , such as beginning-of-day processing and intra-day processing. Multiple batches run throughout the day. As in Figure 56, you configure batch ONE, load and extract data, and then start processing. (Data for

OpenOrder and CorporateAction is not included.) When batch ONE processing is complete, batch TWO processing begins. The same occurs for all other batches until all batch processing is complete.

You can run intra-day processing and add or omit detection runs at the end of (non end-of-day) ingestion batch runs. These cycles of detection should only run BEX and some TC scenarios. They detect only against that day's data and/or data for open batches, dependent on each scenario against which each batch is running. The last intra-day batch should be configured as the end-of-day batch.

You must run a final end-of-day batch that detects on all data loaded into the database for that day, not only looking at the batch that was last loaded. The system can display these events on the next day.

If you want to use either types of intra-day ingestion, you must set up intra-day batches and one end-of-day batch. If you do not, the FDT processes more market data than necessary and runs for a long period.

The following table provides an example of setting up the `KDD_PRCSNG_BATCH` table.

Table 66: Processing Batch Table Set-up

ONE	Intra-Day batch 1	1	NNN
TWO	Intra-Day batch 2	2	NNN
NNN	Intra-Day batch N+ end of day	3	NNN

E.4 Guidelines for Duplicate Record Handling

Records are considered duplicates if the primary business key for multiple records are the same. The Ingestion Manager manages these records by performing either an insert or update of the database with the contents of the first duplicate record. The system inserts the record if a record is not currently in the database with the same business key. The record updates the existing database record if one exists with the same business key. The Ingestion Manager handles additional input records with the same business key by performing database updates. Therefore, the final version of the record reflects the values that the last duplicate record contains.

E.5 Data Rejection During Ingestion

The Ingestion Manager can reject records at the Pre-processing, Transformation, or Loading stages. The following sections provide an overview of the most frequent types of conditions that cause transactions to be rejected:

- **Rejection During Pre-processing Stage:** Describes how rejections occur during the Pre-processing stage and offers guidance on ways to resolve rejections (refer to section *Rejection During the Pre-processing Stage* for more information).
- **Rejection During Transformation Stage:** Describes how rejections occur during the Transformation stage and offers guidance on ways to resolve rejections (refer to section *Rejection During the Transformation Stage* for more information).
- **Rejection During Loading Stage:** Describes how rejections occur during the Loading stage and offers guidance on ways to resolve rejections (refer to section *Rejection During the Loading Stage* for more information).

E.5.1 Rejection During the Pre-processing Stage

The first stage of ingestion is Pre-processing. At this stage, Data Management examines Oracle client reference and trading data for data quality and format to ensure the records conform to the

requirements in the DIS. Common reasons for rejection of data during Pre-processing include problems with data type, missing data, referential integrity, and domain values.

During normal operation, the number of rejections at the Pre-processor stage should be minimal. If the volume of rejections at this stage is high, a decision threshold can halt processing and allow manual inspection of the data. The rejections are likely the result of a problem in the data extraction process. It is possible to correct the rejections and then reingest the data.

E.5.1.1 Data Type

Every field in a record that processing submits to the Ingestion Manager must meet the data type and length requirements that the DIS specifies. Otherwise, the process rejects the entire record. For example, fields with a *Date Type* must appear in the format YYYYMMDD. Thus, the date April 30, 2005 has a format of 20050430 and, therefore, is unacceptable. In addition, a field cannot contain more characters or digits than specified. Thus, if an Order Identifier in an Order record contains more than the maximum allowed length of 40 characters, rejection of the entire record occurs.

E.5.1.2 Missing Data

The DIS defines fields that are mandatory, conditional, and optional. If a record contains a field marked mandatory, and that field has a null value, processing rejects the record. For example, all Trade Execution records must contain a Trade Execution Event Number. If a field is marked conditional, it must be provided in some cases. Thus, an Order record for a limit order must contain a Limit Price, but an Order record for a market order need not contain a Limit Price.

E.5.1.3 Referential Integrity

In some cases, you can configure Ingestion Manager to reject records that refer to a missing reference data record. For example, Ingestion Manager can reject an order that refers to a deal that does not appear in the Deal file. The default behavior is not to reject records for these reasons.

E.5.1.4 Domain Values

Some fields are restricted to contain only one of the domain values that the DIS defines. The Ingestion Manager rejects records that contain some other value. For example, Ingestion Manager rejects any Order record that contains an Account Type other than CR, CI, FP, FB, ER, IA, EE or any Special Handling Code other than that in the DIS.

E.5.2 Rejection During the Transformation Stage

The second stage of ingestion is Transformation. At this stage, the Ingestion Manager derives the order and trade life cycles, and other attributes, that are necessary for trade-related surveillance. The Ingestion Manager rejects order records during Transformation for the following reasons:

- New and Cancel or Replace order events if the order identifier and placement date combination already exists; order identifiers must be unique during a given day.
- New order events for child orders if the referenced parent order is itself a child order; only one level of a parent-child relationship is allowed.

The Ingestion Manager rejects trade execution records for New and Cancel or Replace trade execution events if the trade execution identifier and trade execution date combination already exists. Trade execution identifiers must be unique during a given day.

Other problems can occur that do not cause rejection of records but cause handling of the records to be different:

- Lost Events
- Out of Sequence Events

The following sections describe these issues.

E.5.2.1 Lost Events

If the system receives an order event other than a New or Cancel or Replace in a set of files before receiving the corresponding New or Cancel or Replace, it writes the order event to a lost file. The system examines events in the lost file during processing of subsequent sets of files to determine whether the system received the corresponding New or Cancel or Replace event. If so, processing of this event is normal. If an event resides in the lost file when execution of open order processing occurs (that is, execution of `runDP.sh OPEN_ORDER`), processing rejects the event. The same applies to trade execution events. In addition, if a New trade execution event references an order but the system did not receive the order, the New event also resides in the lost file subject to the same rules.

If rejection of a New or Cancel or Replace order or trade execution occurs during the Pre-processor stage, all subsequent events are considered lost events. Submission of missing New or Cancel or Replace event can occur in a subsequent set of files, and processing of the lost events continue normally.

E.5.2.2 Out-of-Sequence Events

An out-of-sequence event is an order or trade execution event (other than New or Cancel or Replace) that the system processes in a set of files after processing the set of files that contains the corresponding New or Cancel or Replace event. Such an event that has a time stamp prior to the time stamp of the last event against that order or trade is considered an out-of-sequence event.

For example, File Set 1 contains the following events:

- NW order event, timestamp 09:30:00.
- MF order event, timestamp 09:45:00.

File Set 2 contains NW trade execution event (references the above order), timestamp 09:40:00.

This trade execution event is considered out of sequence. It is important to note that this also includes market data. If, in a given batch, market data up to 10:00:00 is used to derive attributes for a given order, any event in a subsequent file against that order with a time stamp prior to 10:00:00 is considered out of sequence.

An out-of-sequence event has no effect on the order or trade that it references. Processing sets the out-of-sequence flag for the event to Y(Yes) and the system writes the event to the database. An Out of Sequence event has no effect on the order or trade that it refers if processing sets the Out-of-sequence flag set for the event to Y

For end-of-day processing, this may not be an issue. For Intra-day processing, subsequent files should contain data in an ever-increasing time sequence. That is, the first set of files should contain data from 09:00:00 to 11:00:00, the second set of files should contain data from 11:00:00 to 12:00:00, and so on. This only affects events in a single order or trade's life cycle. For example, Batch 1 contains the following events:

- NW order event for order X, timestamp 09:30:00.
- MF order event for order X, timestamp 09:45:00.

Batch 2 contains the event NW order event for order Y, timestamp 09:40:00.

This order event is not considered out of sequence; processing continues normally.

E.5.3 Rejection During the Loading Stage

The last stage of ingestion is Loading. At this stage, the Ingestion Manager loads orders, executions, and trades into the database. The Ingestion Manager rejects records during Loading if configuration of the database is incorrect , such as setup of partitions, are incorrect for the data being ingested).

E.6 Alternatives to Standard Data Management Practices

The following sections provide alternative data management practices.

E.6.1 Data Management Archiving

During ingestion processing, the system moves processed files into an archive directory. Firms can use these files to recover from processing malfunctions, and they can copy these files to off-line media for backup purposes.

The Pre-processor moves files in the `/inbox` directory. All other components move their input files to date-labeled subdirectories within the `/backup` directory.

Periodically, an Oracle client can run the `runIMC.sh` script to perform the Ingestion Manager cleanup activities. This script deletes old files from the archive area based on a configurable retention date. Periodic running of the cleanup script ensures that archive space is available to archive more recent data.

E.6.2 Fuzzy Name Matcher Utility

During Datamap processing, the Fuzzy Name Matcher utility is used to match names of individuals and corporations (candidates) against a list of names (targets). The utility calculates a score that indicates how strongly the candidate name matches the target name. All matches are case-insensitive.

E.6.3 Using the Fuzzy Name Matcher Utility

The utility typically runs as part of automated processing that a job scheduling tool such as Maestro or Unicenter AutoSys manages. You can also execute the utility through a UNIX shell script, which the next section describes.

The following topics describe this process:

- Configuring the Fuzzy Name Matcher Utility.
- Executing the Fuzzy Name Matcher Utility.

E.6.3.1 Configuring the Fuzzy Name Matcher Utility

The Fuzzy Name Matcher utility can be used in the following ways:

- Through Ingestion Manager as a standalone Fuzzy Name Matcher. For more information, refer to Executing the Fuzzy Name Matcher Utility. To configure Fuzzy Name Matcher, modify `<ingestion_manager>/fuzzy_match/mantas_cfg/install.cfg`.
- Through Datamaps (`NameMatchStaging.xml`, `RegOToBorrower.xml`) file in folder (`<OFSAAI Installed Directory>/bdf/config/datamaps`). For more information, refer to Managing Data. To configure Fuzzy Name Matcher, modify `<ingestion_manager>/fuzzy_match/mantas_cfg/install.cfg`.

The following figure provides a sample configuration appearing in `<OFSAAI Installed Directory>/bdf/fuzzy_match/mantas_cfg/install.cfg`.

```
################################################################
#
#      Fuzzy Name Matcher System Properties file (install.cfg)
#
################################################################


#---------------------------------------------------------------
#                Log configuration items
#---------------------------------------------------------------
# Specify which priorities are enabled in a hierarchical fashion, i.e.,
if
# DIAGNOSTIC priority is enabled, NOTICE, WARN, and FATAL are  also
enabled,
# but TRACE is not.
# Uncomment the desired log level to turn on appropriate level(s).
# Note, DIAGNOSTIC logging is used to log database statements and will
slow
# down performance.  Only turn on if you need to see the SQL statements
being
# executed.
# TRACE logging is used for debugging during development.  Also only
turn on
# TRACE if needed.
#log.fatal=true
#log.warning=true
log.notice=true
#log.diagnostic=true
#log.trace=true


# Specify where a message should get logged -- the choices are
mantaslog,
# syslog, console, or a filename (with its absolute path).
# Note that if this property is not specified, logging will go to the
console.
log.default.location=mantaslog
# Specify the location (directory path) of the mantaslog, if the
mantaslog
```

```
# was chosen as the log output location anywhere above.
# Logging will go to the console if mantaslog was selected and this
property is
# not given a value.
log.mantaslog.location=mp


#-----------------------------------------------------------------
#                Fuzzy Name Matcher configuration items
#-----------------------------------------------------------------
fuzzy_name.match_multi=true
fuzzy_name.file.delimiter=~
fuzzy_name.default.prefix=P
fuzzy_name.max.threads=1
fuzzy_name.max.names.per.thread=1000
fuzzy_name.max.names.per.process=250000
fuzzy_name.min.intersection.first.letter.count=2
fuzzy_name.temp_file.directory=/scratch/ofsaaapp/BD805/BD805/bdf/data/
temp


fuzzy_name.B.stopword_file=/scratch/ofsaaapp/BD805/BD805/bdf/
fuzzy_match/share/stopwords_b.dat
fuzzy_name.B.match_threshold=80
fuzzy_name.B.initial_match_score=75.0
fuzzy_name.B.initial_match_p1=2
fuzzy_name.B.initial_match_p2=1
fuzzy_name.B.extra_token_match_score=100.0
fuzzy_name.B.extra_token_min_match=2
fuzzy_name.B.extra_token_pct_decrease=50
fuzzy_name.B.first_first_match_score=1


fuzzy_name.P.stopword_file=/scratch/ofsaaapp/BD805/BD805/bdf/
fuzzy_match/share/stopwords_p.dat
fuzzy_name.P.match_threshold=70
fuzzy_name.P.initial_match_score=75.0
```

```
fuzzy_name.P.initial_match_p1=2
fuzzy_name.P.initial_match_p2=1
fuzzy_name.P.extra_token_match_score=50.0
fuzzy_name.P.extra_token_min_match=2
fuzzy_name.P.extra_token_pct_decrease=50
fuzzy_name.P.first_first_match_score=0
```

Figure 57: Sample BDF.xml Configuration Parameters

The following table describes the utility's configuration parameters as they appear in the `BDF.xml` file. Note that all scores have percentage values.

Table 67: Fuzzy Name Matcher Utility Configuration Parameters

Parameter	Description
fuzzy_name.stopword_file	Identifies the file that stores the stop word list. The stop word file is either corporate or personal. The `<prefix>` token identifies corporate as *B* and personal as *P*. Certain words such as *Corp, Inc*, *Mr*, *Mrs*, or *the*, do not add value when comparing names.
fuzzy_name.match_threshold	Indicates the score above which two names are considered to match each other. The utility uses this parameter only when the `match_multi` property has a value of *`true`*. The allowable range is from 0 to 100.
fuzzy_name.initial_match_score	Specifies the score given for matching to an initial. The allowable range is 0 to 100; the recommended default is 75.
fuzzy_name.initial_match_p1	Specifies the number of token picks that must be made before awarding `initial_match_score`. The value is an integer >= 0. The default value is 2.
fuzzy_name.initial_match_p2	Specifies the number of token picks that must be made before awarding `initial_match_score` if only initials remain in one name. The value is an integer >= 0. The default value is 1.
fuzzy_name.extra_token_match_score	Indicates the score given to extra tokens. The allowable range is 0 to 100; the recommended default is 50.
fuzzy_name.extra_token_min_match	Specifies the minimum number of matches that occur before awarding `extra_token_match_score`. The range is any integer >= 0. The recommended setting for corporations is 1; for personal names is 2.

Table 67: Fuzzy Name Matcher Utility Configuration Parameters (Continued)

Parameter	Description
fuzzy_name.extra_token_pct_decrease	Determines the value of the `extra_token_match_score` parameter in regard to extra tokens. If multiple extra tokens are present, reduction of `extra_token_match_score` occurs for each additional extra token. The utility multiplies it by this number. For example, if `extra_token_match_score` = 50, and `extra_pct_decrease` is 50 (percent), the first extra token gets 50 percent, the second extra token gets 25 percent, the third token gets 12.5 percent, the fourth 6.25 percent, the fifth 3.125 percent, etc. The allowable range is 0 to 100. The recommended percentage for corporations is 100 (percent); for personal names, 50 (percent).
fuzzy_name.first_first_match_score	Allows the final score to be more heavily influenced by how well the first token of name #1 matches the first token of name #2. The allowable value is any real number >= 0. The recommended value for corporate names is 1.0; for personal names, 0.0.
fuzzy_name.match_multi	Determines how to handle multiple matches above the `match_threshold` value. If set to "*true,*" the utility returns multiple matches. If set to "*`false`*," it returns only the match with the highest score.
fuzzy_name.file.delimiter	Specifies the delimiter character used to separate each columns in the result file and target name list file.
fuzzy_name.min.intersection.first.letter.count	Specifies the number of words per name whose first letters match. For example, if parameter value = 1 only the first letter of the first or last name would have to match to qualify. If the value = 2, the first letter of both the first and last name would have to match to qualify. **Warning:** By default, the value is set to 2. Oracle recommends using the default value. You must not change the value to 1 or your system performance may slow down.
fuzzy_name.default.prefix	For entries that are not specified as business or personal name, default to this configuration set.

Table 67: Fuzzy Name Matcher Utility Configuration Parameters (Continued)

Parameter	Description
fuzzy_name.max.names.per.process	This property variable determines whether or not the fuzzy matcher algorithm will be run as a single process or as multiple sequential processes. If the total number of names between both the candidate name list and the target name list is less than the value of this property, then a single process will be run. If the number of names exceeds this property's value, then multiple processes will be run, based on how far the value is exceeded. For example, if the candidate name list contains 50 names, the target name list contains 50 names, and the fuzzy_name.max.names.per.process property is set to 200, then one process will be run (because the total number of names, 100, does not exceed 200). If the candidate list contains 400 names, the target name list contains 200 names, and the fuzzy_name.max.names.per.process property is set to 300, then four processes will be run (each with 100 candidate names and 200 target names so that the max number of names per process never exceeds 300). The ability to break apart one large fuzzy matcher process into multiple processes through this property can help to overcome per-process memory limitations imposed by certain Behavior Detection architectures.
fuzzy_name.max.threads	This parameter controls the number of threads to use when Fuzzy Name Matcher is being run. Oracle recommends that this value is not set to a number higher than the number of processing cores on the system.
fuzzy_name.max.names.per.thread	This parameter keeps the processing threads balanced so that they perform work throughout the course of the fuzzy matcher job. That is, instead of splitting the number of names to process evenly across the threads, the value of this parameter can be set to a smaller batch-size of names so that threads that finish ahead of others can keep working.

E.6.3.2 Executing the Fuzzy Name Matcher Utility

To execute the Fuzzy Name Matcher Utility manually, type the following at the UNIX command line:

```
fuzzy_match.sh –t <target_name_list> -c <candidate_name_list> -r <result_file>
```

E.6.4 Refresh Temporary Tables Commands

Prior to running post-processing, you must execute database scripts after ingestion and prior to running AML scenarios. These scripts refresh the required temporary tables for selected AML scenario detection.

E.6.5 Use of Control Data

After installing the OFSBD software, you can use control data provided to test end-to-end processing of data (that is, running data management, executing scenarios, and viewing generated events in the behavior detection UI). Thus, you can verify that installation of the software is correct and works as designed.

To prepare the system for testing, follow these steps:

9. Complete the prerequisites for using control data (refer to section Prerequisites for Using Control Data for more information).
10. Prepare for ingestion of the control data (refer to section Control Data Management for more information).
11. Install the control data (refer to section Loading Control Data Thresholds for more information).
12. Run Behavior Detection on control data to generate events (refer to section Running Behavior Detection on Control Data for more information).

E.6.6 Prerequisites for Using Control Data

Before you use control data to test your installation, the following prerequisites must be fulfilled:

- The maximum lookback that control data considers is of 13 months, which is for change in behavior scenarios. Hence, while creating control data ensure that it is spread over 25 different dates in 13 months.
- The current day according to control data is 20151210.
- Unless specified, set the current date as 20151210, to generate events on control data, before running Behavior Detection Platform.

> **NOTE** For more information about control data on your site, contact your Oracle Administrator.

E.6.7 Control Data Management

Control data uses a specific set of dates to ensure that all the lock-stock scenarios are tested using this data. The maximum lookback that control data considers is of 13 months, which is for change in behavior scenarios. The control data is spread over 25 different dates in 13 months. The dates (YYYYMMDD format) being used by control data are:

Table 68: Dates used by Control Data

20141231	20151123
20150130	20151124
20150227	20151125
20150331	20151126
20150430	20151127
20150529	20151130
20150630	20151203
20150731	20151204
20150831	20151208
20150930	20151209
20151030	20151210

Table 68: Dates used by Control Data

20151201	20151202
20151121	

On all these dates, ingest the data and run the complete batch for the respective date. Except for TBAML and Post-Processing tasks, perform all other activities for the Control Data Management dates. Activities required during any Behavior Detection Framework business day are - `START BATCH > DRM > DATA INGESTION > BEHAVIOR DETECTION > POST PROCESSING > END BATCH.`

Prior to running TBAML on the control data, you must complete the following procedures.

1. Copy all control data from the golden data directory in the database subsystem (`/database/golden_data directory`) to the Ingestion Manager `/inbox` directory bdf /inbox.
2. Run ingestion for all the control Data Management dates. Refer to section Ingestion Timeline - Intra-Day Ingestion Processing, for more information about the ingestion process.

> **NOTE** You must adjust the partitions of the database tables as per the new dates, if you intend to process Control Data after the database upgrade to TBAML.

E.6.8 Loading Control Data Thresholds

To generate breaks on the control data, specific threshold sets and jobs are created. These threshold sets must be installed to the TBAML system for use of control data and generation of test events.

1. Navigate to the directory `<OFSAAI Installed Directory>/database/golden_data/threshold_sets`.
   This directory consists of test threshold sets of all the scenarios that are available with the OFSAAI system.
2. Execute shell script `load_tshld_set.sh`. This shell script installs the control data threshold sets for all the scenarios that are installed at your site. It also creates new jobs and template group IDs corresponding to all the scenarios installed. These template group IDs are same as the scenario IDs of installed scenarios.
3. Once the control data thresholds are installed, the system is ready for a test run, that is, generating test events.

E.6.9 Running Behavior Detection on Control Data

In order to generate events on the ingested control data, execute the new scenario jobs. These jobs consists of same template group ID as the scenario ID. (Refer to Chapter 4, Behavior Detection Jobs to get information regarding about running Behavior Detection Jobs.)

E.6.9.1 Important Notes

1. Run loaded scenarios with the system date as 20151210 with the following exceptions:
   d. For Portfolio Pumping scenario, the system date must be 20151204
   a. For Active Trading scenario, the system date must be 20151130
2. Check for system errors in the appropriate logs (refer to Appendix A, Logging, for more information).
3. Run post-processing procedures.

4. Close the batch to enable display of events in the Behavior Detection UI.
5. Log in to the Behavior Detection UI with the correct user credentials.
6. Verify that you can view events in the UI.

The display of events signifies that installation of the system is correct and works as designed.

> **NOTE** The events that you can view depend on your user privileges.

F TBAML Datamap Details

This appendix lists the TBAML datamaps used in OFSAAI and a brief explanation of the each datamap. This section contains the following sections:

- Trade Finance Datamaps
- Watchlist Datamaps

> **TIP** Oracle recommends all datamaps are run in the order described in the following tables.

F.1 Trade Finance Datamaps

This section provides the required datamaps for deriving and aggregating data. The datamaps appear in the order that processing must execute them during data loading. Where predecessors exist, processing of datamaps cannot begin the predecessor datamap has completed. Predecessors may be internal to the datamap type, or external to the datamap type, as shown in the following examples.

Example For Internal Predecessor

Processing can run the `Frontofficetransactionparty_instnseqid` datamap immediately after completion of `Financialinstitution_fotpspopulation` and `Accounttoclientbank_fotpsinstitutioninsert`.

Example For External Predecessor

Processing cannot run the `Accountprofile_trade` datamap until and unless the `Frontofficetransactionpartyriskstage_entityactivityriskinsert` datamap is run..

> **NOTE** If there is a performance issue with the running sequence of datamaps, they can be re-arranged. However. the predecessor for the datamap must be completed before running the datamap. For example:
>
> Run the following datamaps in order:
>
> 1. `FrontOfficeTransactionParty_InstnSeqID`
> 2. `FrontOfficeTransactionParty_HoldingInstnSeqID`
>
> If there is are performance issues with `FrontOfficeTransactionParty_HoldingInstnSeqID`, the datamap position can be rearranged in the batch script. Since there is the possibility that `FrontOfficeTransactionParty_InstnSeqID` is still running, the current datamap is waiting for the resources to be released.

The following sections describe the Trade Finance Datamaps:

- Trade Finance - Pre-Watch List Datamaps
- Trade Finance- Post-Watch List Datamaps

The following table describes the columns in the Datamap tables that each section provides.

Table 69: Datamap Table Descriptions

Column	Description
Datamap Number	Unique, five-digit number that represents a particular datamap.
Datamap Name	Unique name of each datamap.
Predecessor	Indicator that processing of datamaps cannot begin until completion of predecessor datamaps.

F.1.1 Trade Finance - Pre-Watch List Datamaps

Pre-Watch List Datamaps are used to facilitate the application to populate various business areas such as, Financial Institutions, Account To Client Bank, Settlement Instructions, Front Office and Back Office Transaction. These datamaps populate the relevant data which would again be used in watch list datamaps in calculating risks.

Optional Datamaps are used to perform processing to support other datamaps in multiple functional areas. These datamaps may or may not be completely relevant to a particular solution set. Execute the datamap if a scenario in your implementation requires this information

The following tables are not supported through CSA ingestion methods.

- CustomerImportLicense
- CustomerImportLicensetoGoods
- DocumentaryCollectionInvoice
- DocumentaryCollectionMulti-tenorDetail
- DocumentaryCollectionShipmentDetail
- ExternalInsurancePolicy
- TradeFinanceBrokerageDistributionStage
- TradeFinanceBrokerage
- TradeFinanceDraft.

Table 70: Trade Finance - Pre-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60200	TradeFinanceContractEvent.xml	NA
60210	TradeFinanceContractEventAcknowledgementStage.xml	NA
60220	TradeFinanceContractAmendmentStatusStage.xml	NA
60230	TradeFinanceContractEvent_AcknowledgeUpd.xml	60200 60210
60240	TradeFinanceContractEvent_AmendmentUpd.xml	60200 60220
60250	TradeFinanceContract.xml	60200
60260	TradeFinancetoAccount.xml	NA

Table 70: Trade Finance - Pre-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60270	TradeFinanceDocument.xml	NA
60280	TradeFinanceDraft.xml	NA
60290	TradeFinanceGoodorService.xml	NA
60300	TradeFinanceParty.xml	NA
60310	TradeFinanceParty_TradeFinancePartyStage.xml	60300
60320	TradeFinanceContract_PartyUpd.xml	60300 60240 60230 60220 60210 60200
60330	TradeFinanceContract_DocUpd.xml	60270 60240 60230 60220 60210 60200
60340	TradeFinanceContract_GoodsUpd.xml	60290 60240 60230 60220 60210 60200
60350	DerivedAddress_TradeFinancePartyInsert.xml	60300 60310
60360	DerivedAddress_TradeFinancePartyUpd.xml	60350 60300 60310
60390	FinancialInstitution_TradeFinanceParty.xml	60300 60310
60400	DerivedEntity_TradeFinancePartyInsert.xml	60300 60310
60410	DerivedEntity_TradeFinancePartyUpd.xml	60300 60310 60400

Table 70: Trade Finance - Pre-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60420	TradeFinancePartyTF_DerivedEntityUpd.xml	60410 60400 60310 60300
60430	TradeFinancePartyDC_DerivedEntityUpd.xml	60410 60400 60310 60300
60460	CustomerImportLicense.xml	NA
60470	CustomerImportLicensetoGoods.xml	NA
60480	TradeFinanceBrokerage.xml	NA
60490	ExternalInsurancePolicy.xml	NA
60500	ExternalOrganizationStage	NA
60510	ExternalOrganization	60500
60520	DerivedAddress_ExternalOrganizationStageInsert.xml	60510 60500
60530	DerivedAddress_ExternalOrganizationStageUpd.xml	60520 60510 60500
60540	ExternalOrganization_DerivedAddress.xml	60530 60520 60510 60500
60550	DerivedEntity_ExtrlOrgInsert.xml	60540 60530 60520 60510 60500
60560	TradeFinanceBrokerageDistributionStage.xml	NA
60570	TradeFinanceBrokerageDistribution.xml	60550
60580	FinancialInstitution_BrokerageDistribution.xml	60560 60550 60390 60300 60310

Table 70: Trade Finance - Pre-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60590	BrokerageDistribution_FinancialInstnUpd.xml	60570 60560 60550 60390 60300 60310
60600	DerivedAddress_TradeFinanceBrokerageDistributionStageInsert.xml	60580 60570 60560 60550 60390 60300 60310
60610	DerivedAddress_TradeFinanceBrokerageDistributionStageUpd.xml	60590 60580 60570 60560 60550 60390 60300 60310
60620	BrokerageDistribution_DerivedAddress.xml	60600 60590 60580 60570 60560 60550 60390 60300 60310
60630	DocumentaryCollectionContractEvent.xml	NA
60640	DocCollectionContractAcknowlegementStage.xml	NA
60650	DocumentaryCollectionContractAcceptanceStage.xml	NA
60660	DocumentaryCollectionContractEvent_AcknowledgeUpd.xml	60620 60630
60670	DocumentaryCollectionContractEvent_AcceptanceUpd.xml	60620 60640
60680	DocumentaryCollectionDiscrepancyDetail.xml	NA

Table 70: Trade Finance - Pre-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60690	DocumentaryCollectionDiscrepancyDetail_DiscrpDtUpd.xml	60620 60670
60700	DocumentaryCollectionInvoice.xml	NA
60710	DocumentaryCollectionMulti-tenorDetail.xml	NA
60720	DocumentaryCollectionShipmentDetail.xml	NA
60730	DocumentaryCollectionContract.xml	60630 60640 60650 60660 60670 60680 60690
60740	CountryTradeList	NA
60750	GoodsorService	NA
60760	TradeFinanceGoodorService_Upd	60290 60750

F.1.2 Trade Finance- Post-Watch List Datamaps

Post-Watch List Datamaps are used to populate or ingest data into various transaction tables using Front Office and Back Office Transaction files, these are executed only after the Watch List Datamaps are run. These datamaps are used to populate data into Cash, Wire, Monetary Instruments tables, and to update Trusted Pair and Jurisdiction information into various other entities

Table 71: Trade Finance - Post-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60770	DocumentaryCollectionContract_LiquidationUpd.xml	60720
60780	TradeFinancePartyTF_DerivedAddressUpd.xml	60360 60350 60310 60300
60790	TradeFinancePartyDC_DerivedAddressUpd.xml	60370 60360 60350 60310 60300
60800	TradeFinancePartyTF_EntityActivityRiskUpd.xml	NA
60810	TradeFinancePartyDC_EntityActivityRiskUpd.xml	NA

Table 71: Trade Finance - Post-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
60820	TradeFinanceContractEvent_ActivityRskUpd	
60830	DocumentaryCollectionContractEvent_ActivityRskUpd	
60840	ExternalOrganization_ExternalEntitySeqUpd	60200 60210
60850	ExternalOrganization_EntityRiskInsert	60200 60220

F.2 Watch List Datamaps

The following sections describe the Watch List Datamaps

- Watchlist Datamaps
- Post-Watch List Datamaps

F.2.1 Watchlist Datamaps

Watch List Datamaps facilitate the application of customer-supplied measures of risk to corresponding entities, transactions, and instructions.

These datamaps assist other datamaps which are used to calculate Effective Risk and Activity Risk for various entities, such as Account, Customer, Transaction Tables, and so on.

Table 72: Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
10245	WLMProcessingLock	NA
10250	WatchListEntry_WatchListEntryCurrDayInsert	10020 10030 10040 10050 10060 10070 10245
10260	WatchListAudit_StatusUpd	10020 10030 10040 10050 10060 10070

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10270	WatchList_WatchListSourceAuditInsert	10020 10030 10040 10050 10060 10070
10280	WatchList_WatchListSourceAuditUpd	10020 10030 10040 10050 10060 10070
10290	WatchList_WatchListSourceUpd	10020 10030 10040 10050 10060 10070
10300	WatchListEntry_WatchListAuditUpd	10020 10030 10040 10050 10060 10070 10260
10310	WatchListEntryAudit_WatchListEntryUpdate	10020 10030 10040 10050 10060 10070 10300
10320	Customer_KYCRiskUpd	NA
60090	CorrespondentBankToPeerGroup	NA
10360	DerivedAddress_FrontOfficeTransactioPartyStageInsert	NA
10370	DerivedAddress_FrontOfficeTransactioPartyStageUpd	NA
10380	FrontOfficeTransactionParty_DerivedAddress	10360 10370

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10390	DerivedEntity_FrontOfficeTransactionPartyInsert	10080 10090
10400	DerivedEntity_FrontOfficeTransactionPartyUpd	10080 10090
10430	CorrespondentBank_FrontOfficeTransactionPartyStageInsert	10080 10090
10440	CorrespondentBank_FrontOfficeTransactionPartyStageUpd	10080 10090
10450	WatchListStagingTable_WatchList	10250 10260 10270 10280 10290 10300 10310
10460	WatchListStagingTable_WatchListInstnIDUpd	10250 10260 10270 10280 10290 10300 10310
10470	PreviousWatchList_WatchList	10250 10260 10270 10280 10290 10300 10310
10480	DerivedAddress_WatchListNewCountries	10250 10260 10270 10280 10290 10300 10310
10485	WLMProcessingUnlock	10480

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10490	LinkStaging_FrontOfficeTransactionParty	10360 10370 10380 10390 10400 10485
10510	NameMatchStaging	10450 10460 10470 10480 10390 10400
10520	WatchListStagingTable_NameMatchStageInsert	10510
10530	DerivedEntityLink_LinkStage	10490 10500
10540	DerivedEntitytoDerivedAddress_LinkStage	10490 10500
10550	DerivedEntitytoInternalAccount_LinkStage	10490 10500
10560	DerivedAddresstoInternalAccount_LinkStage	10490 10500
10570	WatchListStagingTable2_WatchListStage2AcctExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10580	WatchListStagingTable2_WatchListStage2CBExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10590	WatchListStagingTable2_WatchListStage2CustExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10600	WatchListStagingTable2_WatchListStage2DAExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10610	WatchListStagingTable2_WatchListStage2EEExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10620	WatchListStagingTable2_WatchListStage	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10630	WatchListStagingTable2_AcctListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10640	WatchListStagingTable2_CBListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10650	WatchListStagingTable2_CustListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10660	WatchListStagingTable2_EEListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10670	WatchListStagingTable2_EEListMembershipStatusUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10680	WatchListStagingTable2_DAListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10690	WatchListStagingTable2_DAListMembershipStatusUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10700	WatchListStagingTable2_WatchListStage2SeqIdUpd	10570 10580 10590 10600 10610 10620 10630 10640 10650 10660 10670 10680 10690
10710	WatchListStagingTable2_WatchListStage2IntrlIdUpd	10570 10580 10590 10600 10610 10620 10630 10640 10650 10660 10670 10680 10690
10720	Customer_WatchListStage2ListRisk	10320 10700 10710
10730	CorrespondentBank_WatchListStage2EffectiveRisk	10320 10700 10710
10740	Customer_WatchListStage2EffectiveRisk	10320 10700 10710
10750	DerivedAddress_WatchListStage2EffectiveRisk	10320 10700 10710

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10760 10700 10710	DerivedEntity_WatchListStage2EffectiveRisk	10320 10700 10710
10770	WatchListStagingTable2_WatchListStage2SeqId	10320 10700 10710
10780	AccountListMembership_WatchListStage2Insert	10700 10710
10790	AccountListMembership_WatchListStage2Upd	10700 10710
10800	CorrespondentBankListMembership_WatchListStage2Insert	10700 10710
10810	CorrespondentBankListMembership_WatchListStage2Upd	10700 10710
10820	CustomerListMembership_WatchListStage2Insert	10700 10710
10830	CustomerListMembership_WatchListStage2Upd	10700 10710
10840	DerivedAddressListMembership_WatchListStage2Insert	10700 10710
10850	DerivedAddressListMembership_WatchListStage2Upd	10700 10710
10860	DerivedEntityListMembership_WatchListStage2Insert	10700 10710
10870	DerivedEntityListMembership_WatchListStage2Upd	10700 10710
10875	Account_EffectiveRiskFactorTxtUpd	10700 10701

Table 72: Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
10880	Account_OverallEffectiveRiskUpd	10720 10730 10740 10750 10760 10770 10780 10790 10800 10810 10820 10830 10840 10850 10860 10870
	Account_AccountCustRiskUpd	
10890	Account_EffRiskUpdAfterWLRiskRemoval	10720 10730 10740 10750 10760 10770 10880
10900	Account_WatchListStage2EffectiveRisk	10720 10730 10740 10750 10760 10770 10880
10910	WatchListStagingTable2_WatchListStage2IntrlId	10320 10700 10710
10920	BackOfficeTransaction_EffectiveAcctivityRiskUpd	10890 10900
10940	FrontOfficeTransactionPartyRiskStage_EntityActivityRiskInsert	10890 10900

F.2.2 Post-Watch List Datamaps

Post-Watch List Datamaps are used to populate or ingest data into various transaction tables using Front Office and Back Office Transaction files, these are executed only after the Watch List Datamaps are run. These datamaps are used to populate data into Cash, Wire, Monetary Instruments tables, and to update Trusted Pair and Jurisdiction information into various other entities.

> **NOTE** Datamaps 10970,10980,10990, 11000,11010,11020 can be run in parallel.

Table 73: Post-Watch List Datamaps

Datamap Number	Datamap Name	Predecessors
20010	CorrespondentBank_JurisdictionUpd	10430 10440
20020	CorrespondentBank_AcctJurisdictionReUpd	10430 10440
20030	FinancialInstitution_InstNameUpd	10430 10440
10970	TransactionPartyCrossReference_BackOfficeTransaction	10360 10370 10380 10940
10980	CashTransaction_FrontOfficeTransaction	10360 10370 10380 10940
10990	MonetaryInstrumentTransaction_FrontOfficeTransaction	10360 10370 10380 10940
11000	TransactionPartyCrossReference_FrontOfficeTransaction	10360 10370 10380 10940
11010	WireTransaction_FrontOfficeTransaction	10360 10370 10380 10940
11020	WireTransactionInstitutionLeg_FrontOfficeTransaction	10360 10370 10380 10940

Table 73: Post-Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
11030	CashTransaction_FrontOfficeTransactionRevAdj	10970 10980 10990 11000 11010 11020
11040	MonetaryInstrumentTransaction_FrontOfficeTransactionRevAdj	10970 10980 10990 11000 11010 11020
11050	WireTransaction_FrontOfficeTransactionRevAdj	10970 10980 10990 11000 11010 11020
11060	TrustedPair_StatusEXPUpd	10970 10980 10990 11000 11010 11020
11070	TrustedPairMember_AcctExtEntEffecRiskUpd	10970 10980 10990 11000 11010 11020
11080	TrustedPair_StatusRRCInsert	10970 10980 10990 11000 11010 11020

Table 73: Post-Watch List Datamaps (Continued)

Datamap Number	Datamap Name	Predecessors
11090	TrustedPair_StatusRRCUpd	10970 10980 10990 11000 11010 11020
11100	ApprovalActionsAudit_TrustedPair	10970 10980 10990 11000 11010 11020 11060 11080 11090
11110	TrustedPairMember_StatusRRCInsert	10970 10980 10990 11000 11010 11020
11120	BackOfficeTransaction_TrustedFlagsUpd	11060 11070 11080 11090 11100 11110
11140	MonetaryInstrumentTransaction_TrustedFlagsUpd	11060 11070 11080 11090 11100 11110
11150	WireTransaction_TrustedFlagsUpd	11060 11070 11080 11090 11100 11110

F.3 Processing TBAML Datamaps

The following table provides a list of datamaps and description for each datamap. These datamaps are listed in order.

Table 74: TBAML Datamaps

Datamap Number	Datamap Name	Description
50010	Customer_TotAcctUpd	This datamap calculates the total number of accounts for an institutional customer.
10015	FrontOfficeTransactionParty_SecondaryNames	This datamap kicks off the Pass Thru process. It generates second originator and beneficiary records for Front Office Transaction. It also sets the pass thru flag based on the a set of expressions.
10050	AccountToClientBank_AIIMSInstitutionInsert	This datamap creates unique identifiers for banks based BIC records on the third party vendors. 1) Retrieve Institution information from ACCT_INSTN_MAP_STAGE in comparison of INSTN_MASTER and loads it into ACCT_ID_INSTN_ID_MAP.
10060	AccountToClientBank_InstitutionInsert	This datamap creates unique identifiers for banks based on the third party vendors. 1) Retrieve Institution information from ACCT_INSTN_MAP_STAGE and load it into ACCT_ID_INSTN_ID_MAP.
10070	AccountToClientBank_InstitutionUpd	This datamap updates unique identifiers for banks based on the third party vendors. 1) Retrieve Institution information from ACCT_INSTN_MAP_STAGE and update it into ACCT_ID_INSTN_ID_MAP.
10080	FinancialInstitution_FOTPSPopulation	This datamap inserts new records in Financial Institution table for the institutions found in front office transaction party table for both party ID type code as IA and BIC, INSTN_SEQ_ID are OFSAAI generated.
10090	AccountToClientBank_FOTPSInstitutionInsert	This datamap marks all institutions with an OFSAAI generated INTSN_SEQ_ID in FOTPS. 1) Prior to this datamap execution the predecessor datamaps finds the new institutions from the transaction data and loads them in the INSTITUTION_MASTER. 2) This data map finds the new institutions from the transaction data for IA and BIC party ID type and loads them in the ACCT_ID_INSTN_ID_MAP table using OFSAAI generated INTSN_SEQ_ID from INSTITUTION_MASTER.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10114	BackOfficeTransaction_UnrelatedPartyCodeUpd	This datamap updates the UNRLTD_PARTY_CD column of Back Office Transaction table with a value of 'J' or 'JS'.
10116	BackOfficeTransaction_CollateralUpd	This datamap updates Collateral Percentage and, Collateral Value for that transaction.
10120	BackOfficeTransaction_OriginalTransactionReversalUpd	This datamap handles reversals for Back Office Transactions. 1) Select the set of information from today's BackOfficeTransaction to update records with columns CXL_PAIR_TRXN_INTRL_ID in BackOfficeTransaction table. 2) Updates the "cancellation pair" column in the original back office transaction table as per the "Internal ID" of the reversing or adjusting record.
10130	BackOfficeTransaction_CancelledTransactionReversalCreditUpd	This datamap updates Cancelled Transaction details for CREDIT record of Back Office Transactions. 1) Finds original-reversal back-office transaction pairs, links them via their respective transaction identifiers. 2) For original transactions: update Canceled Pairing Transaction Identifier by reversal transaction ID; 3) For reversal transactions: update the transaction's Debit Credit Code, Unit Quantity, Transaction Amount, Canceled Pairing Transaction Identifier by original transaction's field values, and Mantas Transaction Adjustment Code by 'REV'.
10140	BackOfficeTransaction_CancelledTransactionReversalDebitUpd	This datamap updates Cancelled Transaction details for DEBIT record of Back Office Transactions. 1) Finds original-reversal back-office transaction pairs, links them via their respective transaction identifiers. 2) For original transactions: update Canceled Pairing Transaction Identifier by reversal transaction ID; 3) For reversal transactions: update the transaction's Debit Credit Code, Unit Quantity, Transaction Amount, Canceled Pairing Transaction Identifier by original transaction's field values, and Mantas Transaction Adjustment Code by 'REV'.
10150	FrontOfficeTransactionParty_InstnSeqID	This datamap marks all the records of FO_TRXN_PARTY_STAGE table with institutions by OFSAAI generated INTSN_SEQ_ID.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10160	FrontOfficeTransactionParty_HoldingInstnSeqID	This datamap marks all the records of FO_TRXN_PARTY_STAGE table with institutions by OFSAAI generated INTSN_SEQ_ID. 1) To update HOLDG_INSTN_SEQ_ID and HOLDG_ADDR_CNTRY_CD based on DATA_DUMP_DT and country code (BASE_COUNTRY).
10210	FrontOfficeTransaction_UnrelatedPartyUpd	This datamap updates the FOT table for records where UNRLTD_PARTY_FL is 'Y' with a value as 'N', by determining the pairs of parties (internal) in the role of Orig & Benef having either common Tax ID/ Common Customer/Common HH.
10245	WLMProcessingLock	This datamap applies lock to restrict UI accessibility for Watch list Management.
10250	WatchListEntry_WatchListEntryCurrDayInsert	This datamap checks for records in watch list from source files for the current day, if there is no records, create the current day watch list records from the previous day.
10260	WatchListAudit_StatusUpd	This datamap take care of watchlist table for the modifications of the WL based on the new user interface WL utility.
10270	WatchList_WatchListSourceAuditInsert	This datamap takes into account the modifications of the watchlist based on the new user interface WL utility. 1) Get all the records that are active from audit table. Order by created time. 2) Take the latest change for each LIST_SRC_CD Watch List and insert records in WATCH_LIST_SOURCE table.
10280	WatchList_WatchListSourceAuditUpd	This datamap takes into account the modifications of the watchlist based on the new user interface WL utility. 1) Get all the records that are active from audit table. Order by created time. 2) Take the latest change for each LIST_SRC_CD Watch List and update records in WATCH_LIST_SOURCE table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10290	WatchList_WatchListSourceUpd	This datamap takes into account the modifications of the watchlist based on the new user interface WL utility. 1) Get all the records that are active from audit table. Order by created time. 2) Take the latest change for each LIST_SRC_CD Watch List and update records in WATCH_LIST_SOURCE table.
10300	WatchListEntry_WatchListAuditUpd	This datamap takes care of watch list entry table for the modifications of the WL based on the new user interface WL utility.
10310	WatchListEntryAudit_WatchListEntry Update	This datamap take care of watchlist entry audit table for the modifications of the WL based on the new user interface WL utility.
10320	Customer_KYCRiskUpd	This datamap calculates risk, If the risk was List driven, then this can ignore that record. If it was BUS/ GEO driven and there is KYC risk. Apply KYC Risk in Customer table.
60090	CorrespondentBankToPeerGroup	This datamap populates the CLIENT_BANK_PEER_GRP table by associating peer group identifiers in the ACCT_PEER_GRP table with institution identifiers in the ACCT_ID_INSTN_ID_MAP table.
10360	DerivedAddress_FrontOfficeTransactioPartyStageInsert	This datamap selects the distinct set of addresses from today's front-office transactions and if non-existent, inserts new address records into Derived Address.
10370	DerivedAddress_FrontOfficeTransactioPartyStageUpd	This datamap selects the distinct set of addresses from today's front-office transactions and if existent, updates new address records into Derived Address.
10380	FrontOfficeTransactionParty_Derived Address	This datamap maintains the addresses in the DerivedAddress table. It derives the addresses from the FrontOfficeTransactionParty table.
40040	DerivedAddress_InsuranceTransactionInsert	This datamap derives the addresses from the INSURANCE table, and inserts the addresses in to the Derived Address table.
40050	DerivedAddress_InsuranceTransactionUpd	This datamap derives the addresses from the INSURANCE table. If the address already exists in Derived Address table, it will update the addresses in to the Derived Address table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
40060	InsuranceTransaction_InstitutionAddrUpd	This datamap updates Mantas Institution Address Identifier in the Insurance Transaction table. 1) A new record is created in Derived Address table prior to this datamap execution. 2) Update the same Derived Address Sequence ID in INSURANCE_TRXN for CP_ADDR_MSTR_SEQ_ID column.
40070	DerivedEntity_InsuranceTransactionInsert	This datamap maintains the External Entity table. It derives the entities from the INSURANCE table on current processing date.
40080	DerivedEntity_InsuranceTransactionUpd	This datamap maintains the External Entity table. It derives the entities from the INSURANCE table on current processing date.
10390	DerivedEntity_FrontOfficeTransactionPartyInsert	This datamap maintains the External Entity table. It derives the entities from the Front Office and Front Office Party transaction table.
10400	DerivedEntity_FrontOfficeTransactionPartyUpd	This datamap maintains the External Entity table. It derives the entities from the Front Office and Front Office Party transaction table.
10410	DerivedEntity_SettlementInstructionInsert	This datamap maintains the External Entity table. It derives the entities from the Instruction table on current processing date.
10420	DerivedEntity_SettlementInstructionUpd	This datamap maintains the External Entity table. It derives the entities from the INSTRUCTION table. 1) Select the distinct set of names, accounts, institutions from today's Instructions and updates matching records in the External Entity table.
10430	CorrespondentBank_FrontOfficeTransactionPartyStageInsert	This datamap populates the client bank table for current day transactions where there is an institution involved.
10440	CorrespondentBank_FrontOfficeTransactionPartyStageUpd	This datamap maintains the Correspondent Bank table. It derives the records from the FOTPS table. If there is an existing correspond bank record available, this datamap updates the LAST_ACTVY_DT for that record.
10450	WatchListStagingTable_WatchList	This datamap determines changes in the Watch List table Each entry is classified as Add, No Change, or Retire based on the comparison of the current-day watch list data to the previous-day watch list data.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10460	WatchListStagingTable_WatchListInstnIDUpd	This datamap only processes watch list entries that are External Accounts, Financial Institutions, and Internal Accounts. 1) It updates the Watch List Stage table with the corresponding Institution Sequence ID of the institution or account.
10470	PreviousWatchList_WatchList	This datamap save off current day's watch list records into PREV_WATCH_LIST.
10480	DerivedAddress_WatchListNewCountries	This datamap inserts new countries from WL in the derived addresses table.
10485	WLMProcessingUnlock	This datamap releases the lock for Watch list Management.
10490	LinkStaging_FrontOfficeTransactionParty	This datamap loads the Link Stage with any entity associations from FOTPS, depending on the combination of Link Type Code defined.
40090	LinkStaging_InsTrxnDerivedEntDerivedAdd	This datamap loads the Link Stage with any entity associations from INSURANCE.
10500	LinkStaging_InstructionDerivedEntDerivedAdd	This datamap loads the Link Stage with any entity associations from instruction. Define the entity association based on existence of entity and address associations in data.
10510	NameMatchStaging	This datamap use fuzzy match to match Candidate Name against the List Name and inserts records in Name Match Stage table.
10520	WatchListStagingTable_NameMatchStageInsert	This datamap is a wrapper for the fuzzy matching mappings and scripts. 1) For each processing day, this datamap joins fuzzy names to their matched watch list records to create additional watch list records for subsequent application to transactional tables.
10530	DerivedEntityLink_LinkStage	This datamap selects the external entity links from today's Link Stage table and insert records in External Entity Link table in associations to various link tables.
10540	DerivedEntitytoDerivedAddress_LinkStage	This datamap writes link-stage associations to various link tables in External Entity Address Table.
10550	DerivedEntitytoInternalAccount_LinkStage	This datamap writes link-stage associations to various link tables in External Entity Account Table.
10560	DerivedAddresstoInternalAccount_LinkStage	This datamap writes link-stage associations to various link tables in Derived Account Address Table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10570	WatchListStagingTable2_WatchListStage2AcctExistence	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with ACCT entity. 2) For IA (ACCT table) watch list entries, the error status is assigned if the entity does not exist in the entity table because these entity records are expected to exist.
10580	WatchListStagingTable2_WatchListStage2CBExistence	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with CLIENT_BANK entity. 2) Evaluates the existence of the CLIENT_BANK entity and assigns a 'Warning"" status to the record if the entity does not exist in the entity table because these entity records are expected to exist.
10590	WatchListStagingTable2_WatchListStage2CustExistence	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with CUST entity. 2) For CU (CUST table) watch list entries, the error status is assigned if the entity does not exist in the entity table because these entity records are expected to exist.
10600	WatchListStagingTable2_WatchListStage2DAExistence	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with DERIVED_ADDRESS entity. 2) Evaluates the existence of the DERIVED_ADDRESS record and assigns status to the record accordingly.
10610	WatchListStagingTable2_WatchListStage2EEExistence	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with EXTERNAL_ENTITY entity. 2) Evaluates the existence of the EXTERNAL_ENTITY record and assigns a 'Warning' status to the record if the entity does not exist in the entity table because these entity records are expected to exist.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10620	WatchListStagingTable2_WatchListStage	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Check for watch list stage CUST_INTRL_ID flag if it is 'Y' means that this name is fuzzy matched. 2) Insert the watch list entry into the second processing table that is Watch list stage 2 table for both the fuzzy matched as well as exact name records.
10630	WatchListStagingTable2_AcctListMembershipUpd	The datamap checks for entry membership in the corresponding entity list membership table.
10640	WatchListStagingTable2_CBListMembershipUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with CB_LIST_MEMBERSHIP entity. 2) Evaluates the existence of the CB_LIST_MEMBERSHIP record and assigns a 'Warning' status to the record if the entity does not exist in the entity table because these entity records are expected to exist.
10650	WatchListStagingTable2_CustListMembershipUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with CUST_LIST_MEMBERSHIP entity. 2) Evaluates the existence of the CUST_LIST_MEMBERSHIP record and assigns a 'Warning' status to the record if the entity does not exist in the entity table because these entity records are expected to exist.
10660	WatchListStagingTable2_EEListMembershipUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with EXTERNAL_NTITY_LIST_MEMBERSHIP entity. 2) Evaluates the existence of the EXTERNAL_NTITY_LIST_MEMBERSHIP record and assigns a 'Warning' status to the record if the entity does not exist in the entity table because these entity records are expected to exist.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10670	WatchListStagingTable2_EEListMembershipStatusUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) It validates the list membership status of External Entities whose Last Activity Date is earlier than the current date. 2) Update the status of the watch list entry based the existence or non-existence of a corresponding list membership record.
10680	WatchListStagingTable2_DAListMembershipUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) Processes all watch list entries that have a possible match with DERIVED_ADDR_LIST_MEMBERSHIP entity. 2) Evaluates the existence of the DERIVED_ADDR_LIST_MEMBERSHIP record and assigns a 'Warning' status to the record if the entity does not exist in the entity table because these entity records are expected to exist.
10690	WatchListStagingTable2_DAListMembershipStatusUpd	This datamap validates each watch list entry and inserts into the processing table WATCH_LIST_STAGE2. 1) It validates the list membership status of DERIVED_ADDRESS whose Last Activity Date is earlier than the current date. 2) Update the status of the watch list entry based the existence or non-existence of a corresponding list membership record.
10700	WatchListStagingTable2_WatchListStage2SeqIdUpd	This datamap updates the list risk of each valid watch list entity based on the entity Sequence ID. The datamap sets various flags and derives the highest List Risk value for each entity on the watch list.
10710	WatchListStagingTable2_WatchListStage2IntrlIdUpd	This datamap updates the list risk of each valid watch list entity based on the entity Internal ID. The datamap sets various flags and derives the highest List Risk value for each entity on the watch list.
10720	Customer_WatchListStage2ListRisk	This datamap calculates the customer's effective risk and set the risk factor if the risk is not found for the current day in watch list stage table. After calculating the risk updates the CUST table. Use nulls for the List Risk and the List Source Code.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10730	CorrespondentBank_WatchListStage2EffectiveRisk	This datamap calculates the Client Bank Effective Risk and applies the Effective Risk and the List Risk to the CLIENT_BANKrecord.
10740	Customer_WatchListStage2EffectiveRisk	This datamap calculates the Effective Risk of Customer and applies the Effective Risk and the List Risk to the CUST record.
10750	DerivedAddress_WatchListStage2EffectiveRisk	This datamap calculates the Effective Risk of all derived address entities and applies the Effective Risk and the List Risk to the DERIVED_ADDRESS record.
10760	DerivedEntity_WatchListStage2EffectiveRisk	This datamap calculates the Effective Risk of all external entities and applies the Effective Risk and the List Risk to the EXTERNAL_ENTITY record.
10770	WatchListStagingTable2_WatchListStage2SeqId	This datamap calculates the Effective Risk of all entities and applies the Effective Risk and the List Risk to the entity record where sequence ID is not null.
10780	AccountListMembership_WatchListStage2Insert	This datamap inserts List Membership records for entities into ACCT_LIST_MEMBERSHIP table that are new to a list.
10790	AccountListMembership_WatchListStage2Upd	This datamap updates the existing retired ACCT_LIST_MEMBERSHIP records by setting List Removal Date to the current processing date.
10800	CorrespondentBankListMembership_WatchListStage2Insert	This datamap inserts List Membership records for entities that are new to a list into CB_LIST_MEMBERSHIP table.
10810	CorrespondentBankListMembership_WatchListStage2Upd	This datamap updates the existing retired CB_LIST_MEMBERSHIP records by setting List Removal Date to the current processing date.
10820	CustomerListMembership_WatchListStage2Insert	This datamap inserts List Membership records for entities that are new to a list into CUST_LIST_MEMBERSHIP table.
10830	CustomerListMembership_WatchListStage2Upd	This datamap updates the existing retired CUST_LIST_MEMBERSHIP records by setting List Removal Date to the current processing date.
10840	DerivedAddressListMembership_WatchListStage2Insert	This datamap maintains the Derived Address List membership table based on the current WL processing results.
10850	DerivedAddressListMembership_WatchListStage2Upd	This datamap maintains the Derived Address List membership tables based on the current WL processing results by setting List Removal Date to the current processing date.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10860	DerivedEntityListMembership_WatchListStage2Insert	This datamap inserts List Membership records for entities that are new to a list into EXTERNAL_NTITY_LIST_MEMBERSHIP table.
10870	DerivedEntityListMembership_WatchListStage2Upd	This datamap maintains the External Entity membership tables based on the current WL processing results by setting List Removal Date to the current processing date.
10875	Account_EffectiveRiskFactorTxtUpd	This datamap updates the Account Effective Risk Factor for the Account.
10876	Account_AccountCustRiskUpd	This datamap updates the account's primary customer risk for the previous account record whose primary customer risk got changed.
10880	Account_OverallEffectiveRiskUpd	This datamap updates the risk on the ACCT based on KYC, Primary customer, as well as other external risks.
10890	Account_EffRiskUpdAfterWLRiskRemoval	This datamap Updates the account Effective Risk to the maximum of the business risk, geographic risk, and customer risk. The account Effective Risk was already set to the higher of the customer-supplied business and geography risk. List risk is ignored here, as this mapping is where we're removing list risk.
10900	Account_WatchListStage2EffectiveRisk	This datamap calculates all risk related values like Effective Risk of Acct and applies the Effective Risk, List Risk to the ACCT record.
10910	WatchListStagingTable2_WatchListStage2IntrlId	This datamap calculates the Effective Risk of all entities and applies the Effective Risk and the List Risk to the entity record based on NTITY_INTRL_ID.
10920	BackOfficeTransaction_EffectiveAcctivityRiskUpd	This datamap updates the risk related values to all parties involved in Back Office Transaction 1) Select risk values from BACK_OFFICE_TRXN, ACCT, Offset Account in the sub query. 2) Derive the effective and activity risks from the transaction. 3) Update BACK_OFFICE_TRXN table using BO_TRXN_SEQ_ID in the main query.
10930	SettlementInstruction_EntityAcctivityRiskUpd	This datamap updates Entity Risk and Activity Risk in INSTRUCTION table
10940	FrontOfficeTransactionPartyRiskStage_EntityActivityRiskInsert	This datamap populates the Effective Risk and Activity Risk related values to all the parties in FO_TRXN_PARTY_RISK_STAGE table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
40100	InsuranceTransaction_EntityAcctivity RiskUpd	This datamap updates the risk related values to all parties in Insurance Transaction. 1) Select different risk related values from various tables like watchlist, external entity and derived address etc. 2) Updates Entity Risk and Activity Risk in INSURANCE_TRXN table.
20010	CorrespondentBank_JurisdictionUpd	This datamap updates the JRSDCN_CD and BUS_DMN_LIST_TX for an existing client bank record where either the JRSDCN_CD or the BUS_DMN_LIST_TX is null.
20020	CorrespondentBank_AcctJurisdiction ReUpd	This datamap updates the jurisdiction for CLIENT_BANK (Correspondent Bank).
20030	FinancialInstitution_InstNameUpd	This datamap updates INSTN_NM for an existing INSTN_MASTER record.
10955	AccountGroup_InvestmentObjectiveU pd	This datamap updates Investment Objective column in Account Group table.
10960	AccountGroup_JurisdictionUpd	This datamap updates the primary account in a HH with the jurisdiction & business domain present in Account table for it.
10970	TransactionPartyCrossReference_Bac kOfficeTransaction	This datamap is used to build the record for Transaction Party Cross Reference table from today's Back Office Transactions. 1) Select the set of information from today's Back Office Transactions and insert records in Transaction Party Cross Reference table. 2) Parameter ProcessTransactionXRefFlag = 'N' or 'Y' accordingly.
10980	CashTransaction_FrontOfficeTransact ion	This datamap is used to build the record for Cash Transaction Table from today's Front Office Transaction and Front Office Transaction Party. 1) Select the set of Cash Transaction categories information from today's Front Office Transaction and Front Office Transaction Party to Insert records In Cash Transaction Table. 2) Some fields are not null-able. The NVL function is used in the SQL to plug the default values in place of a null. Also, various "NB" fields are set to zero whenever they are null in the expression prior to the inserting them into the target table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
10990	MonetaryInstrumentTransaction_FrontOfficeTransaction	This datamap select the set of information from today's Front Office Transaction and Front Office Transaction Party to Insert records In Monetary Instrument Transaction Table.
11000	TransactionPartyCrossReference_FrontOfficeTransaction	This datamap is used to build the record for Transaction Party Cross Reference table from today's Front Office Transaction and Front Office Transaction Party. 1) Select the set of information from today's Front Office Transaction and Front Office Transaction Party to Insert records In Transaction Party Cross Reference Table. 2) Some fields are not null-able. The NVL function is used in the SQL to plug the default values in place of a null. Also, various "NB" fields are set to zero whenever they are null in the expression prior to the inserting them into the target table. 3) Parameter ProcessTransactionXRefFlag = 'N' or 'Y' accordingly.
11010	WireTransaction_FrontOfficeTransaction	This datamap is used to build the record for Wire Transaction Table from today's Front Office Transaction and Front Office Transaction Party. 1) Select the set of Wire Transaction categories information from today's Front Office Transaction and Front Office Transaction Party to Insert records In Wire Transaction Table. 2) Some fields are not null-able. The NVL function is used in the SQL to plug the default values in place of a null. Also, various "NB" fields are set to zero whenever they are null in the expression prior to the inserting them into the target table. 3) Parameter ProcessBankToBank = 'N' or 'Y' accordingly.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
11020	WireTransactionInstitutionLeg_FrontOfficeTransaction	This datamap is used to build the record for Wire Transaction Institution Leg Table from today's Front Office Transaction and Front Office Transaction Party. 1) Select the set of Wire Transaction categories and it should have more than 1 leg information from today's Front Office Transaction and Front Office Transaction Party to Insert records In Wire Transaction Institution Leg Table. 2) Some fields are not null-able. The NVL function is used in the SQL to plug the default values in place of a null. Also, various "NB" fields are set to zero whenever they are null in the expression prior to the inserting them into the target table. 3) Parameter ProcessBankToBank = 'N' or 'Y' accordingly.
11030	CashTransaction_FrontOfficeTransactionRevAdj	This datamap adjusts the reversals for Cash Transaction table. 1) Select the set of information from today's Front Office Transaction to update records with columns CXL_PAIR_TRXN_INTRL_ID, REBKD_TRXN_INTRL_ID in Cash Transaction table.
11040	MonetaryInstrumentTransaction_FrontOfficeTransactionRevAdj	This datamap adjusts the reversals for front office transaction tables in Monetary Instrument Transaction table
11050	WireTransaction_FrontOfficeTransactionRevAdj	This datamap adjusts the reversals for Wire Transaction table. 1) Select the set of information from today's Front Office Transaction to update records with columns CXL_PAIR_TRXN_INTRL_ID, REBKD_TRXN_INTRL_ID in Wire Transaction table.
11060	TrustedPair_StatusEXPUpd	This datamap selects Trusted Pair Records From Kdd_Trusted_Pair Table Which Are To Be Expired, set the Status Code to 'EXP' in Kdd_Trusted_Pair table.
11070	TrustedPairMember_AcctExtEntEffecRiskUpd	This datamap selects The Trusted Pair Records From Kdd_Trusted_Pair Table Which Are Active, and get the trusted Pair parties from kdd_trusted_pair_mbr table with their effective risk and new effective risks from the base tables (i.e. ACCT and EXTERNAL_ENTITY tables) and updates kdd_trusted_pair_mbr table for columns ACCT_EFCTV_RISK_NB, EXTRL_NTITY_EFCTV_RISK_NB for parties whose risk got changed.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
11080	TrustedPair_StatusRRCInsert	This datamap sets the status of a Trusted Pair to expire based on its Expiry Date. Also, if $$TP_RISK_REVIEW_FLAG is set to 'Y' then this mapping reviews/updates the risks for IA and EE parties associated with trusted pairs to reflect the latest risk as in the base tables. If they have increased by substantial amount to move them to a next risk zone it is recommending risk cancellation (RRC).
11090	TrustedPair_StatusRRCUpd	This datamap gets the trusted Pair parties from kdd_trusted_pair_mbr table with their effective risk and new effective risks from the base tables (i.e. ACCT and EXTERNAL_ENTITY tables).Update kdd_trusted_pair table with two columns REVIEW_DT, REVIEW_REASON_TX for existing RRC record.
11100	ApprovalActionsAudit_TrustedPair	This datamap inserts auditing records in KDD_APRVL_ACTVY_AUDIT table. 1) Inserts the EXP record of kdd_trusted_pair table in the KDD_APRVL_ACTVY_AUDIT table 2) Inserts RRC record either which is inserted or updated in KDD_TRUSTED_PAIR with sysdate as review date
11110	TrustedPairMember_StatusRRCInsert	This datamap sets the status of a Trusted Pair to expire based on its Expiry Date. Also, if $$TP_RISK_REVIEW_FLAG is set to 'Y' then this mapping reviews/updates the risks for IA and EE parties associated with trusted pairs to reflect the latest risk as in the base tables. If they have increased by substantial amount to move them to a next risk zone it is recommending risk cancellation (RRC).
11120	BackOfficeTransaction_TrustedFlags Upd	This datamap flags the Back Office Transactions as Trusted or Not Trusted based on entry in the kdd_trusted_pair and kdd_trusted_pair_mbr tables. It only looks at today's transactions. 1) Select the set of information from today's Back Office Transactions, Trusted Pair and Trusted Pair Member Details to update records with columns TRSTD_TRXN_FL, ACCT_OFFSET_ACCT_TRSTD_FL in Back Office Transactions table.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
11130	InsuranceTransaction_TrustedFlagsUpd	This datamap flags today's Insurance Transaction as Trusted or Not Trusted based on entry in the kdd_trusted_pair and kdd_trusted_pair_mbr tables. It only looks at today's transactions. 1) Select the set of information from today's Insurance Transaction and Trusted Pair Member Details to update records with columns TRSTD_TRXN_FL, NSRN_PLCY_ID_CNTRPTY_ID_FL in Insurance Transaction table.
11140	MonetaryInstrumentTransaction_TrustedFlagsUpd	This datamap flags the Monetary Instruction transactions as trusted or not trusted based upon entry in the kdd_trusted_pair and kdd_trusted_pair_mbr tables. It only looks at today's transactions.
11150	WireTransaction_TrustedFlagsUpd	This datamap flags the Wire Transactions as Trusted or Not Trusted based on entry in the kdd_trusted_pair and kdd_trusted_pair_mbr tables. It only looks at today's transactions. 1) Select the set of information from today's Wire Transactions, Trusted Pair and Trusted Pair Member Details to update records with columns TRSTD_TRXN_FL, ORIG_BENEF_TRSTD_FL, ORIG_SCND_BENEF_TRSTD_FL, SCND_ORIG_BENEF_TRSTD_FL,SCND_ORIG_SCND_BENEF_TRSTD_FL in Wire Transaction table.
50050	CustomerDailyProfile_BOT	This datamap aggregates Back Office Transaction data by Customer and Date and updates into CUST_SMRY_DAILY table.
50060	CustomerDailyProfile_FOTPS	This datamap aggregates Front Office Transaction data by Customer and Date and updates into CUST_SMRY_DAILY table.
50070	InstitutionalAccountDailyProfile_DEAL	This datamap updates INSTL_ACCT_SMRY_DAILY table from Deal, grouping by account and data dump date.
50080	CustomerDailyProfile_DEAL	This datamap updates CUST_SMRY_DAILY table from Structured Deal, grouping by customer and data dump date.
50090	InstitutionalAccountDailyProfile_INST	This datamap updates INSTL_ACCT_SMRY_DAILY table from Instruction, grouping by account and data dump date.
50100	CustomerDailyProfile_INST	This datamap updates CUST_SMRY_DAILY table from Instruction data, grouping by Customer and data dump date.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
50110	InstitutionalAccountDailyProfile_CorpAction	This datamap aggregates institutional trading activity, grouping by Account ID and data dump date.
50120	CustomerDailyProfile_CorpAction	This datamap aggregates Corporate Action trading activity, grouping by Customer ID.
50130	InstitutionalAccountDailyProfile_Trade	This datamap updates INSTL_ACCT_SMRY_DAILY table from Trade, grouping by account and data dump date.
50140	CustomerDailyProfile_Trade	This datamap updates CUST_SMRY_DAILY table from Trade data, grouping by customer and data dump date.
60100	ManagedAccountDailyProfile_SameDayTrade	This datamap is used for the daily aggregation of the block allocation day trades data. This populates the managed account daily summary.
60110	ManagedAccountDailyProfile_Trade	This datamap is used for the daily aggregation of the block allocation trades data. This populates the managed account daily summary.
60120	ManagedAccountDailyProfile_BOT	This datamap populates MANGD_ACCT_SMRY_DAILY table using Back Office Transaction.
11170	AccountDailyProfile-Transaction	This datamap populates the table ACCT_TRXN_SMRY_DAILY using both Front office and Back Office transaction for that account on current processing date.
11180	AccountProfile_Trade	This datamap populates the table ACCT_SMRY_MNTH using ACCT_TRADE_SMRY_DAILY table for that account starting from Month Start date till current processing date.
11190	AccountProfile_Transaction	This datamap populates the table ACCT_SMRY_MNTH using ACCT_TRXN_SMRY_DAILY table for that account starting from Month Start date till current processing date.
11200	AccountProfile_Stage	This datamap populates the table ACCT_SMRY_MNTH using ACCT_PRFL_STAGE table for that account starting from Month Start date till current processing date.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
11210	AccountProfile_Position	This datamap populates the table ACCT_SMRY_MNTH using ACCT_POSN table for that account starting from Month Start date till current processing date. Updates values by calculating aggregate values for AGGR_SHRT_PUT_EXPSR_AM, AGGR_SHRT_CALL_EXPSR_AM, SHRT_PUT_EXPSR_RATIO and SHRT_CALL_EXPSR_RATIO for each account internal ID present in ACCT_SMRY_MNTH.
11220	AccountProfile_Balance	This datamap populates the ACCT_SMRY_MNTH table using ACCT_BAL_POSN_SMRY. If there is already record in Account summary Month for Account and Month Start Date, then it will update the record. Else it will do insert, remaining columns defaulted to 0.
60130	HouseholdProfile	This datamap aggregates monthly account summaries into their respective households. All monthly records must be processed each day since account households are subject to change daily.
50150	InstitutionalAccountProfile	This datamap performs Insert or Update of Institutional Account Summary Month Table from its corresponding Daily table. Aggregate daily activity with counts and amounts for the current month. If already record exists for the account in the current month, the datamap will update the record, else insert a new record.
50160	CustomerProfile	This Datamap loads into CUST_SMRY_MNTH from CUST_SMRY_DAILY table. Check for the customer record exists for t he month, if record not available Insert records in CUST_SMRY_MNTH table
60140	ManagedAccountProfile	This datamap updates the Managed Account Summary Month Table from its corresponding Managed Account Daily Summary table.
20040	CorrespondentBankProfile	This datamap performs daily re-aggregation of the Correspondent Bank Summary Month table out of the account summary month table.
20050	AccountATMDailyProfile	This datamap calculates the total Transaction Amount for Account ATM Daily Profile Select information from Front Office Transaction, Account and Account ATM Daily Profile and insert or update (if record exist) into ACCT_ATM_SMRY_DAILY
11230	ChangeLog_AcctProfileInactivity	This datamap creates Change Log records that indicate a change in an accounts activity level as measured by the sum of deposits, withdrawals, and trades over a configurable time period (months).

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
11240	AccountPeerGroupMonthlyTransactionProfile	This datamap calculates average values and insert into Account Peer Group Monthly Transaction Profile. Select and calculate average values for withdrawal amount and count from ACCT_SMRY_MNTH table Insert the above values into ACCT_PEER_TRXN_SMRY_MNTH.
20060	CorrespondentBankPeerGroupTransactionProfile	This datamaps populate CorrespondentBankPeerGroupTransactionProfile from Client Bank Summary Month. 1) Select set of information from CLIENT_BANK_SMRY_MNTH, CLIENT_BANK_PEER_GRP 2) Data is populated in the target table after aggregating the required columns.
20070	AccountChannelWeeklyProfile	This datamap populates the table ACCT_CHANL_SMRY_WKLY using FO_TRXN, BACK_OFFICE_TRXN table for that account starting from Weekly Start date till current processing date.
40110	InsurancePolicyDailyProfile_InsTrxnInsPolicyBal	This datamap performs inserts or updates of Insurance Policy Summary Daily Table from the Insurance Transaction table on the current processing day.
40120	InsurancePolicyProfile_InsurancePolicyDailyProfile	This datamap performs updates of Insurance Policy Summary Month Table using the values from Insurance Policy Daily Profile table. 1) Records are inserted into Insurance Policy Daily Profile table prior to this datamap execution. 2) This datamap inserts new records or Updates matched records in Insurance Policy Profile table using the values from Insurance Policy Daily Profile table.
50170	CustomerBalance_ActiveOTCTradeCtUpd	This datamap counts the records in the Deal table which has an end date greater than or equal to the current date by customer and update the ACTV_OTC_TRD_CT column in customer balance table.
60150	AccountPositionDerived	This datamap processes account option position pair data and updates the corresponding account position records. Updates are made to attributes relating to uncovered option contracts
60160	AccountBalance_AcctPosnPair	This datamap processes account option position pair data and updates the corresponding account balance records. Updates are made to option market value long attributes.

Table 74: TBAML Datamaps (Continued)

Datamap Number	Datamap Name	Description
60170	AccountBalance_Acctposn	This datamap aggregates current-day security positions by product category and account for update of the account balance record. Rejoins for single update to avoid deadlocks.
60180	HouseholdBalance	This datamap aggregates daily records of account balances data and inserts into household balances table based household group ID.

OFSAA Support Contact Details

Raise a Service Request (SR) in My Oracle Support (MOS) for queries related to OFSAA applications.

Send Us Your Comments

Oracle welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, indicate the title and part number of the documentation along with the chapter/section/page number (if available) and contact the Oracle Support.

Before sending us your comments, you might like to ensure that you have the latest version of the document wherein any of your concerns have already been addressed. You can access My Oracle Support site which has all the revised/recently released documents.