

# Administration Guide

## Oracle Financial Services Trade-Based Anti Money Laundering

*Release 8.0.6.0.0*  
*August 2018*





# **Administration Guide**

## Oracle Financial Services: Trade-Based Anti Money Laundering

*Release 8.0.6.0.0  
August 2018*

Part Number: E98716-01

Oracle Financial Services Software, Inc.  
1900 Oracle Way  
Reston, VA 20190

Part Number: E60570\_01  
First Edition (August 2018)

**Copyright © 2018, Oracle and/or its affiliates. All rights reserved.**

Printed in U.S.A. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise without the prior written permission.

**Trademarks**

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.  
Other names may be trademarks of their respective owners.

Oracle Financial Services Software, Inc.  
1900 Oracle Way  
Reston, VA 20190  
*Phone:* 703-478-9000  
*Fax:* 703-318-6240  
*Internet:* [www.oracle.com/financialservices](http://www.oracle.com/financialservices)



## *Revision History*

The following table describes the revision history of the Administration Guide.

<b>Date</b>	<b>Edition</b>	<b>Description</b>
August 2018	First edition of 8.0.6.0.0	First publication of this document.



# Contents

**Revision History** ..... **v**

**List of Figures** ..... **xvii**

**List of Tables** ..... **xix**

**About this Guide** ..... **xxiii**

Who Should Use this Guide ..... xxiii  
     Prerequisites for an Administrator User ..... xxiii  
 Scope of this Guide ..... xxiii  
 How this Guide is Organized ..... xxiv  
 Where to Find More Information ..... xxiv  
 Conventions Used in this Guide ..... xxv  
 Abbreviations Used in this Guide ..... xxvi

**CHAPTER 1            *About Oracle Financial Services Trade-Based Anti Money Laundering (TBAML)*** ..... **1**

About TBAML ..... 1  
 TBAML Architecture ..... 2  
     Deployment View ..... 2  
     Security View ..... 3  
 Operations ..... 4  
     Start Batch ..... 5  
     Managing Data ..... 5  
     Behavior Detection ..... 5  
     Post-Processing ..... 5  
     End Batch ..... 6  
 Utilities ..... 6  
     Batch Utilities ..... 6  
     Administrative Utilities ..... 7

**CHAPTER 2            *Managing User Administration and Security Configuration*** ..... **9**

About User Administration ..... 9  
 Administrator User Privileges ..... 9  
 User Provisioning Process Flow ..... 10  
     Requirements to Access TBAML ..... 10  
 Managing User Administration ..... 11  
     Managing Identity and Authorization ..... 11

<i>Managing Identity and Authorization Process Flow</i> .....	11
<i>Creating and Authorizing Users and User Groups</i> .....	12
<i>Mapping Users with User Groups</i> .....	12
<b>CHAPTER 3</b> <b>Managing Data</b> .....	<b>13</b>
About Data Management .....	13
Data Loading and Processing Flow Overview.....	13
CSA .....	14
Flat Files .....	14
FSDM.....	14
Datamaps.....	14
Managing Data Loading .....	15
FSDF CSA Data Load.....	15
<i>Overview</i> .....	15
<i>Using Table-to-Table (T2T) in the AAI Data Management Framework</i> .....	15
<i>About AAI T2T Data Loading</i> .....	15
<i>Using Datamaps</i> .....	17
Ingestion Flat File Data Load .....	18
<i>Overview</i> .....	18
<i>Using Behavior Detection Datamaps</i> .....	18
<i>Using Pre-processing and Loading</i> .....	19
<i>Configuring RunDP/RunDL</i> .....	21
<i>Ways of Data Loading</i> .....	21
Encrypting Data Files .....	22
Managing Data Processing.....	23
Generating Change Logs with T2T.....	24
<i>Components of the Change Log</i> .....	24
Generating Change Logs.....	26
Processing Data .....	27
<i>About BD Datamaps</i> .....	28
<i>Derived Datamap Types</i> .....	28
<i>Datamap Categories</i> .....	29
<i>Processing Datamaps</i> .....	29
<i>Example for Internal Dependency</i> .....	29
<i>Example for External Dependency</i> .....	30
<i>AML Brokerage Datamaps</i> .....	30
DataMaps .....	30
<i>AML Banking Datamaps</i> .....	30
<i>Trade Finance Datamaps</i> .....	30
Managing Data For BD Applications.....	31
Post Load Changes .....	31
<b>CHAPTER 4</b> <b>Behavior Detection Jobs</b> .....	<b>33</b>
About the OFSBD Job Protocol .....	33
Understanding the OFSBD Job Protocol .....	34
Understanding the Dispatcher Process.....	34

Understanding the MANTAS Process .....	34
Applying a Dataset Override.....	35
<i>Configuring the Dataset Override Feature</i> .....	35
Performing Dispatcher Tasks .....	35
Setting Environment Variables .....	36
<i>About the System.env File</i> .....	36
Starting the Dispatcher.....	37
Stopping the Dispatcher .....	37
Monitoring the Dispatcher .....	38
Performing Job Tasks .....	38
Understanding the Job Status Codes .....	39
Starting Behavior Detection Jobs .....	39
Starting Jobs Without the Dispatcher.....	39
Restarting a Job .....	40
Restarting Jobs Without the Dispatcher.....	41
Stopping Jobs.....	41
Monitoring and Diagnosing Jobs.....	41
Clearing Out the System Logs .....	42
Clearing the Dispatch Log.....	42
Clearing the Job Logs .....	43
Recovering Jobs from a System Crash .....	43
Executing Batches Through the OFSAAI User Interface .....	43
Adding Behavior Detection Batches.....	44
Setting Up Ingestion through AAI.....	45
Adding Tasks to a BD Batch.....	46
Setting Task Precedence.....	47
Running a Single Task Using a Batch .....	48
Scheduling a Batch Once .....	49
Scheduling a Daily Batch .....	50
Scheduling a Weekly Batch .....	51
Configuring a Monthly Batch.....	52
Monitoring a Batch After Execution .....	53
Canceling a Batch After Execution .....	54
Re-starting a Batch.....	55
Re-running a Batch .....	56
Managing the Batch Processing Report.....	57
Managing the View Log .....	57
Starting a Batch Run .....	58
Ending a Batch Run.....	62
Executing a Batch Run.....	63
<b>CHAPTER 5</b> <b>Post-Processing Tasks</b> .....	<b>67</b>
About Post-Processing .....	67
Order of Running Post-Processing Administrative Tasks.....	68

Match Scoring .....	68
Running the Match Scoring Job.....	68
Alert Creation .....	68
Running the Alert Creation Job .....	69
<i>To Run Multi-match Alert Creator.....</i>	<i>69</i>
<i>To Run Single Match Alert Creator.....</i>	<i>69</i>
Understanding Advanced Alert Creator Configuration .....	69
<i>Advanced Rules .....</i>	<i>69</i>
<i>Grouping Algorithms.....</i>	<i>69</i>
Alert Scoring.....	70
Running the Alert Scoring Job.....	70
Highlight Generation .....	71
Historical Data Copy.....	71
Alert Correlation .....	72
Running the Alert Correlation Job .....	72
Understanding Alert Correlation Configuration .....	72
<i>Business Entity Paths .....</i>	<i>73</i>
Correlation Rules.....	75
<i>Activating or Deactivating Correlation Rules.....</i>	<i>77</i>
<i>Custom Scoring Rules.....</i>	<i>77</i>
<i>Configuring Rules.....</i>	<i>81</i>
<i>Structure of the Configuration Table .....</i>	<i>81</i>
<i>Structure of the Temporary Table.....</i>	<i>81</i>
<i>Configuring Custom Rules .....</i>	<i>82</i>
<i>Sample Alert Correlation Rules.....</i>	<i>82</i>
<i>Displaying Alert-to-Business Entity Path Details on the User Interface.....</i>	<i>82</i>
<b>CHAPTER 6           Managing Batch Processing Utilities.....</b>	<b>85</b>
About Batch Processing Utilities.....	85
Managing Common Resources for Batch Processing Utilities.....	87
Install Configuration.....	87
<i>Log4j2.xml Configuration.....</i>	<i>98</i>
Managing Annual Activities .....	108
Loading Holidays .....	108
Loading Non-business Days .....	110
Managing Alert Purge Utility .....	111
Directory Structure .....	111
Logs.....	112
Precautions.....	112
Using the Alert Purge Utility .....	113
<i>Configuring the Alert Purge Utility .....</i>	<i>113</i>
<i>Executing the Alert Purge Utility.....</i>	<i>121</i>
<i>Processing for Purging.....</i>	<i>121</i>
<i>Automatic Restart Capability.....</i>	<i>122</i>
Sample Alert Purge Processes .....	122
<i>Example 1.....</i>	<i>122</i>

<i>Example 2</i> .....	123
Managing Batch Control Utility .....	123
Batches in Behavior Detection.....	124
Directory Structure .....	125
Logs.....	125
Using the Batch Control Utility .....	125
<i>Configuring the Batch Control Utility</i> .....	126
<i>Setting Up Batches</i> .....	126
<b>Single Batch</b> .....	127
<b>Single Site Intra-day Processing</b> .....	127
<b>Multiple Countries</b> .....	127
<i>Starting a Batch Process Manually</i> .....	128
<i>Processing for Batch Start</i> .....	129
<i>Ending a Batch Process</i> .....	130
<i>Processing for End Batch</i> .....	130
<i>Identifying a Running Batch Process</i> .....	131
<b>To Obtain a Batch Name</b> .....	131
<i>Obtaining a Batch Name</i> .....	131
Managing Calendar Manager Utility. ....	132
Directory Structure .....	132
Logs.....	132
Calendar Information.....	132
Using the Calendar Manager Utility .....	133
<i>Configuring the Calendar Manager Utility</i> .....	133
<i>Executing the Calendar Manager Utility</i> .....	134
<b>Starting the Utility Manually</b> .....	134
<i>Updating the KDD_CAL Table</i> .....	134
<i>Configuring Case Age</i> .....	136
Managing Data Retention Manager.....	136
Directory Structure .....	137
Logs.....	137
Processing Flow.....	138
Using the Data Retention Manager.....	138
<i>Configuring the Data Retention Manager</i> .....	139
<i>Executing the Data Retention Manager</i> .....	140
<b>Running the Data Retention Manager</b> .....	141
<i>Creating Partitions</i> .....	141
<i>Maintaining Partitions</i> .....	142
<b>Managing Daily Partitioning Alternative</b> .....	143
<b>Partition Structures</b> .....	143
<b>Recommended Partition Maintenance</b> .....	143
<b>Managing Alternative Monthly Partition</b> .....	143
<i>Maintaining Indexes</i> .....	144
Utility Work Tables.....	144
<i>KDD_DR_MAINT_OPRTN Table</i> .....	144
<i>KDD_DR_JOB Table</i> .....	145
<i>KDD_DR_RUN Table</i> .....	145
Database Statistics Management .....	146

Logs .....	146
Using Database Statistics Management .....	146
Managing ETL Process for Threshold Analyzer Utility .....	147
Running Threshold Analyzer .....	148
Managing Truncate Manager .....	149
Logs .....	149
Using the Truncate Manager .....	149
<b>CHAPTER 7</b>	<b>Managing Administrative Utilities .....</b>
	<b>151</b>
About Administrative Utilities .....	151
Common Resources for Administrative Utilities .....	151
Managing Scenario Migration Utility .....	151
Logs .....	152
Using the Scenario Migration Utility .....	152
<i>Configuring the Scenario Migration Utility</i> .....	152
<i>Configuring the Environment</i> .....	155
<i>Configuring General Scenario Migration</i> .....	155
<i>Configuring Scenario Extraction</i> .....	155
<i>Extracting Scenario Metadata</i> .....	157
<i>Loading Scenario Metadata</i> .....	158
Scenario Migration Best Practices .....	158
<i>Process Overview</i> .....	159
<i>Best Practices</i> .....	159
<i>Sequences to Modify</i> .....	159
Managing the Threshold Editor .....	162
Threshold Sets .....	162
Inactive Thresholds .....	163
<i>Mutually Exclusive Thresholds</i> .....	163
<i>Additional Scenario Thresholds</i> .....	163
About the Threshold Editor Screen Elements .....	163
<i>Search Bar</i> .....	164
<i>&lt;Scenario–Threshold Set&gt; Area</i> .....	164
Using the Threshold Editor .....	166
<i>Changing a Scenario Threshold</i> .....	167
<i>Resetting a Scenario Threshold to the Sample Values</i> .....	167
<i>Viewing a Scenario Threshold’s History</i> .....	167
<i>Viewing Expanded Comments</i> .....	167
<b>CHAPTER 8</b>	<b>Configuring EDQ .....</b>
	<b>169</b>
About EDQ .....	169
EDQ Configuration Process Flow .....	170
General EDQ Configurations .....	171
Importing TBAML Projects .....	172
Configuring Watch List Management .....	172
<i>Preparing Watch List Data</i> .....	173



<i>Setting Up Private Watch List</i> .....	173
<i>The OEDQ Config Folder</i> .....	173
<i>Showing Watch List Staged Data/Snapshots in the Server Console UI</i> .....	174
<i>Configuring Match Rules</i> .....	174
<i>Configuring a Job</i> .....	174
Filtering Watch List Data.....	175
<i>Enabling Watch List Filtering</i> .....	175
<i>Configuring Watch List Filtering</i> .....	175
<i>Primary and Secondary Filtering, and Linked Records</i> .....	176
<i>Setting Multiple Values for Primary and Secondary Filters</i> .....	176
<i>Filtering World Check Data</i> .....	176
Setting Filtering Options in the Run Profiles.....	177
<i>Setting Primary Filters and Linked Profiles in the Watchlist Management project</i> .....	177
<i>Setting Secondary Filters in the TBAML project</i> .....	178
<i>Screening All Data Using Sanctions Rules</i> .....	178
Port, Goods, Name and Address Screening.....	178
<i>Configuring Port, Goods, Name and Address Screening</i> .....	178
<i>Bad BICs Reference Data</i> .....	178
<i>Blacklisted Cities Reference Data</i> .....	179
<i>Blacklisted Countries Reference Data</i> .....	180
<i>Stop Keywords Reference Data</i> .....	181
<i>Goods Prohibition Reference Data</i> .....	181
<i>Ports Prohibition Reference Data</i> .....	182
<i>Extending Prohibition Screening</i> .....	183
<b>CHAPTER 9</b> <b>Creating JSON</b> .....	<b>185</b>
Structure of a JSON.....	186
Creating JSON for SWIFT Messages with Sequences.....	189
Creating Message Elements.....	189
Configuring SWIFT Message Blocks.....	189
<i>Configuring the Basic Header Block</i> .....	189
<i>Configuring the Application Header Block</i> .....	190
<i>Configuring the User Header Block</i> .....	192
<i>Configuring the Text Block</i> .....	193
<i>Configuring the Trailer Block</i> .....	195
Example of MT101 with Sequences.....	196
Creating JSON for SWIFT Messages without Sequences.....	196
Creating Message Elements.....	196
Configuring SWIFT Message Blocks.....	196
<i>Configuring the Text Block</i> .....	196
Example of MT101 without Sequences.....	197
Creating JSON for SWIFT messages with the List of Values (LOV) Attribute.....	197
<b>APPENDIX A</b> <b>Logging</b> .....	<b>199</b>
About System Log Messages.....	199
Message Template Repository.....	199

Logging Levels .....	200
Logging Message Libraries .....	200
Verifying the Schema Creator Log Files .....	200
Administration Tools .....	200
Database .....	201
Scenario Manager .....	201
Services .....	201
Alert Management .....	201
Web Server Logs .....	201
Application Server logs .....	201
Database Objects Logs .....	201
Ingestion Manager .....	201
Logging Configuration File .....	202
Sample Configuration File .....	203
Configurable Logging Properties .....	204
Monitoring Log Files .....	205
<b>APPENDIX B</b> <b>Oracle Software Updates .....</b>	<b>207</b>
Oracle Software Updates - Hotfix .....	207
Hotfix Effect on Customization .....	207
User Interface .....	207
Scenarios .....	208
<b>APPENDIX C</b> <b>User Administration .....</b>	<b>209</b>
Managing User Groups and User Roles .....	209
Managing User Groups .....	209
Defining User Group Maintenance Details .....	209
Adding New User Group Details .....	210
Mapping Users to User Groups .....	210
Mapping User Group(s) to Domain(s) .....	210
Mapping a User to a Single User Group .....	210
<i>Mapping a User to Multiple User Groups</i> .....	210
<i>Mapping a User to an Organization</i> .....	210
<i>Mapping a Function to a Role</i> .....	211
Defining User Access Properties and Relationships .....	211
<b>APPENDIX D</b> <b>Managing Data .....</b>	<b>215</b>
CSA Ingestion .....	215
CSA Datamaps .....	215
List of Data Quality Group Names and T2T Names .....	216
Group Dependencies .....	223
Flat File Ingestion .....	224
BDF.xml File Parameters .....	224

BD Ingest DIS Data Files by Group .....	225
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.....	225
Behavior Detection Flat File Interface .....	226
Pre-processing & Loading Directory Structure .....	237
Directory Structure Descriptions .....	238
jars Subdirectory.....	238
scripts Subdirectory.....	239
<i>config</i> Subdirectory.....	240
<i>Data Ingest Properties Configuration File</i> .....	241
<i>Data Ingest XML Configuration File</i> .....	242
<i>Data Ingest Custom XML Configuration File</i> .....	252
data Subdirectory.....	253
<i>data/errors</i> Subdirectory.....	253
<i>data/backup</i> Subdirectory .....	253
<i>data/firm</i> Subdirectory .....	254
extract Subdirectory .....	254
transform Subdirectory .....	254
load Subdirectory .....	254
inbox Subdirectory.....	254
logs Subdirectory.....	255
BD Directory Structure .....	255
<i>Scripts</i> .....	256
<i>Logs</i> .....	256
<i>Parameters</i> .....	257
<i>Config</i> .....	258
<i>BDF.xml Configuration Parameters</i> .....	259
<i>BD Datamap Configuration File</i> .....	265
<b>APPENDIX E            Processing Derived Tables and Fields .....</b>	<b>267</b>
Customizing Scripts .....	267
Derivations .....	268
AccountDailySecurityProfile .....	269
Ingestion Timeline - Intra-Day Ingestion Processing.....	270
Guidelines for Duplicate Record Handling.....	271
Data Rejection During Ingestion .....	271
Rejection During the Pre-processing Stage.....	272
<i>Data Type</i> .....	272
<i>Missing Data</i> .....	272
<i>Referential Integrity</i> .....	272
<i>Domain Values</i> .....	272
Rejection During the Transformation Stage.....	272
<i>Lost Events</i> .....	273
<i>Out-of-Sequence Events</i> .....	273

Rejection During the Loading Stage .....	274
Alternatives to Standard Data Management Practices.....	274
Data Management Archiving .....	274
Fuzzy Name Matcher Utility .....	274
Using the Fuzzy Name Matcher Utility.....	274
<i>Configuring the Fuzzy Name Matcher Utility</i> .....	274
<i>Executing the Fuzzy Name Matcher Utility</i> .....	278
Refresh Temporary Tables Commands.....	278
Use of Control Data .....	279
Prerequisites for Using Control Data.....	279
Control Data Management .....	279
Loading Control Data Thresholds .....	280
Running Behavior Detection on Control Data .....	280
<i>Important Notes</i> .....	280
<b>APPENDIX F</b> <b><i>TBAML Datamap Details</i></b> .....	<b>283</b>
Trade Finance Datamaps.....	283
Trade Finance - Pre-Watch List Datamaps.....	283
Trade Finance- Post-Watch List Datamaps.....	286
Watchlist Datamaps .....	287
Processing BD Datamaps.....	298
<b>APPENDIX G</b> <b><i>Configuring Administration Tools</i></b> .....	<b>317</b>
<b>APPENDIX H</b> <b><i>Watch Lists</i></b> .....	<b>319</b>
HM Treasury Reference Data.....	319
OFAC Reference Data .....	319
EU Reference Data .....	320
UN Reference Data.....	320
World-Check .....	320
Dow Jones Watchlist.....	321
Dow Jones Anti-Corruption List .....	321
Accuity Reference Data .....	322
Using the Accuity Group File .....	322
New Alerts Resulting from Use of the Group File.....	323
PLI Reference Data.....	323
PLI Attributes .....	324
Individual Private Watch List Input Attributes .....	324
Entity Private Watch List Input Attributes.....	327
<b>APPENDIX I</b> <b><i>Match Score Rules</i></b> .....	<b>331</b>

# List of Figures

Figure 1. TBAML Architecture.....	2
Figure 2. TBAML Architecture - Deployment View .....	3
Figure 3. Security View.....	4
Figure 4. User Provisioning Process Flow .....	10
Figure 5. Managing Identity and Authorization Process Flow .....	11
Figure 6. Data Loading and Processing Flow Overview .....	14
Figure 7. Data Management Flow Using CSA .....	15
Figure 8. Process Flow for AAI T2T.....	16
Figure 9. Data Loading Flow Using Flat File Interface .....	18
Figure 10. Input and Output Directories .....	20
Figure 11. TCS Data Loading Process.....	21
Figure 12. Data Loading For TBAML Application.....	31
Figure 13. Batch Maintenance Page .....	45
Figure 14. Add Batch Definition page.....	45
Figure 15. Batch Maintenance Page .....	47
Figure 16. Batch Maintenance page .....	47
Figure 17. Task Precedence Mapping.....	48
Figure 18. Batch Execution page.....	49
Figure 19. Task Mapping Window .....	49
Figure 20. Scheduling a Batch Once .....	50
Figure 21. Scheduling a Daily Batch.....	51
Figure 22. Scheduling a Weekly Batch.....	52
Figure 23. Configuring a Monthly Batch.....	53
Figure 24. Batch Monitor Page .....	54
Figure 25. Batch Cancellation Page.....	55
Figure 26. Re-starting a Batch .....	55
Figure 27. Re-running a Batch .....	56
Figure 28. Batch Processing Report .....	57
Figure 29. View Log .....	58
Figure 30. Run page.....	59
Figure 31. Run Definition page.....	59
Figure 32. Process page.....	60
Figure 33. Process Definition page .....	60
Figure 34. Component Selector page.....	61
Figure 35. Process page.....	62
Figure 36. Component Selector page.....	63
Figure 37. Run page.....	64
Figure 38. Fire Run .....	64
Figure 39. Managing Database Activities with Utilities.....	86
Figure 40. Sample install.cfg File.....	98

Figure 41. Sample Logging Information in the Log4j2.xml File .....	107
Figure 42. Sample KDD_CAL_HOLIDAY Table Loading Script .....	109
Figure 43. Sample KDD_CAL_WKLY_OFF Table Loading Script.....	110
Figure 44. Configuration Information .....	115
Figure 45. Configuration Information .....	118
Figure 46. Configuring Batch Control Utility.....	126
Figure 47. Sample KDD_PRCNG_BATCH_HIST Table—Batch Start Status.....	130
Figure 48. Sample KDD_PRCNG_BATCH_HIST Table—Batch End Status .....	131
Figure 49. Database Partitioning Process.....	138
Figure 50. install.cfg Data Retention Manager Configuration .....	139
Figure 51. Sample install.cfg File for Scenario Migration .....	154
Figure 52. Search Bar.....	164
Figure 53. <Scenario-Threshold Set> Area.....	164
Figure 54. Scenario Test Execution window .....	166
Figure 55. Example Expanded Comment Dialog Box .....	168
Figure 56. EDQ Configuration Process Flow.....	170
Figure 57. Updating the Schema Details .....	171
Figure 58. Sample Logging Configuration File.....	203
Figure 59. User Authorization Model.....	212
Figure 60. Data Management Subsystem Directory Structure.....	237
Figure 61. BD Subsystem Directory Structure .....	256
Figure 62. Intra-Day Data Management Processing .....	270
Figure 63. Sample BDF.xml Configuration Parameters .....	276
Figure 64. Edit Data Store .....	323

## List of Tables

Table 1. Conventions Used in this Guide .....	xxv
Table 2. Abbreviations Used in this Guide.....	xxvi
Table 3. Access Permissions for Administrators .....	9
Table 4. User Provisioning Process Flow .....	10
Table 5. Requirements.....	11
Table 6. Administration Process Flow.....	12
Table 7. TBAML Roles and User Groups .....	12
Table 8. Change Log Components .....	24
Table 9. T2T Change Log Metadata Table Component Values.....	24
Table 10. T2T Change Log Table Component Values .....	25
Table 11. Change Log Parameters.....	27
Table 12. Datamap Table Descriptions.....	29
Table 13. Managing Application Data .....	31
Table 14. OFSBD Job Protocol Shell Scripts.....	33
Table 15. KDD_JOB_TEMPLATE with Sample Job Template Group .....	34
Table 16. OFSBD Environment Variables in system.env File .....	36
Table 17. Database Environment Variables in system.env File.....	36
Table 18. Operating System Environment Variables in system.env File .....	36
Table 19. New Batch Details.....	45
Table 20. Adding Fire Run Details.....	65
Table 21. HDC Configurable Parameters .....	71
Table 22. KDD_BUS_NTITY_PATH (Metadata Table).....	73
Table 23. KDD_BUS_NTITY_PATH_CFG (Metadata Table) .....	74
Table 24. Structure of the Configuration Table .....	81
Table 25. Structure of the Temporary Table .....	82
Table 26. KDD_CAL_HOLIDAY .....	110
Table 27. KDD_CAL_WKLY_OFF .....	111
Table 28. Alert Purge Utility Directory Structure .....	112
Table 29. Alert Purge Utility Parameters.....	116
Table 30. Alert Purge Utility Parameters.....	118
Table 31. Batch Control Utility Directory Structure .....	125
Table 32. KDD_PRCNSG_BATCH Table Contents.....	126
Table 33. Sample KDD_PRCNSG_BATCH Table with Single Batch .....	127
Table 34. Sample KDD_PRCNSG_BATCH Table with Intra-day Processing.....	127
Table 35. KDD_PRCNSG_BATCH_SRC FSDM Columns.....	128
Table 36. Sample KDD_PRCNSG_BATCH Table with Multiple Country Processing.....	128
Table 37. KDD_PRCNSG_BATCH_CONTROL Table Contents.....	129
Table 38. KDD_PRCNSG_BATCH_HIST Table Contents.....	129
Table 39. Calendar Manager Utility Directory Structure .....	132
Table 40. KDD_CAL Table Contents .....	134

Table 41. Data Retention Manager Directory Structure .....	137
Table 42. Data Retention Manager Processing Parameters.....	140
Table 43. Partition Name Formats.....	142
Table 44. BUSINESS.KDD_DR_MAINT_OPRTN Table Contents .....	144
Table 45. BUSINESS.KDD_DR_JOB Table Contents .....	145
Table 46. BUSINESS.KDD_DR_RUN Table Contents .....	146
Table 47. General Scenario Migration Parameters .....	155
Table 48. Scenario Extraction Parameters .....	155
Table 49. Scenario Load Parameters.....	156
Table 50. Environment 1 (Development).....	159
Table 51. Environment 2 (Test/UAT).....	160
Table 52. Environment 3 (PROD).....	160
Table 53. Mutually Exclusive Thresholds .....	163
Table 54. Scenario Test Execution components.....	166
Table 55. Filter Settings.....	175
Table 56. Sample Data for BICs.....	179
Table 57. Sample Data for Blacklisted Cities.....	179
Table 58. Sample Data for Blacklisted Countries .....	180
Table 59. Sample Data for Stop Keywords .....	181
Table 60. Sample Data for Prohibited Goods.....	182
Table 61. Sample Data for Prohibited Ports.....	182
Table 62. Logging Levels .....	200
Table 63: Logging Configuration Files .....	202
Table 64. Configurable Parameters for Common Logging.....	204
Table 65. Function to Role Mapping Details .....	211
Table 66. Relationships between Data Points .....	213
Table 67. CSA Datamaps.....	215
Table 68. Data Quality Group Names and Related T2T Names .....	217
Table 69. Parameters Related to Processing DIS Files .....	224
Table 70. BD Ingest DIS Data Files By Group.....	225
Table 71. Group 1 Interface Ingestion Flat Files.....	226
Table 72. Group 2 Interface Ingestion Flat Files.....	228
Table 73. Group 3 Interface Ingestion Flat Files.....	229
Table 74. Group 4 Interface Ingestion Flat Files.....	230
Table 75. Group 5 Interface Ingestion Flat Files.....	234
Table 76. Group 6 Interface Ingestion for Market Data .....	236
Table 77. Group 7 Interface Ingestion for Trade Finance Data .....	236
Table 78. Data Management Directory Structure Description.....	238
Table 79. Run Scripts by Component.....	239
Table 80. Environment Variable Descriptions.....	240
Table 81. Application Configuration Files .....	240
Table 82. DataIngest.properties File Configuration Parameters .....	241



---

Table 83. DataIngest.xml File Configuration Parameters .....	242
Table 84. Error File Signatures Output by Component .....	253
Table 85. Backed Up Files by Component .....	254
Table 86. Log Files Output by Component.....	255
Table 87. Directory Structure Description.....	255
Table 88. BDF.xml File Configuration Parameters .....	259
Table 89. BD Datamap Configuration Parameters.....	266
Table 90. Utilities .....	268
Table 91. Processing Batch Table Set-up .....	271
Table 92. Fuzzy Name Matcher Utility Configuration Parameters.....	277
Table 93. Dates used by Control Data .....	279
Table 94. Trade Finance - Pre-Watch List Datamaps .....	283
Table 95. Trade Finance - Post-Watch List Datamaps .....	287
Table 96. Watch List Datamaps.....	287
Table 97. AML Banking - Post-Watch List Datamaps .....	296
Table 98. BD Datamaps.....	298
Table 99. Private List for Individuals.....	324
Table 100. Private List for Individuals .....	327



---

# 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:

- [Who Should Use this Guide](#)
- [Scope of this Guide](#)
- [How this Guide is Organized](#)
- [Where to Find More Information](#)
- [Conventions Used in this Guide](#)

## 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.

## Prerequisites for an Administrator User

User must have knowledge of UNIX and LINUX.

## 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.

## ***How this Guide is Organized***

The *Administration Guide*, includes the following chapters:

- *Chapter 1, About Oracle Financial Services Trade-Based Anti Money Laundering (TBAML)*, provides a brief overview of the Oracle Financial Services 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.
- *Chapter 8, Configuring EDQ*, provides information about how to configure EDQ within TBAML.
- *Chapter 9, Creating JSON*, provides information about creating the JSON used by the Swift Parser functionality of TBAML.
- *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 in the BD applications.
- *Appendix F, TBAML Datamap Details*, lists the Datamap XML and their use in TBAML.
- *Appendix G, Configuring Administration Tools* describes how to configure the Administration Tools feature.
- *Appendix H, Watch Lists* provides details of the pre-configured watch lists that can be used by Transaction Filtering within TBAML.
- *Appendix I, Match Score Rules* provides details of the pre-configured watch lists that can be used by Transaction Filtering within TBAML.

## ***Where to Find More Information***

For more information about Oracle Financial Services, refer to the following TBAML application documents, which can be found at [https://docs.oracle.com/cd/E60570\\_01/tbamlhome.htm](https://docs.oracle.com/cd/E60570_01/tbamlhome.htm):

- *Trade-Based Anti Money Laundering Data Interface Specification (DIS)*
- *TBAML Installation 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](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*
- *Swift Parser User Guide*
- *Transaction Filtering Matching 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.

## ***Conventions Used in this Guide***

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

**Table 1. Conventions Used in this Guide**

<b>Convention</b>	<b>Meaning</b>
<i>Italics</i>	<ul style="list-style-type: none"> <li>● Names of books, chapters, and sections as references</li> <li>● Emphasis</li> </ul>
<b>Bold</b>	<ul style="list-style-type: none"> <li>● Object of an action (menu names, field names, options, button names) in a step-by-step procedure</li> <li>● Commands typed at a prompt</li> <li>● User input</li> </ul>
Monospace	<ul style="list-style-type: none"> <li>● Directories and subdirectories</li> <li>● File names and extensions</li> <li>● Process names</li> <li>● Code sample, including keywords and variables within text and as separate paragraphs, and user-defined program elements within text</li> </ul>
<Variable>	<ul style="list-style-type: none"> <li>● Substitute input value</li> </ul>

## ***Abbreviations Used in this Guide***

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

**Table 2. Abbreviations Used in this Guide**

<b>Abbreviation</b>	<b>Description</b>
TBAML	Oracle Financial Services Trade-Based Anti Money Laundering
OFSBD	Oracle Financial Services Behavior Detection
T2T	Table to Table
H2T	Hive to Table
T2H	Table to Hive
AAI	Analytical Applications Infrastructure
CSA	Common Staging Area
FSDM	Financial Services Data Model
BD	Behavior Detection
OFS	Oracle Financial Services
DQ	Data Quality
DT	Data Transformation

# *About Oracle Financial Services Trade-Based Anti Money Laundering (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](#)

## ***About TBAML***

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

- Efficiently screen goods, ports and involved parties extracted from SWIFT MT messages 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).

## 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.

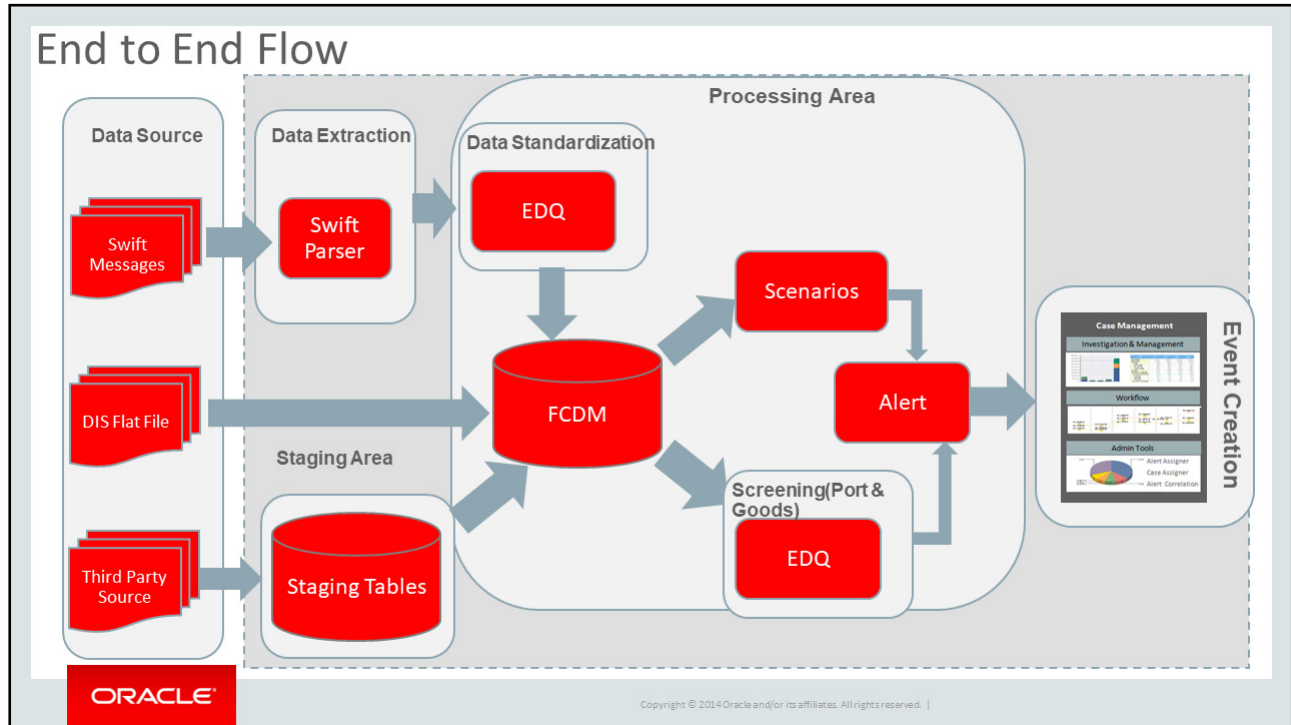


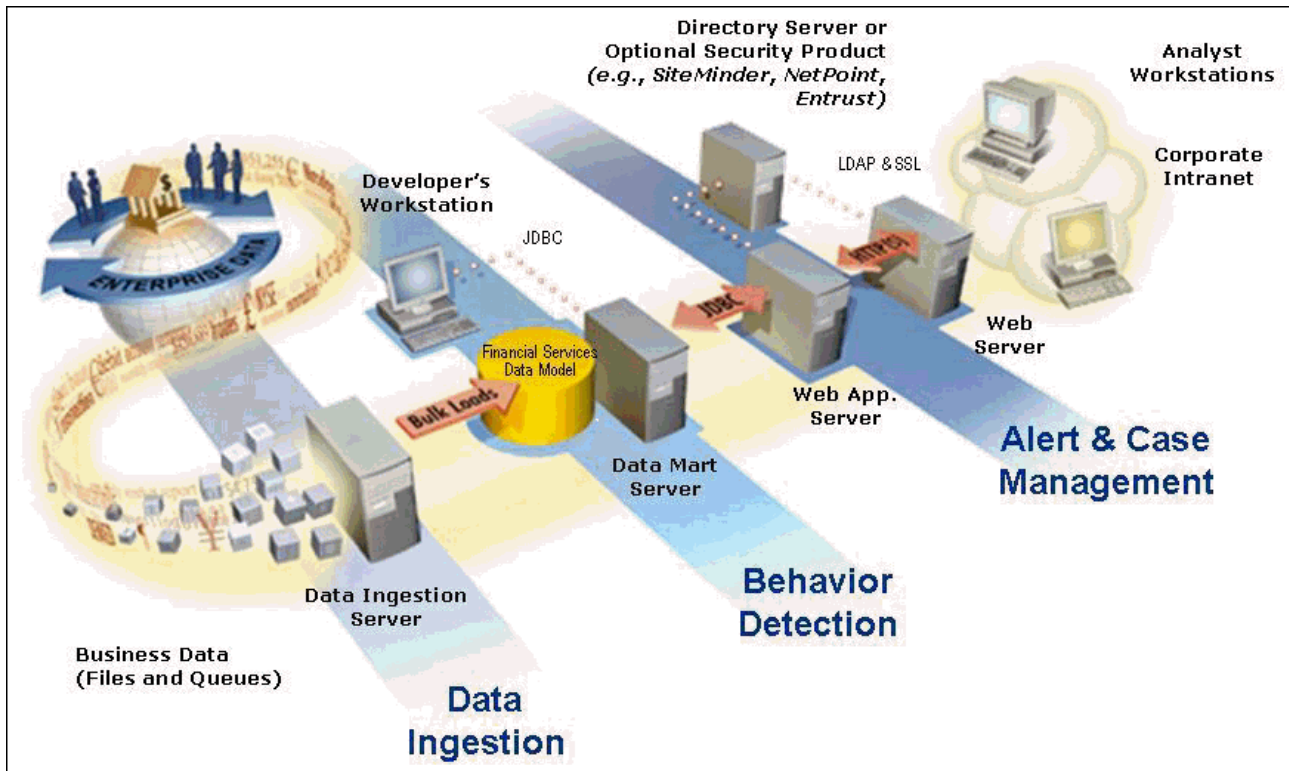
Figure 1. TBAML Architecture

TBAML extracts data from SWIFT messages via the Swift Parser functionality. That data, and data provided by the Oracle client via DIS File or another source, is fed into staging tables and then into the FSDM where the data is either standardized (Port) and screened (Port, Goods, Name and Address) through EDQ functionality, or run through scenarios to generate an FCM event.

## 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 2. TBAML Architecture - Deployment View**

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

## 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.

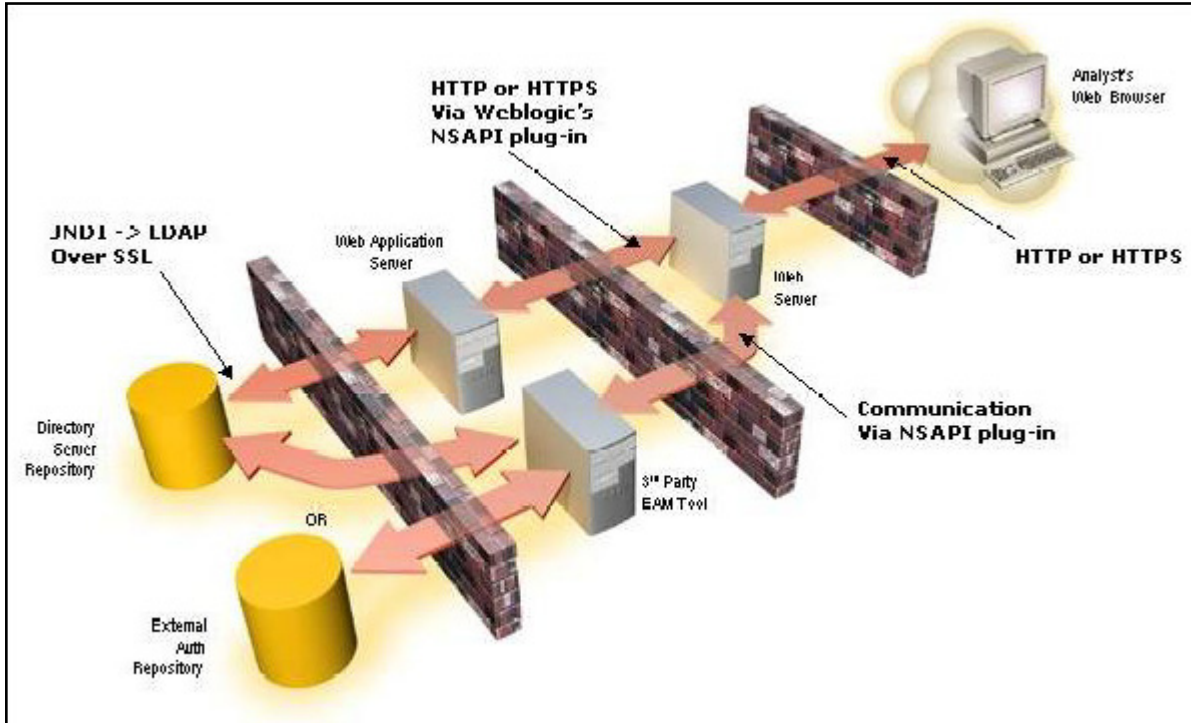


Figure 3. Security View

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.

## 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](#)

## Start Batch

Using the Batch Control Utility, you can manage the beginning of the batch process (see *Chapter 6 - Managing Batch Processing Utilities* for more information).

## 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 *Chapter 3 - Managing Data* for more information).

## 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 FSDM. 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.

## 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.

## 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 *Behavior Detection User Guide* and *Enterprise Case Management User Guide* for more information.

## 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

### 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*.

## 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*.



# *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:

- [About User Administration](#)
- [Administrator User Privileges](#)
- [User Provisioning Process Flow](#)
- [Managing User Administration](#)
- [Adding Security Attributes](#)
- [Mapping Security Attributes to Organizations and Users](#)

## ***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

## ***Administrator User Privileges***

The following table lists the access permissions of the administrators depending on the different product suite under TBAML:

**Table 3. Access Permissions for Administrators**

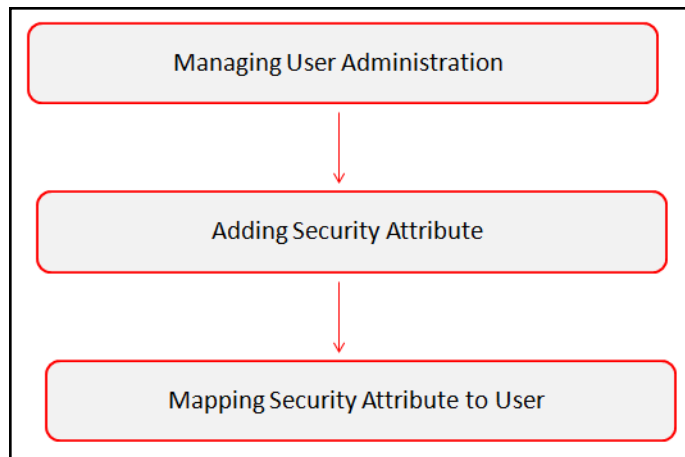
<b>Privileges</b>	<b>TBAML Administrator</b>
User Security Administration	X
Preferences	X
User Administration	X
Security Management System	X
Swift Parser	X
EDQ Jobs	X

**Table 3. Access Permissions for Administrators**

Privileges	TBAML Administrator
Data Management Tools	x
Unified Metadata Manager	x

**Note:** If KYC/FATCA is deployed with BD, the respective Administrator must be mapped with the KYC/FATCA Administrator group, as well for other BD-related access.

## User Provisioning Process Flow



**Figure 4. User Provisioning Process Flow**

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

**Table 4. 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.
Adding Security Attributes	Load security attributes. Security attributes are loaded using either Excel or SQL scripts.
Mapping Security Attributes to Organizations and Users	Map security attributes to users. This is done to determine which security attributes control the user's access rights.

## 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 5. Requirements**

Applications	Conditions
TBAML	<ul style="list-style-type: none"><li>● Set of privileges that associate functional role with access to specific system functions.</li><li>● One or more associated organizational affiliations that control the user's access to events.</li><li>● Relationship to one or more scenario groups.</li><li>● Access to one or more jurisdictions.</li><li>● Access to one or more business domains.</li></ul>
Watch List Management	<ul style="list-style-type: none"><li>● Set of policies that associate functional roles with access to specific system functions.</li><li>● Access to one or more jurisdictions.</li><li>● Access to one or more business domains.</li></ul>

## 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 BD applications.

## Managing Identity and Authorization

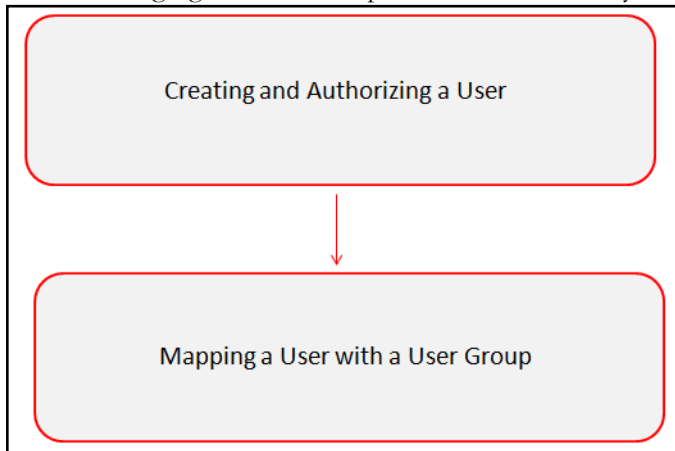
This section explains how to create a user and provide access to Oracle applications.

This section covers the following topics:

- [Managing Identity and Authorization Process Flow](#)
- [Creating and Authorizing Users and User Groups](#)
- [Mapping Users with User Groups](#)

### 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 6. Administration Process Flow**

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

## Creating and Authorizing Users and User Groups

The SYSADMIN 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 *Chapter 9, Oracle Financial Services Analytical Applications Infrastructure 8.0.4 User Guide*.

## 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 SYSADMIN 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 7. TBAML Roles and User Groups**

Role	Group Name	User Group Code
TBAML Administrator	TBAML Administrator User Group	TBAMLADMINISTRATORGRP

---

**Note:** 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**.

---

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

---

This chapter explains how your raw business data can be loaded into the Oracle Financial Services Data Model (FSDM) 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:

- [About Data Management](#)
- [Data Loading and Processing Flow Overview](#)
- [Managing Data Loading](#)
- [Managing Data Processing](#)
- [Managing Data For BD Applications](#)

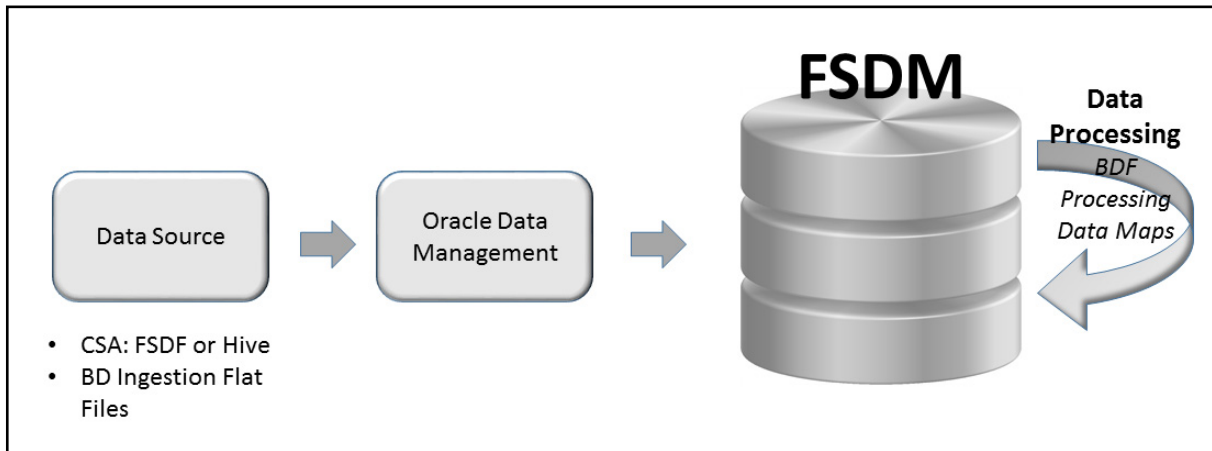
## ***About Data Management***

Data Management consists of two main activities:

- **Data Loading:** Data is loaded into the Financial Services Data Model (FSDM) using various approaches such as Analytical Applications Infrastructure Table-to-table (AAI T2T), Behavior Detection (BD), and Run DP (Data Processing)/Run DL (Data Loading).
- **Data Processing:** Data loaded into the FSDM 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.

## ***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 FSDM from the following data sources:

- Common Staging Area (CSA) in either FSDF or Hive
- BD Flat File Interface

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

## 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 FCCM using BD.

## Flat Files

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

## FSDM

The FSDM 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.

## Datamaps

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

## Managing Data Loading

Your raw business data can be loaded into the Oracle Financial Services Data Model (FSDM) 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](#)

### FSDF CSA Data Load

This section covers the following topics:

- [Overview](#)
- [Using Table-to-Table \(T2T\) in the AAI Data Management Framework](#)
- [Using Datamaps](#)

#### 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. In the CSA approach, you can load data using the Oracle Analytical Application Infrastructure (AAI) Table-to-Table (T2T) Data Management Framework and BD. The following figure provides an overview of the data loading flow using CSA:

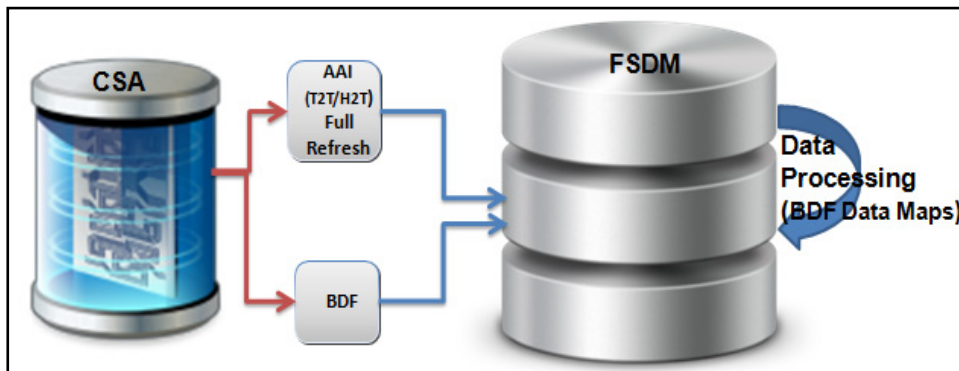


Figure 7. Data Management Flow Using CSA

#### Using Table-to-Table (T2T) in the AAI Data Management Framework

Table-to-Table (T2T) is used in the AAI Framework for data loading. The source for T2T data is the Oracle RDBMS.

#### About AAI T2T Data Loading

AAI (Analytical Applications Infrastructure) is a complete end-to-end Business Intelligence solution. It is a single interface that lets you access your company's operational data and use it to track and respond to business trends. It also facilitates the analysis of processed data.

The AAI framework is the process of retrieving structured data from data sources for further data processing, storage, or migration. The intermediate extraction process is followed by data transformation and metadata addition before exporting it to the Business Data Model. For more information, see *Chapter 2, Section - Data Mapping, Oracle Financial Services Analytical Applications Infrastructure User Guide*.

This section covers the following topics:

- Process Flow for AAI T2T
- Setup Using AAI Batch
- Running Data Quality Batch
- Executing Data Transformation using DT
- Moving Data through T2T
- Executing Jobs
- Ending the Batch

### Process Flow for AAI T2T

The following figure shows the process flow for AAI T2T:



Figure 8. Process Flow for AAI T2T

### Setup Using AAI Batch

**Note:** Ensure that the staging data has the same batch date records.

To start the batch, run the `start_mantas_batch.sh` and `set_mantas_date.sh` scripts. For more information, see *Managing Batch Control Utility*.

### Running Data Quality Batch

Data Quality (DQ) is a check that is done at every level based on the FSDM table. When data is moved from CSA to FSDM, a check is done on CSA. This check is done in order to move only useful data into the FSDM table. For example, a column in FSDM should not be blank if it is mandatory. These checks are also called rules.

The following data quality checks are done:

- **Length Validation Check:** If the length of data in source column is more than the length of target column, then an error message is generated. For example, if the `ACCT_INTRL_ID` column, which has a column length of 50 characters, needs to be populated from the source table column `V_ACCOUNT_NUMBER`, which has a few data with length more than 50 characters. An error message is raised.
- **Domain Check:** If any data does not qualify for the domain values, then an error message is generated. For example, if the valid value that column `ADDR_USAGE_CD` accepts is one of `M|B|L|A|O|P|D|H|X|V`, but the source column `V_ADDRESS_PURPOSE_TYPE_IND` has additional values such as `E` or `C`. An error message is raised.
- **Mandatory Check:** If a column which must have a value for the record to be valid has a null value, then an error message is generated. For example, if the column `ADDR_STRT_LINE1_TX` needs to have a value for the

record to be valid and is mapped to the source table column `V_ADDRESS_LINE1`. If the column `ADDR_STRT_LINE1_TX` has a null value, an error message is raised.

- **Threshold Test:** If a target table column must have a value that is greater than 0 but has a value of 0, then an error message is generated. For example, if the target table column `LDGR_AM` must have a value that is greater than 0 but the source table column `N_LEDGER_BAL` has a value as 0 or null, an error message is raised.

---

**Note:** In addition to the above data checks, another data check is done for duplicate data during data loading through AAI T2T.

---

### *Executing Data Transformation using DT*

The Data Transformation (DT) functionality allows you to delete the existing data in the AAI. For more information, see *Adding Tasks to a BD Batch*.

### *Moving Data through T2T*

Data is exported or moved from the CSA to the FSDM using AAI T2T. For more information on moving data through T2T, see *Chapter 2, Section - Data Mapping* of the *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

For the table to be loaded, the list of T2Ts are in *Managing Data*.

### *Executing Jobs*

After the data quality check, data transformation, and data movement through T2T is completed, execute the following jobs:

- Transformation jobs. For more information, see *Derived Datamap Types*.
- Scenario jobs. For more information, see *Managing Scenario Migration Utility*.
- Scenario post-processing jobs. For more information, see *Post-Processing Tasks*.

### *Ending the Batch*

To end the batch, run the `end_mantas_batch.sh` script. For more information, see *Managing Batch Control Utility*.

### **Using Datamaps**

The datamap takes the data from the CSA, enhances it, and then loads it into a target database table (FSDM). The Data Interface Specification (DIS) datamaps are used to load client-provided data, either through DIS files as specified in the DIS or through CSA tables.

---

**Note:** All the DIS datamaps in the Flat File Interface for which staging representation is marked as *Yes* are applicable for CSA loading. For more information, see *Behavior Detection Flat File Interface*.

---

To load data in the FSDM using flat files, follow these steps:

1. Configure the `DIS.source` parameter to FSDW. For more information on configuring other parameters, see *Behavior Detection Flat File Interface*.
2. Execute the Account datamap which loads data into the Account (ACCT) table using the following sample script:

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

The above step can be repeated for all datamaps for which staging representation is marked as *Yes*.

**Note:** If there are any errors or rejections in loading data, refer to the <OFSAAI Installed Directory>/bdf/logs path to know about the errors in the log file.

## Ingestion Flat File Data Load

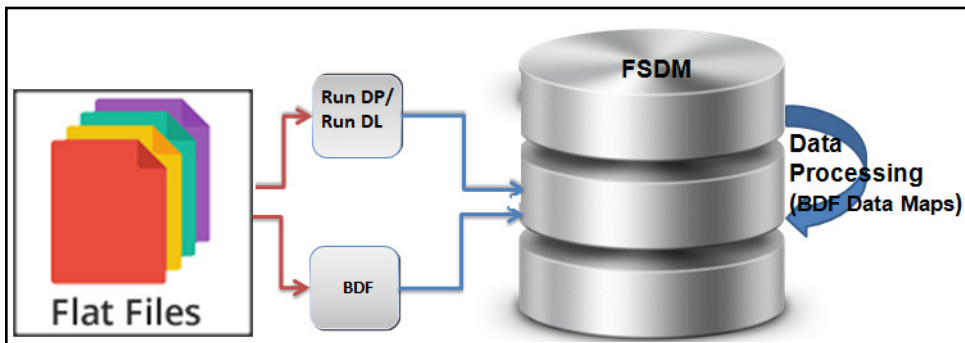
The loading process receives, transforms, and loads Market, Business, and Reference data that event detection and assessment investigation processing requires. After loading the base tables, the Oracle client's job scheduling system invokes datamaps to derive and aggregate data.

This section covers the following topics:

- [Overview](#)
- [Using Behavior Detection Datamaps](#) (Known as BD datamaps)
- [Using Pre-processing and Loading](#) (Known as runDP- runDL)

### Overview

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



**Figure 9. 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 [Behavior Detection Flat File Interface](#).

### Using Behavior Detection Datamaps

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

To load data in the FSDM 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. For more information on configuring other parameters, see [Appendix D, Managing Data](#).
  - 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.

---

### Using Pre-processing and Loading

The pre-processor component (runDP) use XML configuration files in the `/config/datamaps` directory to verify that the format of the incoming Oracle client data is correct and validate its content, specifically:

- Error-checking of input data
- Assigning sequence IDs to records
- Resolving cross-references to reference data
- Checking for missing records
- Flagging data for insertion or update

The loader component (runDL) receive pre-processed Reference data and business data. The components then load this data into the database.

---

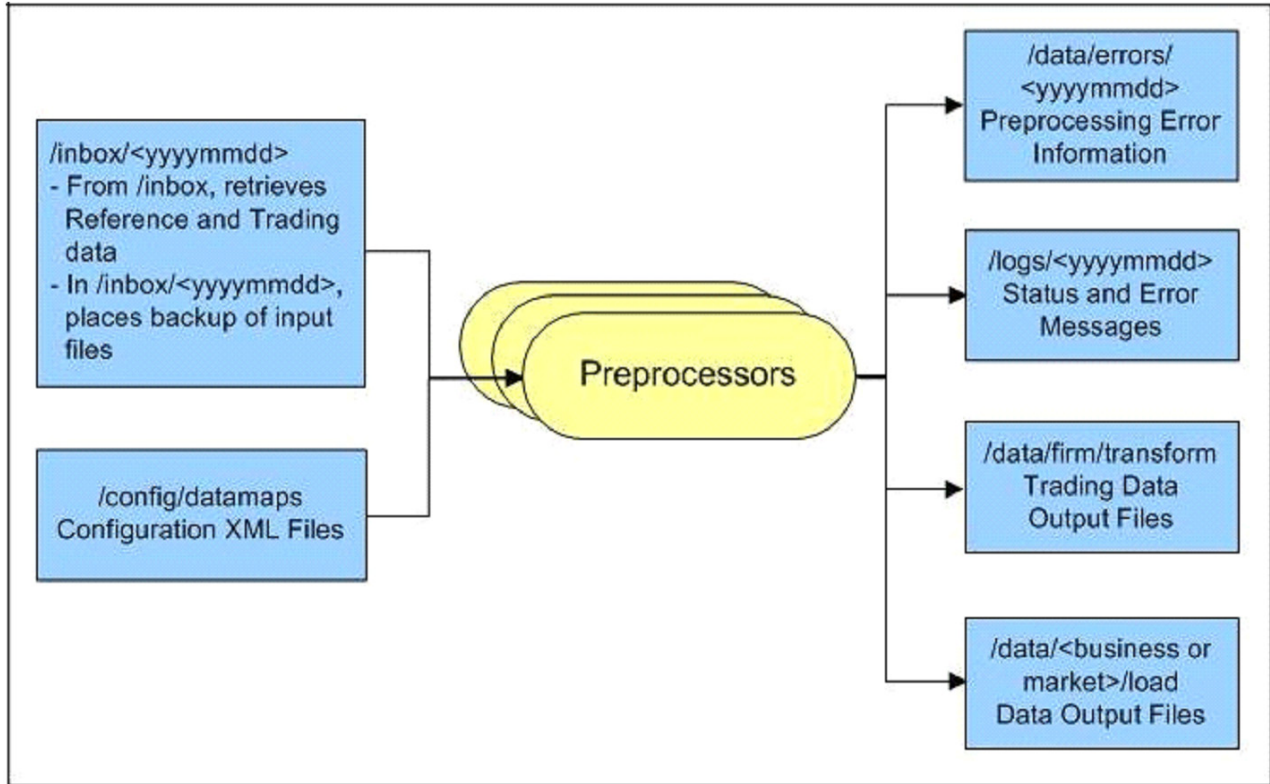
**Note:** The Pre-processor addresses only those files that match naming conventions that the DIS describes, and which have the date and batch name portions of the file names that match the current data processing date and batch. Oracle clients must only supply file types required by the solution sets on their implementation.

---

To load data in the FSDM using Pre-processing and Loading, follow these steps:

1. Place the ASCII.dat flat files in the `<OFSAAI Installed Directory>/ingestion_manager/inbox` directory. The component then performs data validation and prepares the data for further processing.
2. Execute runDP and runDL using the following sample scripts:
  - For runDP: `<OFSAAI Installed Directory>/ingestion_manager/scripts/runDP.sh AccessEvents`
  - For runDL: `<OFSAAI Installed Directory>/ingestion_manager/scripts/runDL.sh AccessEvents`

Pre-processors place output files in the directories that *Table 16* lists. The following figure summarizes Pre-processing input and output directories.



**Figure 10. Input and Output Directories**

The following figure illustrates the Trading Compliance Solution data loading process.

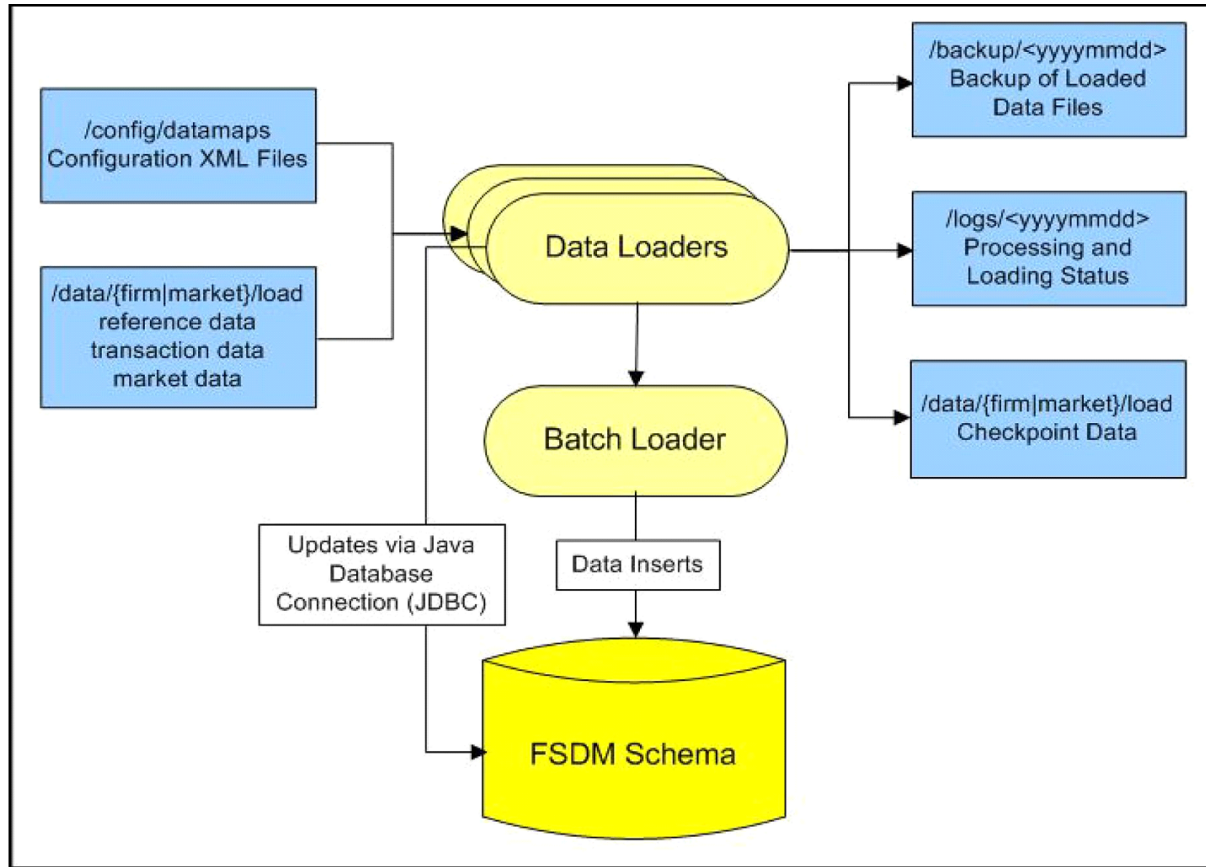


Figure 11. TCS Data Loading Process

**Note:** For more information on the directory structure, see *Appendix D, Managing Data*.

### Configuring RunDP/RunDL

For flat files, TBAML receives firm data in ASCII .dat flat files, which an Oracle client's data extraction process places in the /inbox directory.

### 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.

### *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.

### *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, suppose 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:** The time taken to do an incremental data load is more than for a full refresh data load, although there is no need to give complete data every time. Only updated or new data is required.

---

## **Encrypting Data Files**

To minimize exposure of data or personal information to users with access to the server, Oracle clients can encrypt ingestion files using a simple encryption technique which requires a generic 16 digit encryption key, combination of numerals and alphabets, such as: `AmritaP123456789`

Standard "AES" key spec and transformation "AES/ECB/PKCS5Padding" are used for encryption and decryption. Client can encrypt files using these on their own.

To run data ingestion on encrypted files, follow these steps:

1. Encrypt the ingestion files by running `encryptFileUtil.sh`, as shown below:

```
encryptFileUtil.sh <ALG_FILE_PWD> false <absolute_path_to_the_ingestion_
files_you_want_to_encrypt>
```

For example:

```
encryptFileUtil.sh AmritaP123456789 false
/scratch/ofsaaweb/BD806A/bdf/inbox/Account_20151209_DLY_01.dat
```

2. Update the BDF.Encryption.Password parameter in the bdf.xml file in <FIC\_HOME>/config/install path with the encryption key as shown below:

```
<Parameter name="BDF.Encryption.Password" type="STRING" value="<Encryption Key>"/>
```

3. Update the BDF.Encryption.Enable parameter in the bdf.xml file in <FIC\_HOME>/config/install path with the encryption key as shown below:

```
<Parameter name="BDF.Encryption.Enable" type="STRING" value="true"/>
```

4. Run execute.sh to invoke file ingestion.

## ***Managing Data Processing***

This section explains the concept of data processing and various methods of data processing.

This section covers the following topics:

- [Generating Change Logs with T2T](#)
- [Generating Change Logs with Hive](#)
- [Generating Change Logs](#)
- [Processing Data](#)
- [Processing Data Using FDT and MDT](#)

The following tables are currently supported for change log functionality:

- Account
- AccountAddress
- AccountPhone
- AccountEmailAddress
- AccountToCustomer
- Customer
- CustomerAddress
- CustomerPhone
- CustomerEmailAddress
- AccountRestriction
- InsurancePolicyToCustomer
- EmployeeAddress
- SettlementInstruction

## Generating Change Logs with T2T

The change log captures data which is added, deleted, or modified in the FSDM table. Data is initially moved from the staging table to the FSDM table through T2T. Once this is done, any modifications made such as adding new data, changing the data, or deleting the existing data are recorded in the change log table.

For example:

- Records are moved from the staging area into the FSDM table through T2T on day 1.
- A row is deleted from the FSDM table.
- Records are moved from the staging area into the FSDM table through T2T on day 2.
- A row is inserted into the FSDM table.

Both the deleted and added rows are displayed in the change log table.

The above example mentions two scenarios for displaying data in the change log table: a deleted value and an inserted value. A third scenario is if a value is changed in the column of a table. In this case, the old and new values are both captured in the change log.

To generate the change log, you must provide CHGLOG\_CAPTURE as the DT name in the Rule Name field and the name of the table that you want captured in the change log in the Parameter List field as mentioned in *Adding Tasks to a BD Batch*. Additionally, to delete a table, the DT name must be TRUNC\_FSDM\_TBL.

---

**Note:** The change log can only be derived from the second day onwards. Since change log functionality derives changes by comparing the data of two days, the first day data acts as a reference against which the second day data is compared and changes are derived.

---

## Components of the Change Log

The following table describes the functions of the different components in the change log:

**Table 8. Change Log Components**

Component	Description
Wrapper	The Wrapper contains logic that is required to record all changes (inserted, updated, and deleted data) in the log table.
Metadata table	The metadata table contains the metadata to support the wrapper.
FSDM table	The table in which the data is added, deleted, or modified.
AAI	AAI creates the DT which helps to execute the wrapper and run the T2T.
Change log table	The change log table captures the change log data after the wrapper is executed.

For the Metadata table component, the following values are available as metadata and can be captured in the change log:

**Table 9. T2T Change Log Metadata Table Component Values**

Value	Description
Table Name	This value is the name of the FSDM table in which a change has been made.
Table's Column Name	This value is the name of the FSDM column in the table in which a change has been made.

**Table 9. T2T Change Log Metadata Table Component Values**

Value	Description
User defined primary key	This value is the primary key of the column in which a change has been made.
Source Table Name	This value is the name of the source table in which a change has been made.
Source Table's Column name	This value is the name of the column in the source table in which a change has been made.
Table Enable Flag	This value must be changed to N if you do not want to capture the changes made in a table in the change log. The default value is Y.
Table's Column Enable Flag	This value must be changed to N if you do not want to capture the changes made in the column of a table in the change log. The default value is Y.
Customer Notification Suppress Flag	This value indicates whether the customer is notified of the change through email or not.
Additional Columns	This value is used to mention any additional columns of the table whose changes must be captured in the change log.

For the Change log table component, the following values are available:

**Table 10. T2T Change Log Table Component Values**

Value	Description
Table Name	This value is the name of the FSDM table in which a change has been made.
Column Name	This value is the name of the FSDM column in the table in which a change has been made.
Source File Name	This value is the name of the source extract file from which the field with the changed value was loaded.
Source Field Name	This value is the name of the field with the changed value in the source extract file where the value is changed.
Format	This value is a textual representation of the column or field format or data type.
Change Entry User Identifier	This value is the identifier of the person who entered the change. This is a unique identifier for the user who makes the change.
Change Entry User System Logon Identifier	This value is the user name of the user who makes the change.
Change Date	This value is the date on which the change was made.
Change Time	This value is the time at which the change was made.
Old Value	This value is the old value which was assigned to the specified table column.
New Value	This value is the new value which is assigned to the specified table column.
Key 1	This value is the textual representation of the value associated with the first column in the Primary Key or the user-defined primary key of the table containing the changed record.
Key 2	This value is the textual representation of the value associated with the second column in the Primary Key or the user-defined primary key of the table containing the changed record.
Key 3	This value is the textual representation of the value associated with the third column in the Primary Key or the user-defined primary key of the table containing the changed record.
Key 4	This value is the textual representation of the value associated with the fourth column in the Primary Key or the user-defined primary key of the table containing the changed record.
Change Type	This value is the code that indicates whether the change is an insert (add), a delete (removal), or an update.

Table 10. T2T Change Log Table Component Values

Value	Description
Customer Notification Suppression Indicator	This value indicates whether the customer is notified of the change through email or not.
Source System	This value is the source system from which this data content is extracted.

## Generating Change Logs

Change log and Change log summary records will be generated through BD.

When loading referential DIS files that are defined as Overwrite, it is possible to generate Change Log records which signify when certain fields associated with a reference data entity have changed. This is done by comparing the contents of the DIS file with the current contents of the associated database table. For performance reasons, this change log processing can be done when external tables are used to load the DIS files, so it is a requirement that `DIS.Source=FILE-EXT`. This requires an external directory, which is created during installation. In order to give access to an oracle user, place the .dat files in the external directory.

The change log records can also be derived with `DIS.Source = 'FSDW'` (CSA Ingestion). While `FILE_EXT` derives the change log based on comparison of reference data with newly ingested modified data (through the `DAT FILE`) on the next day, with the `DIS.Source=FSDW`, the change log is derived on comparing the reference data which is loaded to FSDM tables from staging table data.

---

**Note:** To derive the change log records the change log parameters in `<OFSAAI Installed Directory>/BDF/config/BDF.xml` should be uncommented.

---

Change log records can be generated in the following ways:

- Compare fields on a single reference data record that can be identified by a primary key. For example, an Account record can be identified by an Account Identifier. When an Account file is ingested, the Primary Customer Identifier on Account XYZ is compared to the Primary Customer Identifier currently in the database for Account XYZ. If they are different, then a Change Log record is created. This process only accounts for updates to already existing records. Change Log records are not created for new reference data records or deleted reference data records.
- Compare the set of values for a given field on several reference data records that map to a given key. For example, an Account Address record is identified with a combination of Account Identifier and Address Record Number. However, the information required is whether an Account Address record for a given Account has a field value that is different than any other Account Address record for that Account. For example, every Account Address record has a Country field. If there are two Account Address records for Account XYZ in the database with values for Country of US and CN, respectively. On the next day, an Account Address file is processed and there is an Account Address for Account XYZ with a value for Country of IR. A Change Log record is generated for the Country field of this Account Address record. Furthermore, in the case of Account Address, it is not just the Account Identifier of an Account Address record that is of interest. The Address Purpose is also of interest. So when we look in the database for Account Address records that match a given Account Address record in a DIS file, we look to match both the Account Identifier field and the Address Purpose field.



This processing is controlled by parameters in `<OFSAAI Installed Directory>/bdf/config/BDF.xml`. All of these parameters have been commented out, which means change log processing is turned off by default. To derive the change log records if `DIS.Source = 'FILE-EXT'`, the relevant parameters for the DIS files of interest should be copied to `<OFSAAI Installed Directory>/bdf/config/custom/BDF.xml` and uncommented.

**Table 11. Change Log Parameters**

Parameter	Description
<code>ChangeLog.&lt;DIS File Type&gt;.Fields</code>	The fields of this particular DIS file type which will be monitored for changes.
<code>ChangeLog.&lt;DIS File Type&gt;.IsSet</code>	Whether change log records are generated based on mechanism 1 above (false) or mechanism 2 (true). The default is false.
<code>ChangeLog.&lt;DIS File Type&gt;.QueryKey</code>	This is only relevant when <code>IsSet=true</code> . This defines the key that is used to query for reference data records matching the given one. In the Account Address example given above, the value would be <code>AccountIdentifier,AddressPurpose</code> . If this parameter is not present, then the business key located in the given DIS file type's data map (for example <code>bdf/datamaps/AccountAddress.xml</code> ) is used.
<code>ChangeLog.&lt;DIS File Type&gt;.OutputKey</code>	This is only relevant when <code>IsSet=true</code> . This defines the set of fields that are mapped to the <code>Key1, Key2, Key3, and Key4</code> fields of a Change Log record. This can be different from the <code>QueryKey</code> and business key in order to match what is expected in Change Log DIS file records, and also to support the Change Log Summary data maps. If this parameter is not present, then the business key located in the given DIS file type's data map (for example, <code>bdf/datamaps/AccountAddress.xml</code> ) is used.

To turn on Change Log processing for a given DIS file type, all the parameters for that file type must be uncommented. The values of the `ChangeLog.<DIS File Type>.Fields` parameter are preset based on the needs of the KYC application. If different fields are required, then this parameter should be changed. It is not necessary to change any of the other parameters.

For Example: If Address Street line fields are to be considered for change log generation, then the `ChangeLog.<DIS File Type>.Fields` parameter should be changed for that particular table as shown below.

```
<Parameter name="ChangeLog.AccountAddress.Fields" type="STRING"
value="Country,Region,State,City,PostalCode,MailHandlingInstruction" list="true"/>
```

should be changed to

```
<Parameter name="ChangeLog.AccountAddress.Fields" type="STRING"
value="Country,Region,State,City,PostalCode,MailHandlingInstruction,StreetLine1,StreetLine2,
StreetLine3,StreetLine4,StreetLine5,StreetLine6" list="true"/>
```

As in the example above, `StreetLine1,StreetLine2,StreetLine3,StreetLine4,StreetLine5` and `StreetLine6` will also be considered for change log generation. Similar steps can be followed for other change log related tables well.

Change Log records are written to the `CHG_LOG` table as the DIS file is being loaded. There are no additional scripts to be run. As soon as the parameters are uncommented, Change Log records are generated the next time DIS files are loaded.

## Processing Data

This section covers the following topics:

- [About BD Datamaps](#)
- [Derived Datamap Types](#)
- [Datamap Categories](#)
- [Processing Datamaps](#)
- [Configuring Risk Zones](#)
- [Customizing Review Reason Text](#)
- [DataMaps](#)

### **About BD 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.

BD datamaps can perform the following activities:

- Update summaries of trading, transaction, and instruction 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](#)

### **Derived Datamap Types**

The Oracle solution implemented determines the required datamaps, or a subset thereof:

- [AML Brokerage Datamaps](#)
- [AML Banking Datamaps](#)
- [Broker Compliance Datamaps](#)
- [Fraud Detection Datamaps](#)
- [Insurance Datamaps](#)
- [Trade Finance Datamaps](#)
- [Market Derived Datamaps](#)

---

**Caution:** If you are running multiple solutions, you must perform table comparisons to avoid running duplicate datamaps.

---

The following table describes the columns in the datamap tables that each section provides.

**Table 12. 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.

## 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.

---

## Processing Datamaps

This section provides the required datamaps for deriving and aggregating data based on the solution. Discussions of the datamaps appear in the order that processing must execute them during data loading, and include tables that describe each datamap. Datamap numbers that the accompanying tables provide also reflect this order.

Where predecessors exist, processing of datamaps cannot begin until completion of *predecessor* datamaps. These dependencies, or predecessors, may be internal to the datamap type, or external to the datamap type such as Summary datamaps dependent on watch list datamaps.

---

**Note:** If there is any performance issue with the running sequence of datamaps, it can be re-arranged. However, the predecessor for the datamap must be completed before running the datamap.

---

**Example:** The following is the order for the datamap to run:

```
FrontOfficeTransactionParty_InstnSeqID
FrontOfficeTransactionParty_HoldingInstnSeqID
```

If there is any performance issue with the datamap `FrontOfficeTransactionParty_HoldingInstnSeqID`, the datamap position can be rearranged in the batch script. Since there is the possibility that the previous process (`FrontOfficeTransactionParty_InstnSeqID`) is still running, the current datamap is waiting for the resources to be released.

### Example for Internal Dependency

For example, processing can run the `FrontOfficeTransactionParty_InstnSeqID` datamap immediately after completion of `FinancialInstitution_FOTPSPopulation` and `AccountToClientBank_FOTPSInstitutionInsert`.

### **Example for External Dependency**

Processing cannot run the AccountProfile\_Trade datamap until and unless the FrontOfficeTransactionPartyRiskStage\_EntityActivityRiskInsert datamap is run.

### **AML Brokerage Datamaps**

The following sections describe the Datamaps that are required for deriving and aggregating data for the AML Brokerage solution:

- [AML Brokerage - Pre-Watch List Datamaps](#)
- [AML Brokerage - Watch List Datamaps](#)
- [AML Brokerage - Post-Watch List Datamaps](#)
- [AML Brokerage - Summary Datamaps](#)
- [AML Brokerage - Balances and Positions Datamaps](#)

Each section provides a table that illustrates the datamaps and order of each datamap. This table describes the process by datamap number, datamap name, and internal or external predecessors, if any.

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.

### **DataMaps**

This section displays the different BD datamap types and covers the following topics:

- [AML Banking Datamaps](#)
- [Fraud Detection Datamaps](#)
- [Insurance Datamaps](#)
- [Trade Finance Datamaps](#)
- [Trusted Pair](#)

### **AML Banking Datamaps**

The following sections describe the required datamaps for deriving and aggregating data for the AML Banking solution:

- [AML Banking - Pre-Watch List Datamaps](#)
- [AML Banking - Watch List Datamaps](#)
- [AML Banking - Post-Watch List Datamaps](#)
- [AML Banking - Summary Datamaps](#)

### **Trade Finance Datamaps**

The following sections describe the datamaps that are required for deriving and aggregating data for the Trade Finance Solution:

- [Trade Finance - Pre-Watch List Datamaps](#)

- Trade Finance- Post-Watch List Datamaps

## Managing Data For BD Applications

This section explains different methods used to load and process data in various BD applications. Figure 12 shows the sequence for data loading:

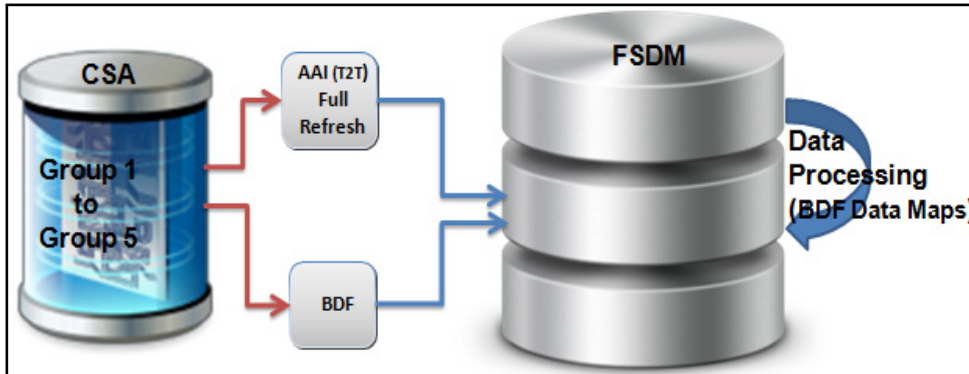


Figure 12. Data Loading For TBAML Application

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

Table 13. Managing Application Data

Steps	Group
<ol style="list-style-type: none"> <li>1. Execute Group 1 through Group 5 in sequence in the CSA using AAI T2T/H2T. For more information, see <i>Loading T2T using the AAI Framework</i>. For more information on the interface files available in Group 1 to Group 5, see <i>Behavior Detection Flat File Interface</i>.</li> <li>2. Process the loaded data using BD datamaps in FSDM. For more information, see <i>Managing Data Processing</i>.</li> <li>3. Interface files in the same group loaded through different loading method can be executed in parallel.</li> <li>4. Run AML BD transformation. For more information on the AML datamaps, see <i>AML Brokerage Datamaps</i> and <i>AML Banking Datamaps</i>.</li> <li>5. For network scenarios, refresh the temporary tables.</li> </ol>	Group 1 Group 2 Group 3 Group 4 Group 5
<ol style="list-style-type: none"> <li>1. Execute Group 7 using FDT/MDT. For more information, see <i>Table 77</i>.</li> <li>2. Process the derived datamaps for Trade Finance. For more information, see <i>Trade Finance Datamaps</i>.</li> </ol>	Group 7

## Post Load Changes

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



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 (,).

### ***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 [Table 14](#):

**Table 14. OFSBD Job Protocol Shell Scripts**

OFSBD Job Protocol Process Shell Script	Description
<code>start_mantas.sh</code>	Starts all OFSBD jobs. This script invokes the <b>cloner</b> and MANTAS processes. This is the integration point for a third-party scheduling tool such as Maestro or AutoSys.
<code>start_chkdisp.sh</code>	Calls on the <code>check_dispatcher.sh</code> script to ensure that the <code>dispatcher</code> runs.
<code>stop_chkdisp.sh</code>	Stops the <code>dispatcher</code> process.
<code>restart_mantas.sh</code>	Changes job status codes from the ERR status to the RES status so that the <code>dispatcher</code> can pick up the jobs with the RES status.
<code>recover_mantas.sh</code>	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 <code>restart_mantas.sh</code> 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

## 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 15. KDD\_JOB\_TEMPLATE with Sample Job Template Group**

JOB_ID	TEMPLATE_GROUP_ID
37	1
41	1

## 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.

## 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.



## 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.

## Configuring the Dataset Override Feature

To configure a dataset override, follow these steps:

1. 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.

---

2. 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`.

## 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*

## 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.

### 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 16. 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 17. Database Environment Variables in system.env File**

Variable	Environment	Description
ORACLE_HOME	Oracle	Identifies the base directory for the Oracle binaries. You must include: <ul style="list-style-type: none"><li>• <code>\$ORACLE_HOME</code> and <code>\$ORACLE_HOME/bin</code> in the <code>PATH</code> environment variable value.</li><li>• <code>\$ORACLE_HOME/lib</code> in the <code>LD_LIBRARY_PATH</code> environment variable value.</li></ul>
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 <code>SQL*NET</code> .

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

**Table 18. Operating System Environment Variables in system.env File**

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

## Starting the Dispatcher

Although multiple jobs and MANTAS instances can run concurrently in OFSBD, 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 dispatchers 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.

## 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.

---

**Caution:** 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.

## Monitoring the Dispatcher

The `install.cfg` file that was set up during server installation contains the `kdd.dispatcher.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.

---

**Caution:** The `dispatch.log` file can be a lengthy printout.

---

## 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:

- [Understanding the Job Status Codes](#)
- [Starting Behavior Detection Jobs](#)
- [Starting Jobs Without the Dispatcher](#)
- [Restarting a Job](#)
- [Restarting Jobs Without the Dispatcher](#)
- [Stopping Jobs](#)
- [Monitoring and Diagnosing Jobs](#)

## 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.

## 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* on page 34 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.

## 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.

## 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* on page 39 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>`.

## 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.

## 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.

## 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.

---

**Caution:** This job log file may be a lengthy printout.

---

## ***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.

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

---

**Caution:** 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](#)

### **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.

## 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.

## 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.

## 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 AAI) Applications Pack Installation and Configuration Guide*.

This section includes the following topics:

- Adding Behavior Detection Batches
- Adding Tasks to a BD 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.

## **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**.

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

**Figure 13. Batch Maintenance Page**

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

**Figure 14. Add Batch Definition page**

- Enter the batch details as described in the following table:

**Table 19. 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.

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

## 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`

- `runDP.sh`
- `runDL.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 *Post Load Changes* in the *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

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

- `runDP.sh`
- `runDL.sh`
- `execute.sh`

## **Adding Tasks to a BD 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**.

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

The screenshot shows the 'Batch Maintenance' page with the following details:

- Search filters: Batch ID Like (BDINFO806), Batch Description Like, Module (dropdown), Last Modification Date (Between/And with calendar icons).
- Batch Name section: + Add, View, Edit, Delete icons.
- Table of Batches:
 

Batch ID	Batch Description	Batch Edit/Non Edit
<input type="checkbox"/> BDINFO806_BATCH1	AM_GDPR	E
<input type="checkbox"/> BDINFO806_BATCH2	AM_GDPR	E
<input type="checkbox"/> BDINFO806_BATCH3	AM_GDPR	E
- Page 1 of 1 (1-3 of 3 items), Records Per Page 15.
- Task Details section:
 

Task ID	Task Description	Metadata Value	Component ID	Precedence
Page 0 of 0 (0-0 of 0 items)				

Figure 15. 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 *Operation* chapter of the *Oracle Financial Services Advanced Analytical Applications Infrastructure (OFSAAI) User Guide*.

## 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:

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

This screenshot is identical to Figure 15, showing the 'Batch Maintenance' page with search filters, a table of batches, and task details sections.

Figure 16. Batch Maintenance page

- 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.

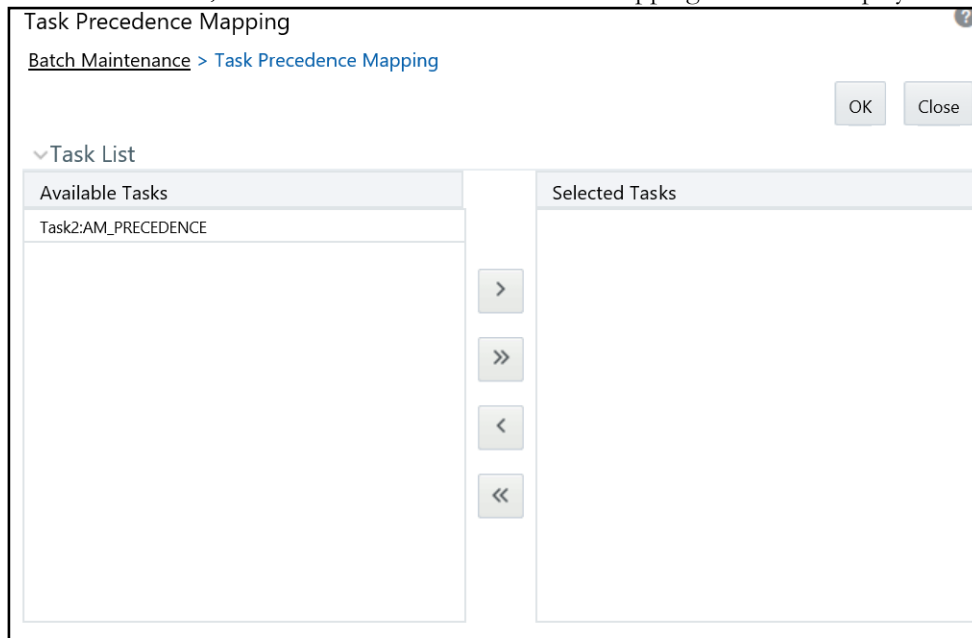


Figure 17. Task Precedence Mapping

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.

## Running a Single Task Using a Batch

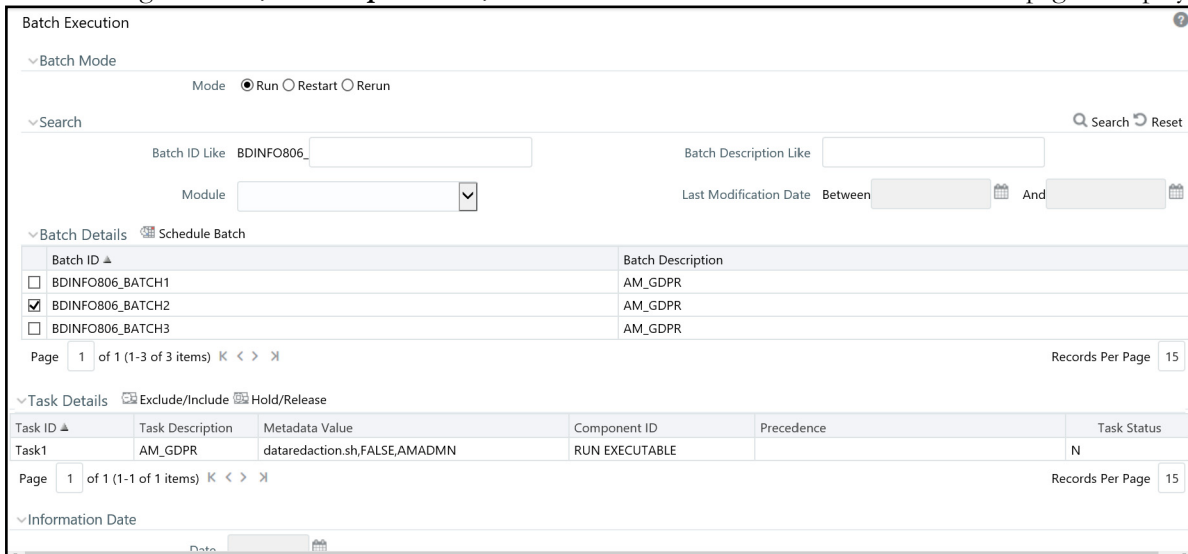
From the Batch Execution page, you can also run a single task from a batch.

**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:

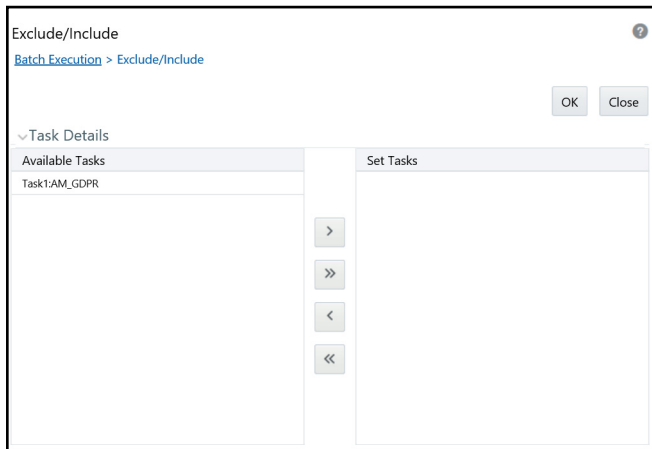
1. Login as the Administrator. The OFSAAI Applications page is displayed.
2. Click **Trade Based Anti Money Laundering**.

- In the Navigation List, select **Operations**, then **Batch Execution**. The Batch Execution page is displayed.



**Figure 18. Batch Execution page**

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



**Figure 19. Task Mapping Window**

- Retain the tasks that you want to execute under Available Tasks section and move the rest to the Set Tasks section.
- 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?*
- Click **OK**.
- Click **Execute Batch**.

## 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**.

The screenshot shows the 'Batch Scheduler' interface. At the top, there are search filters for 'Batch ID Like' (BDINFO806), 'Batch Description Like', 'Module', and 'Last Modification Date'. Below this is a 'Server Time' section showing 'Current Server Time: 15/05/2018 15:18:42'. A table lists three batches: BDINFO806\_BATCH1 (checked), BDINFO806\_BATCH2, and BDINFO806\_BATCH3, all with 'AM\_GDPR' descriptions. Below the table, the 'Batch Scheduler' section shows 'Domain: BDINFO806' and 'Batch: BDINFO806\_BATCH1'. The 'Schedule' section has 'New Schedule' selected. The 'New Schedule' section has 'Schedule Name' and frequency options: 'Once' (selected), 'Daily', 'Weekly', 'Monthly', and 'Adhoc'. The 'Schedule Time' section has 'Start Date', 'End Date', 'Run Time' (00Hours, 00Minutes), and 'Lag' (0Days). 'Save' and 'Cancel' buttons are at the bottom.

Figure 20. Scheduling a Batch Once

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

## 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.

The screenshot shows the 'Batch Scheduler' interface. At the top, there are search and filter fields for 'Batch ID Like' (containing 'BDINFO806'), 'Batch Description Like', 'Module', and 'Last Modification Date'. Below this is a 'Server Time' section showing 'Current Server Time: 15/05/2018 15:18:42'. A table lists three batches: 'BDINFO806\_BATCH1' (checked), 'BDINFO806\_BATCH2', and 'BDINFO806\_BATCH3', all with 'AM\_GDPR' descriptions. Below the table, the 'Batch Scheduler' section shows 'Domain: BDINFO806' and 'Batch: BDINFO806\_BATCH1'. The 'Schedule' section has 'New Schedule' selected. The 'New Schedule' section includes a 'Schedule Name' field, radio buttons for 'Once', 'Daily', 'Weekly', 'Monthly', and 'Adhoc', and a 'Schedule Time' section with 'Start Date', 'End Date', 'Run Time' (00 Hours, 00 Minutes), 'Every' (Days), and 'Lag' (0 Days) fields.

**Figure 21. Scheduling a Daily Batch**

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

## 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.

The screenshot shows the 'Batch Scheduler' interface. At the top, there are search and filter fields for 'Batch ID Like' (containing 'BDINFO806'), 'Batch Description Like', 'Module', and 'Last Modification Date'. Below this is a 'Server Time' section showing 'Current Server Time: 15/05/2018 15:18:42'. A table lists three batches: 'BDINFO806\_BATCH1' (checked), 'BDINFO806\_BATCH2', and 'BDINFO806\_BATCH3', all with 'AM\_GDPR' descriptions. Below the table, the 'Batch Scheduler' section shows 'Domain: BDINFO806' and 'Batch: BDINFO806\_BATCH1'. The 'Schedule' section has 'New Schedule' selected. Under 'New Schedule', there is a 'Schedule Name' field and radio buttons for 'Once', 'Daily', 'Weekly', 'Monthly', and 'Adhoc'. The 'Schedule Time' section includes 'Dates' (Start and End), 'Run Time' (00 Hours, 00 Minutes), 'Lag' (0 Days), 'Every' (Weeks), and 'Working days of the Week' (checkboxes for Sunday through Saturday). 'Save' and 'Cancel' buttons are at the bottom.

Figure 22. Scheduling a Weekly Batch

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

## 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.

The screenshot shows the 'Batch Scheduler' interface. At the top, there are search filters for 'Batch ID Like' (BDINFO806), 'Batch Description Like', 'Module', and 'Last Modification Date'. Below this is a 'Server Time' section showing 'Current Server Time: 15/05/2018 15:18:42'. A table lists three batches: 'BDINFO806\_BATCH1' (checked), 'BDINFO806\_BATCH2', and 'BDINFO806\_BATCH3', all with 'AM\_GDPR' descriptions. The 'Batch Scheduler' section shows 'Domain: BDINFO806' and 'Batch: BDINFO806\_BATCH1'. The 'New Schedule' section is active, with 'Schedule Name' empty. The frequency is set to 'Once'. The 'Schedule Time' section shows 'Start Date' and 'End Date' as empty date pickers, 'Run Time' as '00 Hours' and '00 Minutes', and 'Lag' as '0 Days'. The 'Interval Every' section is selected, with 'Month(s)' as the unit. Below this are checkboxes for months: Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec. The 'Dates' section is selected, with a field for 'of the month (comma delimited)' and an option to 'include month's last date'. The 'Occurrence' section is unselected, with a dropdown for 'of the weekday'. 'Save' and 'Cancel' buttons are at the bottom.

Figure 23. Configuring a Monthly Batch

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

## 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.

Batch Monitor

Batch ID Like: BDINFO806\_      Batch Description Like:      Search      Reset

Module:      Status:      Start Date:      End Date:

▼ Batch Details

Batch ID	Batch Description
<input checked="" type="checkbox"/> BDINFO806_BATCH1	AM_GDPR
<input type="checkbox"/> BDINFO806_BATCH2	AM_GDPR
<input type="checkbox"/> BDINFO806_BATCH3	AM_GDPR

Page 1 of 1 (1-3 of 3 items)      Records Per Page 15

▼ Batch Run Details      Start Monitoring      Stop Monitoring      Reset

Information Date:      Monitor Refresh Rate (seconds): 5      Batch Run ID:

▼ Batch Status

Batch Run ID	Batch Status
No data found	

▼ Task Details

Task ID	Task Description	Metadata Value	Component ID	Task Status	Task Log
No data found					

Page 0 of 0 (0-0 of 0 items)      Records Per Page 0

▼ Event Log

Message ID	Description	Severity	Time
No data found			

Page 0 of 0 (0-0 of 0 items)      Records Per Page 0

Figure 24. 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.

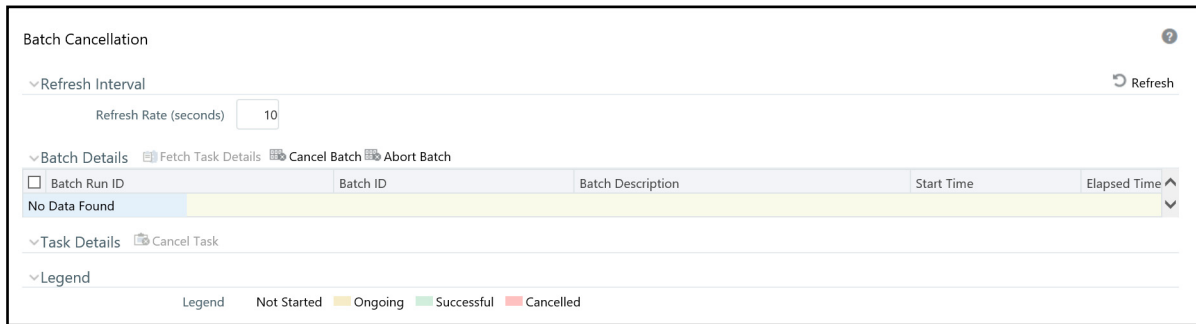
## Canceling a Batch After Execution

Cancellation of a batch cancels a current batch execution.

**Note:** 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 25. Batch Cancellation Page**

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

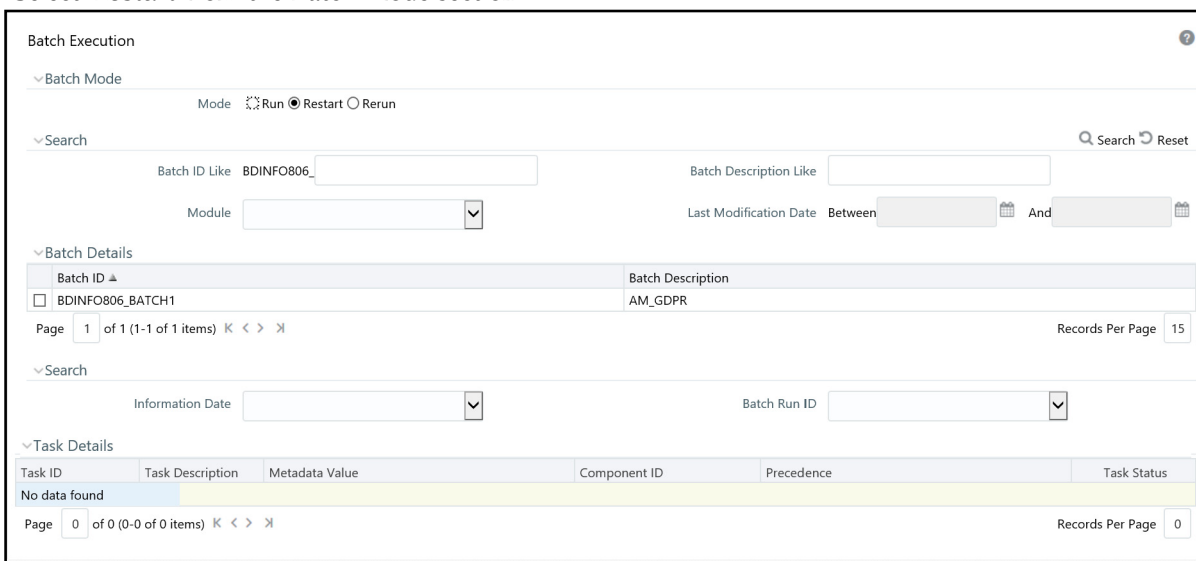
## Re-starting a Batch

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.

**Note:** 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 26. 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**.

## 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.

The screenshot shows the 'Batch Execution' page with the following sections:

- Batch Mode:** Mode selection with radio buttons for Run, Restart, and Rerun (selected).
- Search:** Fields for Batch ID Like (BDINFO806), Batch Description Like, Module, and Last Modification Date (Between/And).
- Batch Details:** A table with columns Batch ID and Batch Description. It lists three items: BDINFO806\_BATCH1, BDINFO806\_BATCH2, and BDINFO806\_BATCH3, all with description AM\_GDPR. Below the table is a pagination control showing 'Page 1 of 1 (1-3 of 3 items)' and 'Records Per Page 15'.
- Search:** Fields for Information Date and Batch Run ID.
- Task Details:** A table with columns Task ID, Task Description, Metadata Value, Component ID, Precedence, and Task Status. It displays 'No data found'. Below is a pagination control showing 'Page 0 of 0 (0-0 of 0 items)' and 'Records Per Page 0'.
- Execute Batch:** A button at the bottom center.

Figure 27. 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**.

## 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

The screenshot shows the 'Batch Processing Report' interface. At the top, there is a search bar with a dropdown for 'Information Date' set to 'Latest Batch Run' and a dropdown for 'Batch Status' set to 'ALL'. Below this, it displays the report for 'Thursday, August 16, 2018 2:10:52 PM EDT for Information domain: FCCMINFO'. There are two expandable sections for execution details. The first section is for 'Execution Date : 2018-08-10 12:46:17.0' and 'Batch Run ID : FCCMINFO\_1533919576825\_20180810\_1'. The second section is for 'Execution Date : 2018-08-10 12:31:43.0' and 'Batch Run ID : FCCMINFO\_1533918702875\_20180810\_1'. Below these are two tables of components.

Component	Task	Parameters	Status
RUN EXECUTABLE	Task1	Batch Parameter : Y Datastore Name : FCCMINFO Datastore Type : EDW Executable : "EDQCall.sh","watchlist-management.properties","Watchlist-Management","Download-Prepare-Filter-and-Export-All-Lists","Watchlist-Management" IP Address : whf00arl.in.oracle.com Optional Parameters : "\$RUNID=1533803759056,\$PHID=Watchlist_Management_process,\$XEID=1533918702875,\$RUNSK=4"	S
RUN EXECUTABLE	Task2	Batch Parameter : Y Datastore Name : FCCMINFO Datastore Type : EDW Executable : "EDQCall.sh","watchlist-management.properties","Watchlist-Management","Generate-StopPhrases","Watchlist-Management" IP Address : whf00arl.in.oracle.com Optional Parameters : "\$RUNID=1533803759056,\$PHID=Watchlist_Management_process,\$XEID=1533918702875,\$RUNSK=4"	S

**Figure 28. 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.

## 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

The screenshot shows the 'View Log' interface. At the top, there are search filters: 'Component Type' (Model Upload), 'Folder', 'User', 'As of Date', 'Task Name', and 'Batch Run ID'. Below the filters is a table titled 'Task ID Information (Click on the Task ID for More Information)'. The table has columns for Component, Task Name, Task ID, Status, Start Date, End Date, Elapsed Time, and User. There are two rows of data. At the bottom, there is a pagination bar showing 'Page 1 of 1 (1-2 of 2 items)' and a 'Records Per Page' dropdown set to 2.

Component	Task Name	Task ID	Status	Start Date	End Date	Elapsed Time	User
Model Upload	MODEL_CMD_EXECUTE_200001	200001	Success	08/10/2018 10:52:07	08/10/2018 10:58:47	00:06:40	sysadmn
Model Upload	MODEL_CMD_EXECUTE_200000	200000	Success	08/10/2018 10:46:54	08/10/2018 10:52:05	00:05:11	sysadmn

Figure 29. 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.

## 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.

### Starting a Batch Run

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

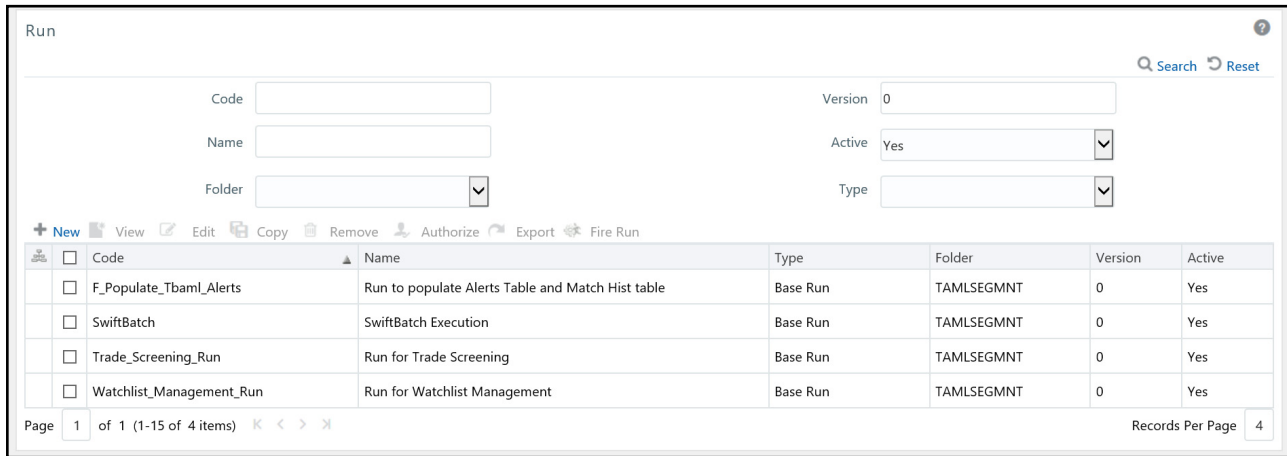
This section explains how to start the batch run.

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**.

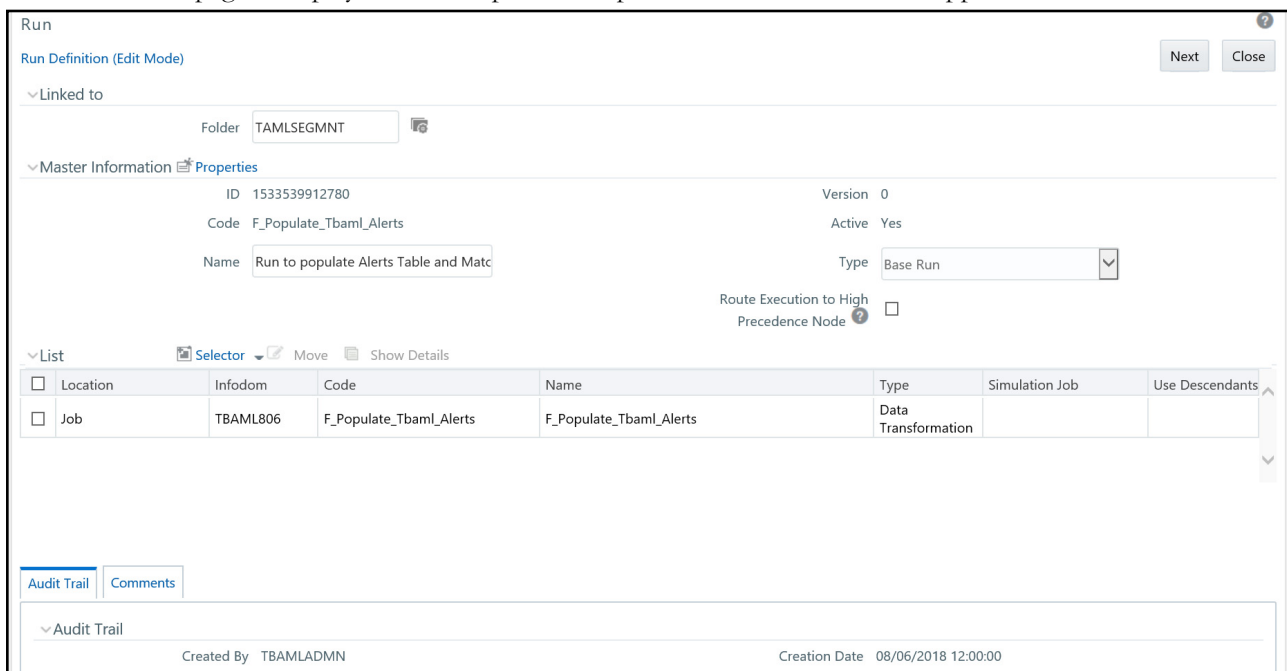


4. Click **Run**. The Run page is displayed with the available applications.



**Figure 30. Run page**

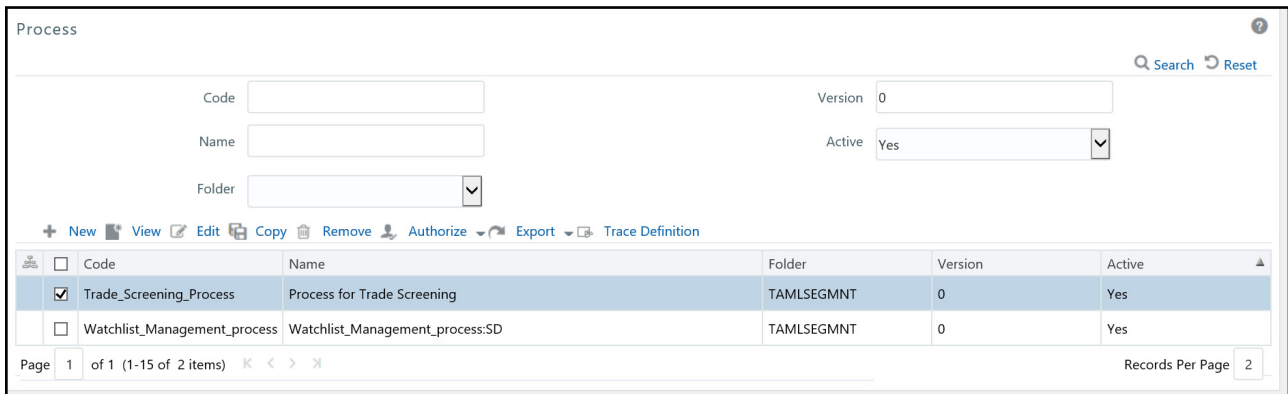
5. Select an application from the Code column (for example, F\_Populate\_Tbaml\_Alerts) and click **Edit**. The Run Definition page is displayed with the process or processes associated to this application.



**Figure 31. Run Definition page**

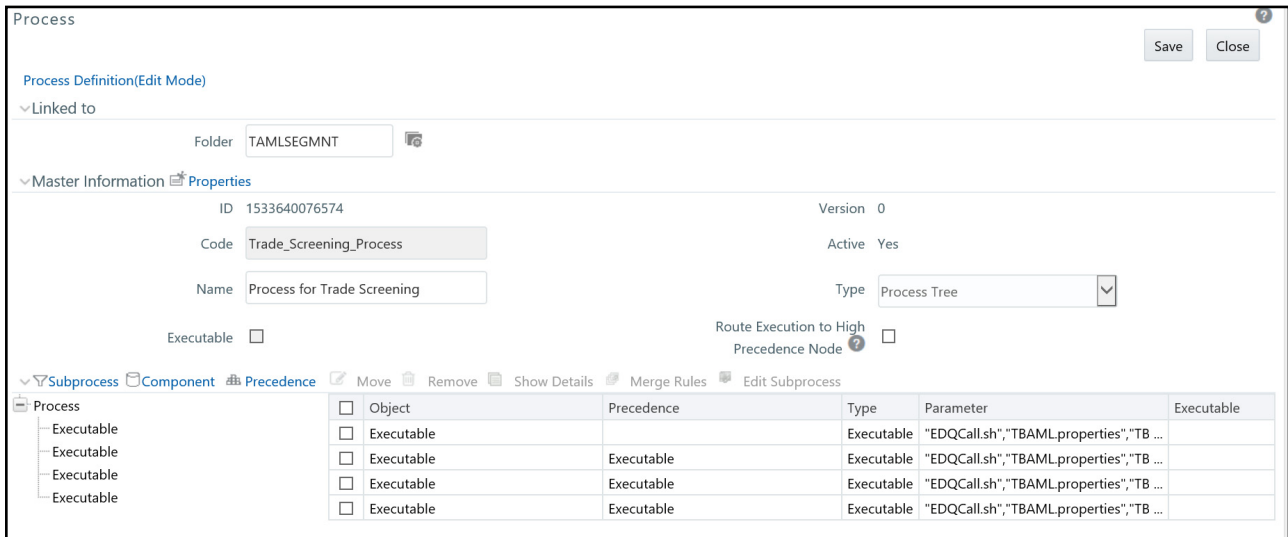
6. Select the job or jobs for the batch and click **Next**. The Detail section is populated. Complete your edits and click **Save**.

7. In the Navigation List, select **Process** to open the Process page.



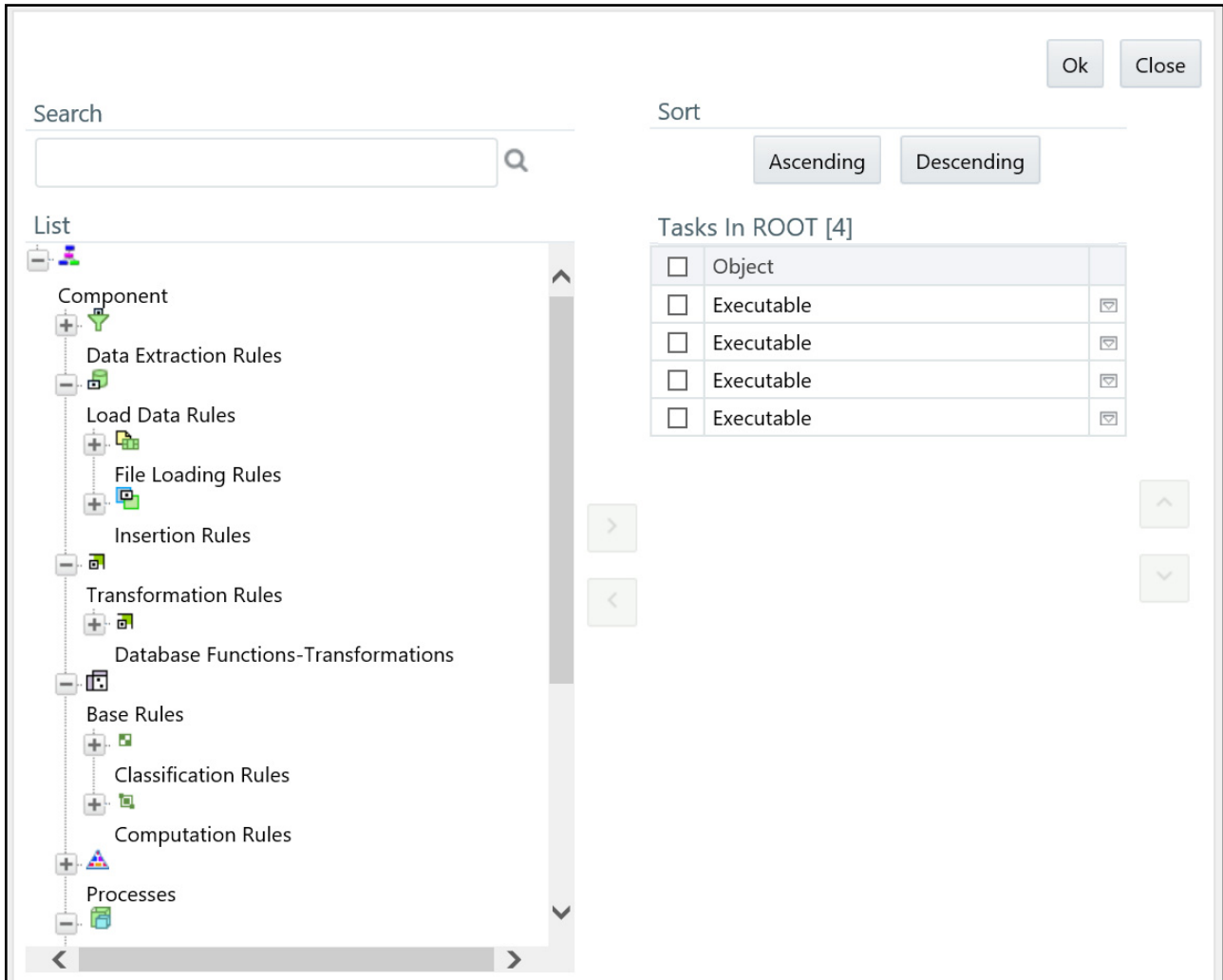
**Figure 32. Process page**

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 33. Process Definition page**

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



**Figure 34. 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"

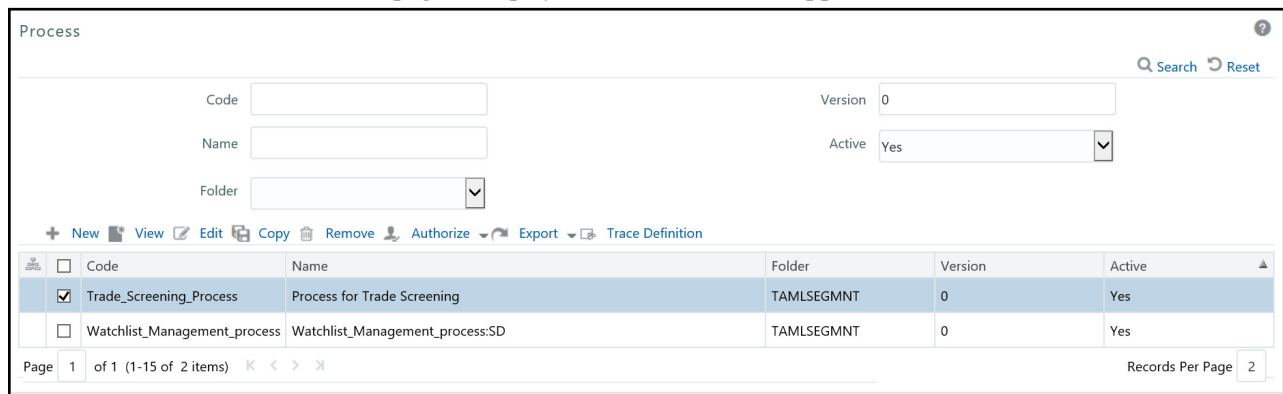
10. Modify the parameters and click **OK**.

## Ending a Batch Run

This section explains how to end the batch run.

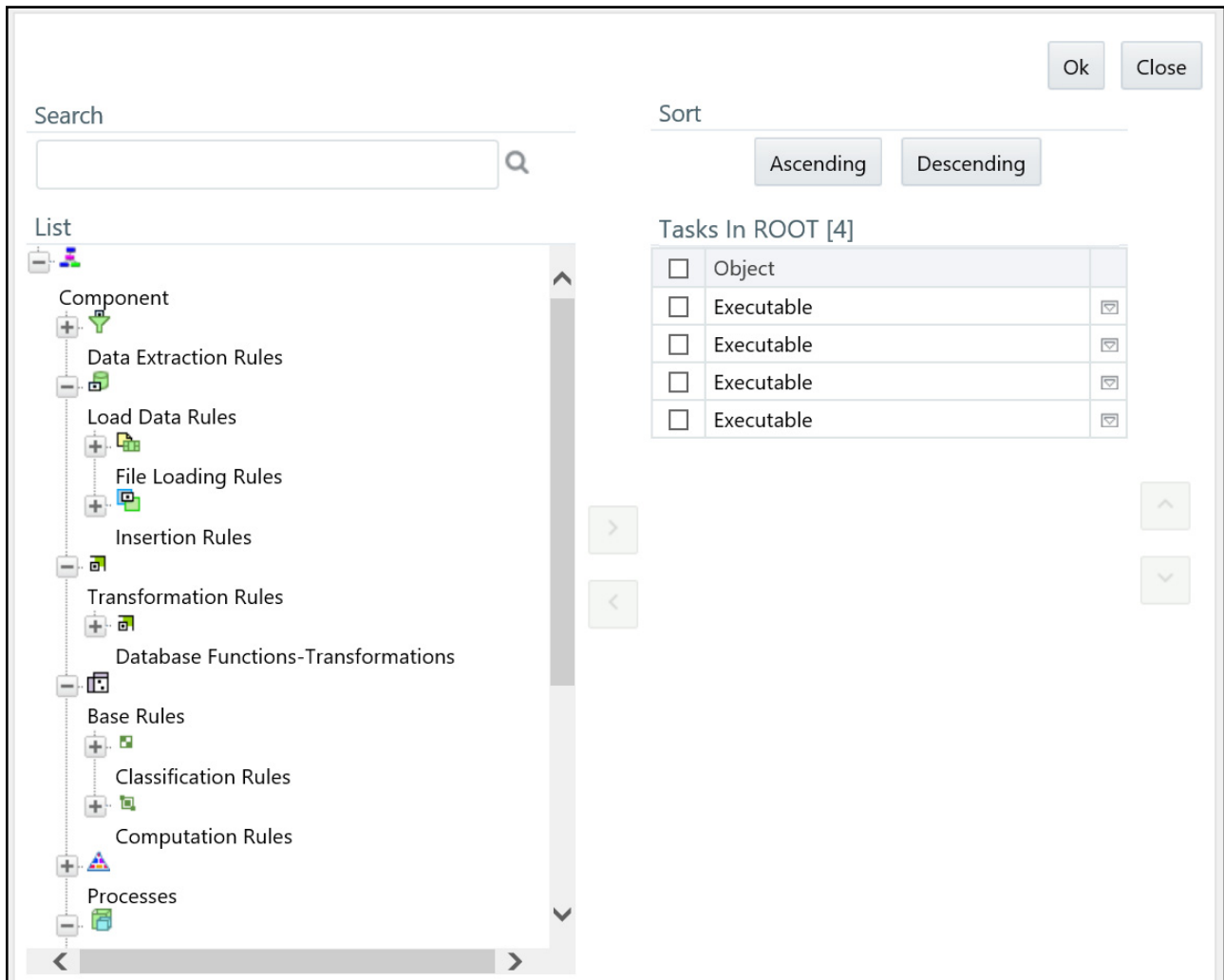
To end the batch run, follow these steps:

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 35. 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 36. 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.

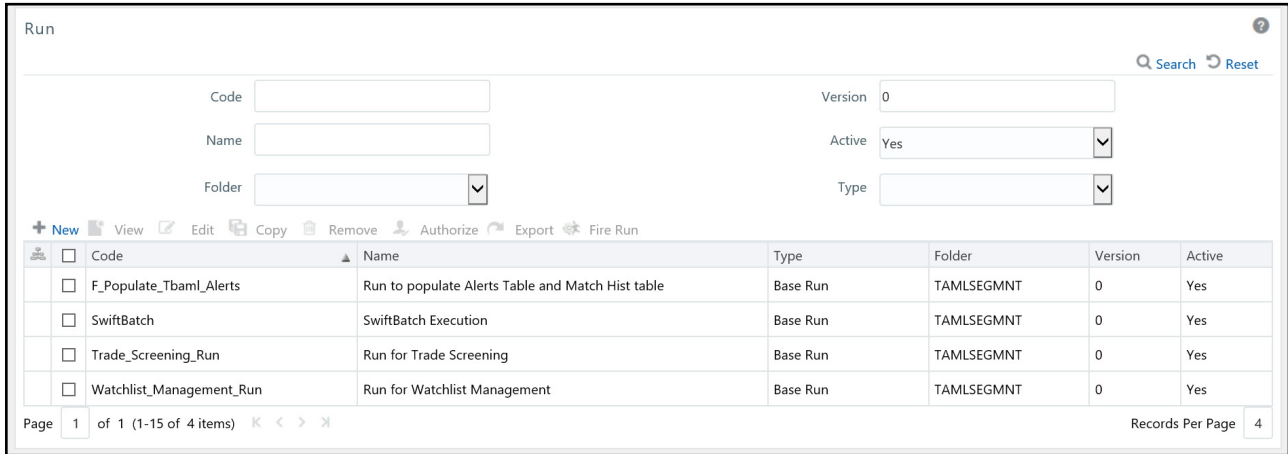
10. Modify the parameters and click **OK**.

## Executing a Batch Run

This section explains how to execute the batch run.

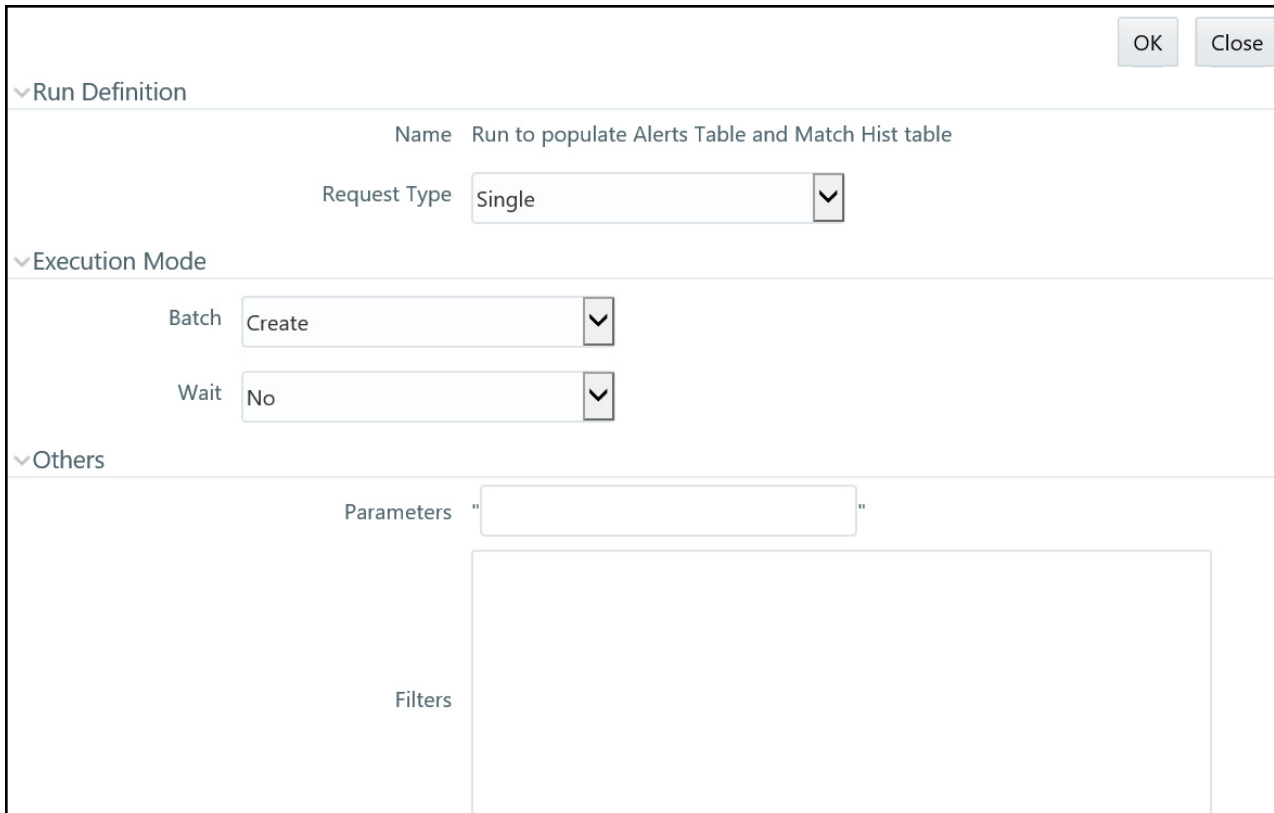
To access and execute the batch run, follow these steps:

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 37. Run page**

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



**Figure 38. Fire Run**

6. Enter the following details:

**Table 20. Adding Fire Run Details**

Field	Description
Request Type	Select Request Type based on the following options: <ul style="list-style-type: none"> <li>● Single: If the batch must be executed once.</li> <li>● Multiple: If the batch must be executed multiple times at different intervals.</li> </ul>
Batch	Select Batch. It has the following options: <ul style="list-style-type: none"> <li>● Create</li> <li>● Create &amp; Execute</li> </ul> From these options, select <b>Create &amp; Execute</b> .
Wait	Select Wait. It has the following options: <ul style="list-style-type: none"> <li>● Yes: This executes the batch after a certain duration. Enter the duration as required.</li> <li>● No: This executes the batch immediately.</li> </ul>
Parameters	Enter the parameters for this batch.
Filters	Enter the filter details. <b>Note:</b> \$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 <b>Note:</b> For \$MISDATE option: <ul style="list-style-type: none"> <li>● For either Date or Timestamp datatypes, to_date is mandatory for the filter.</li> <li>● Activity Table Name and Activity Column Name should be in capital.</li> </ul>

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*.





This chapter defines the following post-processing administrative tasks:

- [About Post-Processing](#)
- [Augmentation](#)
- [Match Scoring](#)
- [Alert Creation](#)
- [Alert Scoring](#)
- [Highlight Generation](#)
- [Historical Data Copy](#)
- [Alert Correlation](#)

## About Post-Processing

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

- **Augmentation:** Collects information for pattern detection, which enables proper display or analysis of these results may be required. This process is automatically executed at the end of each scenario run.
- **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).
- **Alert Correlation:** Uncovers relationships among events by correlating events to business entities and subsequently correlating events to each other based on these business entities (this latter step is optional). The relationships are discovered based on configurable rule sets (Refer to [Alert Correlation](#) for more information).

---

**Note:** You can re-run any failed post-processing job.

---

## Order of Running Post-Processing Administrative Tasks

Run the post-processing administrative tasks in this order:

1. Match Scoring (501)
2. Multi Match Alert Creation (502)
3. Single Match Alert Creation (503)
4. Alert Scoring (504)
5. Highlight Generation
6. Historical Data Copy
7. Alert Correlation (508)

---

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

---

## 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.

### 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 the 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.

## 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.

## 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 sections describe running each type of Alert Creator.

### To Run Multi-match Alert Creator

To run the multi-match Alert Creator, follow the steps:

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

```
start_mantas.sh 502
```

where 502 is the job template that Behavior Detection provides to run the Alert Creator algorithm.

### To Run Single Match Alert Creator

To run the single match Alert Creator, follow the 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.

## 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.

### 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.

### 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*.
- **BIND\_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 centrlicity names in the bindings string. This is the *catch all* algorithm that ensures that all matches that have a bound centrlicity 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 centrlicity in the first binding name in the binding string. The centrlicity name is used to determine the event's centrlicity ID.

## Alert Scoring

OFSBD 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.

## 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.

## Highlight Generation

The behavior detection subsystem displays event and match highlights in the OFSBD 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.

## Historical Data Copy

Behavior Detection 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
```

---

**Note:** 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 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 21. HDC Configurable Parameters**

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

By default, Behavior Detection writes log messages for this script in the `<OFSAAI_Installed_Directory>/database/db_tools/logs/hdc.log` file.

## **Alert Correlation**

OFSBD provides a mechanism to correlate events to business entities and optionally to each other based on configurable rule sets. This functionality is performed by the Alert Correlation process. Details on configuring the data paths to correlate events to business entities as well as information on constructing the rules to correlate events to each other is provided in the following sub-sections.

### **Running the Alert Correlation Job**

Alert Correlation is a part of the Behavior Detection subsystem. OFSBD delivers job template group 508 to run the Alert Correlation job (for information on how to run this process through a web service, refer to the *Oracle Financial Services Behavior Detection Framework Services Guide*).

To run an Alert Correlation job, follow the steps:

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

```
start_mantas.sh 508
```

where, 508 is the job template that OFSBD provides to run the Alert Correlation algorithm.

### **Understanding Alert Correlation Configuration**

As mentioned above, Alert Correlation performs two major tasks correlating events to business entities and correlating events to events. The second step is optional, and is governed by the `correlate_alerts_to_alerts` job parameter delivered with the template job associated to group 508. If this parameter's value is set to `true` then this step will be performed, and if this value is set to `false` then it will not be performed.

The other job parameter associated with Alert Correlation is `correlation_actions`. This parameter has a value of a comma-delimited list that defines what optional actions to take against a correlation that is found by the `correlate_alerts_to_alerts` task. The currently-supported actions are `prioritize`, which will assign a score to the correlation, and `promote_to_case`, which will promote a correlation to a case. Both actions have associated parameters that are defined and dictated by the rule that generated the correlation (these rule sets are discussed below). Note that the `promote_to_case` action is also a licensable feature (dependent on Enterprise Case Management license). The same information as above applies in terms of obtaining and configuring a license file.

Both parameters above can be configured by changing their associated `VALUE_TX` values in the `KDD_PARAM_BINDING` table.

**Note:** To assist with performance tuning, the *filter\_by\_batch* job parameter can optionally be added to the `KDD_PARAM_BINDING` table. A value of *true* causes a filter to be appended to all queries retrieving events/correlations/cases by the Alert Correlation algorithm based on the current batch name. A value of *false* (default behavior) does not include this filter. For example, if your organization varies batches by country, and only needs to pull in data for a specific country in each batch, turning this filter on prevents them from pulling in unnecessary data (that is, from other countries) in each batch. This parameter is added as a job parameter instead of the `install.cfg` parameter because the requirement to filter by batch may vary from job to job.

**Note:** Execute the following query in order to run the 508 Job for the *filter\_by\_batch* parameter:

```
insert into kdd_param_binding values ('filter_by_batch', 'Alert Correlation',
113000023, 'true').
```

This query must be manually executed.

In addition to the job parameters, there is a certain metadata that must be in place in order to successfully run Alert Correlation. These include the definitions of the paths used to correlate events to business entities and the correlation rules that define the criteria for correlating events to events, and the parameters associated to any subsequent actions performed (if this step in the process is chosen to be run). Details on this metadata is provided in the following sub-sections.

### Business Entity Paths

The business entity paths are currently managed through manual interaction with the `KDD_BUS_NTITY_PATH` and `KDD_BUS_NTITY_PATH_CFG` tables in the FSDM. These tables are populated with a comprehensive set of sample data paths. However, the following information will assist in modifying these paths or adding to them. The structure of the tables is as follows:

**Table 22. KDD\_BUS\_NTITY\_PATH (Metadata Table)**

Column Name	Primary Key	Foreign Key	Column Type	Nullable (Y/N)	Default
PATH_ID	*		NUMBER(10)	No	
PATH_NM			VARCHAR2(50)	No	
QUERY_DEF_NM			VARCHAR2(50)	Yes	
ALERT_FOCUS_ID		KDD_CENTRICITY.CNTRY_ID	NUMBER(10)	Yes	
MTCHD_TABLE_NM		KDD_EJB_NAME.EJB_NM	VARCHAR2(50)	Yes	
BUS_NTITY_ID		KDD_CENTRICITY.CNTRY_ID	NUMBER(10)	Yes	

The purpose of this table is to define paths that can be used by the Alert Correlation algorithm to perform the first step in its process, correlating events to business entities. To do this, you must define whether the origin of the path should be the focus of an event or a matched record, by populating either. This is established by either populating the `ALERT_FOCUS_ID` column (indicating that the origin should be the focus of the event), or the `MTCHD_TABLE_NM` column (indicating that the origin should be a matched record of the event). The destination of the path (the business entity we are trying to correlate to by executing this path) is defined by the `BUS_NTITY_ID` column.

The actual SQL to execute to establish the relationship between the event's focus or matched record and this business entity defined by a "query definition" represented in the `KDD_QUERY_DEFS` table as follows:

- The QUERY\_DEF\_NM column provides a name for the query definition.
  - The FILTER\_TABLE\_NM provides the name of the source data table containing the data for the business entity we are trying to correlate to.
  - The FILTER\_ATTR\_NM provides the column name from the FILTER\_TABLE\_NM that defines the focal attribute or matched record attribute (path origin) that we are filtering business entity source data records by (path destination).
  - The FILTER\_ATTR\_TYPE\_CD provides the type code of this attribute (L for long/numeric, S for string).
- Finally, the SQL\_TX provides the actual query where we must select three columns:
- origin key id (focal/matched-attribute key ID)
  - destination key id (business entity key ID)
  - display id (business entity display ID)

For example, if we are trying to establish an event-to-business-entity path/correlation from an event's focal account to primary customer, the record in KDD\_QUERY\_DEFS would be defined as follows: QUERY\_DEF\_NM of "AC to CU-Prmry", FILTER\_TABLE\_NM of "ACCT", FILTER\_ATTR\_NM of "ACCT\_INTRL\_ID", FILTER\_ATTR\_TYPE\_CD of "S", and SQL\_TX of "SELECT ACCT\_INTRL\_ID, PRMRY\_CUST\_INTRL\_ID, PRMRY\_CUST\_INTRL\_ID FROM BUSINESS.ACCT WHERE ACCT.PRMRY\_CUST\_INTRL\_ID is NOT NULL" The Alert Correlation engine will add a filter to this query at run-time based on the FILTER\_TABLE\_NM and FILTER\_ATTR\_NM (In this example it would add "AND ACCT.ACCT\_INTRL\_ID in (?)" where "?" would be replaced with the event's focal entity ID).

The PATH\_ID and PATH\_NM in the table above are used to establish unique identifiers for this path.

The above paths may not necessarily apply to all types of events, and they may have different levels of importance depending on what types of events they are applied to. This variance is defined by a path configuration, which is stored in the KDD\_BUS\_NTITY\_PATH\_CFG table. Its structure is as follows:

**Table 23. KDD\_BUS\_NTITY\_PATH\_CFG (Metadata Table)**

Column Name	Primary Key	Foreign Key	Column Type	Nullable (Y/N)	Default
PATH_CFG_ID	*		NUMBER(10)	No	
PATH_ID		KDD_BUS_NTITY_PATH.PATH_ID	NUMBER(10)	No	
SCNRO_ID		KDD_SCNRO.SCNRO_ID	NUMBER(10)	Yes	
SCNRO_CLASS_CD		KDD_SCNRO_CLASS.SCNRO_CLASS_CD	VARCHAR2(3)	Yes	
PRSDNC_NB			NUMBER(10)	Yes	

We can choose to apply the path identified by the PATH\_ID in this table to only events of a certain scenario or scenario group. This is established by populating either the SCNRO\_ID or the SCNRO\_CLASS\_CD column, respectively. If neither of these columns are populated, this path configuration is considered for an event of any scenario or scenario group. The “importance” or “strength” of a correlation determined by this path may vary depending on the scenario or scenario group of the event. This is defined by the PRSDNC\_NB (the lower the number, the higher the precedence). A NULL PRSDNC\_NB indicates not to apply this PATH\_ID to any events of this SCNRO\_ID or SCNRO\_CLASS\_CD.



## Correlation Rules

Once events are correlated to business entities, the event-to-business entity relationships can be used to correlate events to each other. Events will be grouped into a correlation if they share common business entities, and if they meet the criteria defined in the Alert Correlation Rules. These rules are managed through the Alert Correlation Rule Migration Utility. The logic of an Alert Correlation Rule is defined in XML, and the Alert Correlation Rule Migration Utility is responsible for reading this XML from a file, validating it, and inserting it into the KDD\_CORR\_RULE table.

**Note:** You can set the precedence for each rule in the KDD\_CORR\_RULE table by providing the appropriate precedence number in the PRECEDENCE\_NB column.

For more information on validating/loading correlation rules, refer to the *Repeat for each environment, remembering to change the values for min, max, and current.* section. The following is an example of the rule logic defined in an Alert Correlation Rule XML file, followed by detailed descriptions of the elements contained in the XML file:

```
<CorrelationRule id="123" name="Possible Identity Theft">
  <MinAlertCount>2</MinAlertCount>
  <PrecedenceThreshold>5</PrecedenceThreshold>
  <AlertAttrOperations>
  <![CDATA[ (CORR.SCORE_CT >= 0) OR (CORR.ALERT_CT > 2) ]]>
  </AlertAttrOperations>
  <Lookback number="1" unit="D"/>
  <Scenarios>
    <Scenario id="234"/>
    <Scenario id="345"/>
  </Scenarios>
  <ExistingCorrelationParams>
    <ExtendFlag>TRUE</ExtendFlag>
    <NonExtendableCaseStatuses>
      <CaseStatus>CCL</CaseStatus>
      <CaseStatus>NVST</CaseStatus>
    </NonExtendableCaseStatuses>
  </ExistingCorrelationParams>
  <Actions>
    <Scoring strategy="MAX" incStrategy="ALERT_COUNT"/>
    <CasePromotion>
      <FocusTypePref>CU,AC</FocusTypePref>
      <AlertCorrAttrOperations>
        <![CDATA[(CORR.SCNRO_ID = 114000074 ) AND (CORR.SCNRO_CT) >= 3]]>
      </AlertCorrAttrOperations>
      <ExistingCasePromoteLossRcvryData>TRUE
      </ExistingCasePromoteLossRcvryData>
      <Case type="AML" subtype="SAR" subClassTagLevel1="CHK_FRD"
        subClassTagLevel2="ALTD_INST"/>
    </CasePromotion>
  </Actions>
</CorrelationRule>
```

- **MinAlertCount** (*required*): The minimum number of events involved in a correlation for it to be considered a valid correlation. The minimum acceptable value is 2.

- **PrecedenceThreshold** (*required*): Number indicating the maximum precedence value that a business entity shared between events must have in order to be considered a correlation by this rule. The lower the precedence number the stronger the relationship. Events will not be considered for the correlation unless the precedence number associated with the business entity-to-event is less than or equal to ( $\leq$ ) the value defined.
- **AlertAttrOperations** (*optional*): Defines operations used to further constrain the events to be used for correlation. An operation consists of an event attribute (identified by `ATTR_NM`) compared to a numerical value, such as *from alert* and *to alert* which can be correlated if they both have `SCORE_CT >= 0`, represented by `CORR.SCORE_CT >= 0`, or a *from alert* and *to alert* which can be correlated if `CORR.ALERT_CT > 2`. The set of supported comparison operators are: `=`, `!=`, `<`, `>`, `<=`, `>=`, `IN`, and `NOT IN`. Note that because the `SCNRO_ID` attribute of both events and correlations can potentially have multiple values, only the `IN` and `NOT IN` operators should be used in expressions involving `SCNRO_ID`. The rest of the operators can only support a single value operands. Also, there should be no space in the scenario ID list specified. For example, `BOTH.SCNRO_ID IN (115600002,114690101)` is correct and `BOTH.SCNRO_ID IN (115600002, 114690101)` is incorrect.

Multiple operations can be strung together by logical `AND` and `OR` operators and operation precedence can be defined with parentheses. Note that the text of an *AlertAttrOperation* must be wrapped in a `CDATA` tag as above to account for any special XML characters contained in the expression, such as `>` or `<`.

- **Lookback** (*optional*): The *number* attribute indicates the number of seconds/minutes/hours/days to look back from the current date/time to create a time window in which to consider events for correlation. This is a create timestamp of the event. The *unit* attribute identifies the unit of the lookback number. Possible values are S, M, H, D, and CM for seconds, minutes, hours, days, and current month, respectively. All of these require a valid number value except for CM, which essentially just makes the lookback the 1st of the current month, such as if the current date is October 14, we will look back to October 1 if the CM unit is selected. The create timestamp of the event is used to determine whether or not an event falls within the lookback period.

---

**Note:** Do not use a unit less granular than a day in rules intended for batch events (S, M, and H are intended for posted events). For batch processing, use D or CM as a unit.

---

- **Scenarios** (*optional*): Identifies the Scenario(s) an event should have been generated from in order to be considered for a correlation by this rule. If not specified, system will consider all the scenarios.
- **ExistingCorrelationParams** (*required*): Defines the conditions for extending existing correlations. When a new correlation is discovered, it is possible that it is a superset (with only the triggering event not already included in the existing correlation) of a correlation that has previously been identified. `ExtendFlag` defines whether this correlation rule can result in extending an existing correlation. If this is set to `FALSE` (do not extend) then a new correlation is created when this condition is identified. If it is set to `TRUE` then the existing correlation is added to unless it has already been promoted to a case that has a status identified in the `CaseStatus` tags of `NonExtendableCaseStatuses`.
- **Actions** (*optional*): Once correlations are discovered, multiple types of actions can be taken on the correlation. These actions and their associated parameters are defined in between the `Actions` tags. The current set of possible actions include scoring the correlation and promoting the correlation to a case.
- **Scoring** (*optional*): The *strategy* attribute defines whether the correlation score should be derived from the max of the associated event scores (`MAX_SCORE`) or the average of the associated event scores (`AVERAGE_SCORE`). The *incStrategy* attribute provides the option of defining a fixed score to be added to the correlation score. The possible values can be `ALERT_COUNT` (each additional event above *MinAlertCount* adds to the score),

SCENARIO\_COUNT (each distinct scenario (starting with the second scenario) involved in the correlation adds to the score), or NONE (the score is not incremented above what has already been calculated).

**Note:** The calculated correlation score is bounded by the values of the *min\_correlation\_score* and *max\_correlation\_score* parameters found in the following configuration files:

```
<OFSAAI Installed Directory>/behavior_detection/algorithms/mantas_cfg/  
install.cfg (for the Alert Correlation batch algorithm)
```

```
<OFSAAI Installed Directory>/services/install.cfg (for the Alert Correlation step of the  
PostAlert operation of the behavior detection Service)
```

- **CasePromotion** (*optional*): Defines the parameters used to determine whether a newly discovered correlation should be promoted to a case. Correlations that are already part of a case, such as when a correlation is extended, are not considered by this type of rule, except the `ExistingCasePromoteLossRcvryData` element, which determines whether or not to augment the existing case's fraud loss and recovery data with the related data from the new events added to the case. Logical operations based on attributes of the correlation (including scenarios involved in the correlation) defined under *AlertCorrAttrOperations* can be used to determine whether or not the correlation should be promoted to a case. The syntax, supported operators, and others are same as that of the *AlertAttrOperations* defined above (including the CDATA requirement). If the conditions result in the promotion of a correlation to a case, the resulting type, subtype, subclass tag level 1, and subclass tag level 2 of the case are determined by the *type*, *subtype*, *subClassTagLevel1*, and *subClassTagLevel2* attributes of the Case element. The focus of the case is determined by using the ordered list of preferred business entity types defined in the `FocusTypePref` element. In the example above, if the events involved in the associated correlation are correlated to the same CUSTOMER then CUSTOMER would become the focus of the case. If not, and if they are correlated to the same ACCOUNT, ACCOUNT would become the focus of the case. If not, the correlation will not be promoted to a case.

**Note:** This is only applicable if your firm has implemented Enterprise Case Management.

## Activating or Deactivating Correlation Rules

Running the Alert Correlation job will execute only those correlation rules that are designated as Active. Rules that are designated as Inactive will be ignored and not executed. To deactivate an active correlation rule the correlation rule metadata need to be modified to change `KDD_CORR_RULE.STATUS_CD` from a value of ACT to NULL. To activate an inactive rule, modify `KDD_CORR_RULE.STATUS_CD` from a value of NULL to ACT. Changes made to the metadata are effective immediately and will be utilized the next time event correlation is run.

## Custom Scoring Rules

The custom scoring rules enhancement allows users to configure rules which define the attributes that are evaluated before arriving at a defined score. Rules are PL/SQL procedures, so these attributes can be bindings, matched attributes or any other event data.

The following is a sample rule with a temporary table called `@corr_score_temp`, which is a dynamic table:

```
DECLARE  
  c_corr_id @corr_score_temp.corr_id%type;  
  c_score_ct @corr_score_temp.score_ct%type;  
  CURSOR c_corr_scoring is
```

```
select z.corr_id,sum (z.score_ct_curr)score_ct from
(

  select distinct a.corr_id,a.score_ct,a.score_ct+06 score_ct_curr
  from @corr_score_temp a
  INNER JOIN kdd_review kr ON kr.review_id =a.review_id
  INNER JOIN kdd_break kb ON kr.review_id =kb.prnt_break_id
  INNER JOIN kdd_break_binding kbb ON kb.break_id= kbb.break_id and
kbb.bindg_nm='Tot_Trxn_Am' and kbb.value_tx>=30000

  union

  select distinct b.corr_id,b.score_ct,b.score_ct+16 score_ct_curr
  from @corr_score_temp b
  INNER JOIN kdd_review_scnro krs ON krs.review_id = b.review_id
  group by b.corr_id, b.score_ct
  having count(distinct(krs.scnro_id))>=2

  union

  select distinct c.corr_id,c.score_ct,c.score_ct+26 score_ct_curr
  from @corr_score_temp c,cust cu,kdd_review kr
  where c.review_id= kr.review_id
  and kr.focal_ntity_dsply_id = 'CUMLN0AAC-701'
  and kr.creat_ts BETWEEN TO_DATE('2016/11/09', 'yyyy/mm/dd') AND
      TO_DATE('2016/11/09', 'yyyy/mm/dd')

  union

  select distinct d.corr_id, d.score_ct,d.score_ct+36 score_ct_curr
  from @corr_score_temp d
  INNER JOIN kdd_review_scnro krs ON krs.review_id=d.review_id and krs.scnro_id =
'114000081'

  union

  select distinct e.corr_id,e.score_ct,e.score_ct+46 score_ct_curr
  from @corr_score_temp e
```

```
INNER JOIN kdd_review_bus_ntity krbn on krbn.review_id = e.review_id and
krbn.bus_ntity_id='113000004'

INNER JOIN cust cu ON krbn.bus_ntity_key_id = cu.cust_intrl_id and
cu.cust_bus_risk_nb >= 6

) z
group by z.corr_id,
        z.score_ct;

BEGIN
    OPEN c_corr_scoring;
    LOOP
        FETCH c_corr_scoring into c_corr_id, c_score_ct;
        EXIT WHEN c_corr_scoring%notfound;
        UPDATE @corr_score_temp z SET z.SCORE_CT = c_score_ct WHERE z.CORR_ID =
c_corr_id;
    END LOOP;
    CLOSE c_corr_scoring;
    COMMIT;
END;
```

---

**Note:** An entry is created in the log file after the correlation job is executed and the @corr\_score\_temp placeholder is replaced with the temporary table that is created when the job is executed.

---

---

**Note:** If the job is executed successfully, the temporary table is truncated. If the job fails, the temporary table remains in the database.

---

The following is the rule with the @corr\_score\_temp table replaced with a CORR\_SCORE\_TMP table:

```
DECLARE
    c_corr_id CORR_SCORE_TMP_560201.corr_id%type;
    c_score_ct CORR_SCORE_TMP_560201.score_ct%type;
    CURSOR c_corr_scoring is

select z.corr_id,sum (z.score_ct_curr)score_ct from
(

select distinct a.corr_id,a.score_ct,a.score_ct+06 score_ct_curr
from CORR_SCORE_TMP_560201 a
```

```
INNER JOIN kdd_review kr ON kr.review_id =a.review_id
INNER JOIN kdd_break kb ON kr.review_id =kb.prnt_break_id
INNER JOIN kdd_break_binding kbb ON kb.break_id= kbb.break_id and
kbb.bindg_nm='Tot_Trxn_Am' and kbb.value_tx>=30000
```

union

```
select distinct b.corr_id,b.score_ct,b.score_ct+16 score_ct_curr
from CORR_SCORE_TMP_560201 b
INNER JOIN kdd_review_scnro krs ON krs.review_id = b.review_id
group by b.corr_id, b.score_ct
having count(distinct(krs.scnro_id))>=2
```

union

```
select distinct c.corr_id,c.score_ct,c.score_ct+26 score_ct_curr
from CORR_SCORE_TMP_560201 c,cust cu,kdd_review kr
where c.review_id= kr.review_id
and kr.focal_ntity_dsply_id = 'CUMLN0AAC-701'
and kr.creat_ts BETWEEN TO_DATE('2016/11/09', 'yyyy/mm/dd') AND
      TO_DATE('2016/11/09', 'yyyy/mm/dd')
```

union

```
select distinct d.corr_id, d.score_ct,d.score_ct+36 score_ct_curr
from CORR_SCORE_TMP_560201 d
INNER JOIN kdd_review_scnro krs ON krs.review_id=d.review_id and krs.scnro_id =
'114000081'
```

union

```
select distinct e.corr_id,e.score_ct,e.score_ct+46 score_ct_curr
from CORR_SCORE_TMP_560201 e
INNER JOIN kdd_review_bus_ntity krbn on krbn.review_id = e.review_id and
krbn.bus_ntity_id='113000004'
INNER JOIN cust cu ON krbn.bus_ntity_key_id = cu.cust_intrl_id and
cu.cust_bus_risk_nb >= 6
```

```

) z
group by z.corr_id,
        z.score_ct;

BEGIN
  OPEN c_corr_scoring;
  LOOP
    FETCH c_corr_scoring into c_corr_id, c_score_ct;
    EXIT WHEN c_corr_scoring%notfound;
    UPDATE CORR_SCORE_TMP_560201 z SET z.SCORE_CT = c_score_ct WHERE z.CORR_ID =
c_corr_id;
  END LOOP;
  CLOSE c_corr_scoring;
  COMMIT;
END;
```

### Configuring Rules

This section covers the following topics:

- [Structure of the Configuration Table](#)
- [Structure of the Temporary Table](#)
- [Configuring Custom Rules](#)

#### **Structure of the Configuration Table**

The KDD\_CORR\_SCRNG\_CSTM\_RULE is a metadata table which is added to configure rules.

**Table 24. Structure of the Configuration Table**

Column Name	Primary Key	Column Type	Nullable?
RULE_ID	*	NUMBER (10)	NO
STATUS_CD		VARCHAR2 (50)	NO
SQL_TX		CLOB	NO

Following are the description for each column name:

- **RULE\_ID:** This column contains a unique number which identifies the event correlation scoring rule.
- **STATUS\_CD:** This column contains either ACT or INACT, which stands for active or inactive respectively.
- **SQL\_TX:** This column contains the PL/SQL procedure that contains the score evaluation logic.

#### **Structure of the Temporary Table**

The CORR\_SCORE\_TMP\_<run ID> is a temporary table which is created when the correlation job is run

**Table 25. Structure of the Temporary Table**

Column Name	Primary Key	Column Type	Nullable?
CORR_ID	*	NUMBER (10)	NO
REVIEW_ID	*	NUMBER (10)	NO
SCORE_CT		NUMBER (10)	NO
CORR_RULE_ID	*	NUMBER (10)	NO

Following are the description for each column name:

- **CORR\_ID:** This column contains a unique number which identifies the event correlation that is scored.
- **REVIEW\_ID:** This column contains a unique number that identifies an event in the correlation.
- **SCORE\_CT:** This column contains the correlation score.
- **CORR\_RULE\_ID:** This column contains the conditions which need to be satisfied in the CLOB column in order to promote the event to a case.

### Configuring Custom Rules

The following are the steps to configure custom rules:

1. Create a PL/SQL procedure that has the logic for scoring with a place holder for the temporary table.
2. Make an entry in the `KDD_CORR_SCRNG_CSTM_RULE` with `STATUS_CD` as `ACT`.
3. Change the score strategy to `CUSTOM_SCORE` in the `correlation_rule.xml` file. (`KDD_CORR_RULE.RULE_OP_TX`) Example: `<Scoring strategy="CUSTOM_SCORE" incStrategy="ALERT_COUNT" />`
4. A new `install.cfg` property called `alert_alert_corr.ptc.threshold`, is added. This is an integer value. Correlations are promoted to a case only if the correlation score exceeds this threshold.

### Sample Alert Correlation Rules

OFSBD delivers two sample alert correlation rules:

- **Correlated Alerts by Business Entity:** Groups events created in the past month based on a common correlated business entity. For example, this rule would correlate all events with a business entity-to-event correlation on customer CU12345 that were created in the past month.
- **Potential Identity Theft:** Groups events created in the past seven days that are generated on one or more specified scenarios where the events share a common correlated business entity. Specified scenarios are those scenarios which identify behaviors that, in isolation or when considered as a whole, may be indicative of identity theft. For example, this rule would correlate all events generated on one or more of the specified scenarios with a business entity-to-event correlation to CU12345 that were created in the past seven days.

OFSBD installs these sample alert correlation rules in the `<OFSAAI Installed Directory>/database/db_tools/data` directory.

### Displaying Alert-to-Business Entity Path Details on the User Interface

To view Alert-to-Business Entity Path rules in the UI (in addition to the default rules), you must add entries to `KDD_CODE_SET_TRNLN` with a `CODE_SET` value of "Relationship", a `CODE_VAL` value that corresponds to the `KDD_BUS_NTITY_PATH.PATH_NM` of the new rule, and a `CODE_DISP_TX` with the desired display



text to appear on the UI for the new rule. All correlation rules must also be added to the User/Organization under the Security Attribute Administration screen.



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:

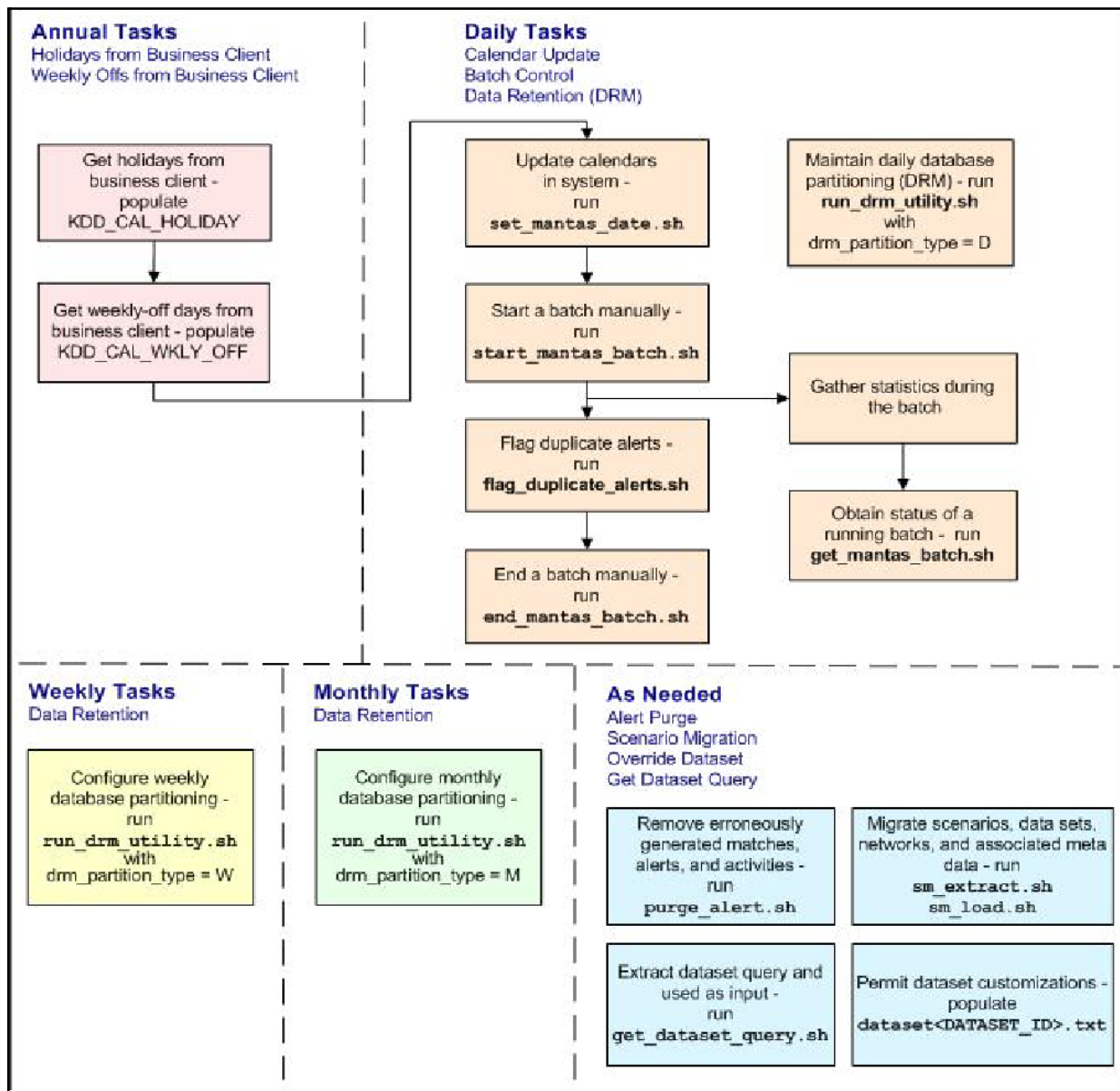
- [About Batch Processing Utilities](#)
- [Managing Common Resources for Batch Processing Utilities](#)
- [Managing Annual Activities](#)
- [Managing Batch Control Utility](#)
- [Managing Calendar Manager Utility.](#)
- [Managing Data Retention Manager](#)
- [Database Statistics Management](#)

## ***About Batch Processing Utilities***

Behavior Detection 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 Notification:** Enables you to configure users of behavior detection 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.
- **Managing Truncate Manager:** Truncates tables that require complete replacement of their data.

Figure 39 illustrates the frequency with which you use these batch-related database utilities when managing activities: daily, weekly, monthly, annually, or as needed.



**Figure 39. Managing Database Activities with Utilities**

Figure 39 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 Behavior Detection 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.

## 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](#)

### 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 40 (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. You should 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:Ti5012L64
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==
```

```
##### 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
```

*(Continued on next page)*

*(Continued from previous page)*

```
##### 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

#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
```

*(Continued on next page)*

*(Continued from previous page)*

```
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/Ti5012L64

#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=${schema.market.owner}
extract.user.miner=${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/te
mp

#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
restriced, you can use range.check

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

*(Continued on next page)*



*(Continued from previous page)*

```
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=Ti5012L64
# 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
(Continued on next page)
```

*(Continued from previous page)*

```
# 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.
purge_fatca_flag=Y

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

# Specify default rule set.
purge_default_rule_set=

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

*(Continued on next page)*

*(Continued from previous page)*

```
# 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/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

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

# Max number of active db connections
utils.database.max_connections=4
```

*(Continued on next page)*

*(Continued from previous page)*

```
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
```

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

```
#
# Username and password for connecting to the database

dat.database.username=${ingest.user}
dat.database.password=${ingest.password}
```

*(Continued on next page)*

*(Continued from previous page)*

```
# 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/a
nalysis.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

dbscript.execution.users=(system|business|mantas|market|miner|ingest|report|kdd|a
lgorithms|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=
```

*(Continued on next page)*

*(Continued from previous page)*

```
##### 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 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
```

*(Continued on next page)*

*(Continued from previous page)*

```
# (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
# Specify the full path to the categories.cfg file
log.categories.file.path=/scratch/ofsaadb/BD802_Final/BD802FL/database/db_tools/m
antas_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/log
s/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 40. Sample install.cfg File**

### **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 41 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</FileName>
    <PatternLayout>
        <Pattern>[%d{E dd/M/yyyy hh:mm:ss}] [CALENDAR_MANAGER] [%5p] - %m%n</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">
    <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>
```

*(Continued on next page)*

*(Continued from previous page)*

```
<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"/>
  </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)*

*(Continued from previous 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">
  <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>

```

*(Continued on next page)*

*(Continued from previous page)*

```
<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"/>
  </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>
```

*(Continued on next page)*

*(Continued from previous page)*

```

<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>
  <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>

```

*(Continued on next page)*

*(Continued from previous page)*

```
<SizeBasedTriggeringPolicy size="10000kb"/>
  </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
    </pattern>>
  </PatternLayout>
</Console>
</Appenders>
```

*(Continued on next page)*

*(Continued from previous page)*

```
<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>

<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"/>
```

*(Continued on next page)*

*(Continued from previous page)*

```
</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>

  <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>
```

*(Continued on next page)*



*(Continued from previous page)*

```

    <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">
    <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 41. Sample Logging Information in the Log4j2.xml File**

## ***Managing Annual Activities***

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

This section covers the following topics:

- [Loading Holidays](#)
- [Loading Non-business Days](#)

### **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 42 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 42. Sample KDD\_CAL\_HOLIDAY Table Loading Script**

The following table describes the contents of the KDD\_CAL\_HOLIDAY table.

**Table 26. 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.

## 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) VALUES (
  'SYSCAL', 1);

INSERT INTO KDD_CAL_WKLY_OFF (CLNDR_NM, DAY_OF_WK) VALUES (
  'SYSCAL', 7);

COMMIT;
```

**Figure 43. 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 27. 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.

## 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](#)

### Directory Structure

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

**Table 28. Alert Purge Utility Directory Structure**

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

## 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* on page 87 and Appendix A, *Logging*, on page 245 for more information.

## 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

---

**Note:** 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.

---

## 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 39). 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](#)

### Configuring the Alert Purge Utility

To configure the Alert Purge Utility, follow these steps:

1. Navigate to the <OFSAAI Installed Directory>/database/db\_tools/mantas\_cfg.
2. 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 40). 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.
purge_fatca_flag=Y
```

*(Continued on next page)*



*(Continued from previous page)*

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

# Specify default 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 44. 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 29. Alert Purge Utility Parameters**

Parameter	Description
purge	Determines how the utility performs processing, depending on the specified value: <ul style="list-style-type: none"> <li>● N (default): Performs all processing up to the point of the purge. The utility identifies resulting matches, alerts, and actions, but performs no purging.</li> <li>● Y: Performs the above in addition to purging matches, alerts, and actions.</li> </ul>
limit_matches	Identifies restrictions on the matches to delete: <ul style="list-style-type: none"> <li>● 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).</li> <li>● 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.</li> </ul> <p><b>Note:</b> The system purges matches that do not relate to alerts, regardless of the value of <code>limit_matches</code>.</p>
batch_size	<i>Optional:</i> 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: <ul style="list-style-type: none"> <li>● 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.</li> <li>● Y (default): Purges the alerts as identified by the purge rule used to perform the purge operation.</li> </ul>
purge_case_flag	Determines whether or not the utility would purge cases, depending on the specified value: <ul style="list-style-type: none"> <li>● 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.</li> <li>● Y (default): Purges the cases as identified by the purge rule used to perform the purge operation.</li> </ul>
purge_default_rule_set	<i>(Optional)</i> 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	<i>(Optional)</i> 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 <code>Purge_&lt;YYYYMMDD.HH.MM.SS&gt;.txt</code> . Here <code>&lt;YYYYMMDD.HH.MM.SS&gt;</code> represents current timestamp when the utility was executed.
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 45. Configuration Information**

The following table describes the Purge Rules Configuration Parameters.

**Table 30. 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. <b>Note:</b> 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 <code>comma</code> .
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 <code>comma</code> . <b>Note:</b> 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 <code>comma</code> . <b>Note:</b> This property is specific to alerts only. This should not be specified for cases

**Table 30. Alert Purge Utility Parameters (Continued)**

Parameter	Description
CreateDate	<p>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.</p> <ul style="list-style-type: none"> <li>● <b>StartDate:</b> 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.</li> <li>● <b>EndDate:</b> Identifies the date up to which the alerts are to be purged. The date should be provided in the format YYYY-MM-DD</li> <li>● <b>Age:</b> 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.</li> </ul> <p>The example below gives more details: (Assume Current date: 21 NOV 2012)</p> <p>Case1:</p> <p>(i) if age = 1Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2011 (includes both days)</p> <p>(ii) if age = 5Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2007 (includes both days)</p> <p>Case2:</p> <p>(i) if age = 1M: Date range would be considered: 21 NOV 2012 to 21 OCT 2012 (includes both days)</p> <p>(ii) if age = 5M: Date range would be considered: 21 NOV 2012 to 21 JUN 2012 (includes both days)</p> <p>Case3:</p> <p>(i) if age = 1D: Date range would be considered: 21 NOV 2012 to 20 NOV 2012 (includes both days)</p> <p>(ii) if age = 5D: Date range would be considered: 21 NOV 2012 to 16 NOV 2012 (includes both days)</p> <p>(iii) if age = 0D: Date range would be considered: 21 NOV 2012 to 21 NOV 2012 (that is, current date only)</p> <p><b>Note:</b> 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.</p>
BatchId	<p>Identifies the list of Batch IDs for which the alerts should be purged.</p> <p><b>Note:</b> This property is specific to alerts only. This should not be specified for cases.</p>
DomainCode	<p>Identifies the list of domains for which the alerts should be purged. Acceptable values include:</p> <ul style="list-style-type: none"> <li>● MTS</li> <li>● TST</li> <li>● PFM</li> <li>● NVZ</li> </ul> <p><b>Note:</b> This property is specific to alerts only. This should not be specified for cases.</p>

Table 30. Alert Purge Utility Parameters (Continued)

Parameter	Description
LastActionDate	<p>Identifies the dates to be considered for purging the alerts and cases by the 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.</p> <ul style="list-style-type: none"> <li>● <b>StartDate:</b> 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</li> <li>● <b>EndDate:</b> Identifies the date up to which the alerts are to be purged. The date should be provided in the format YYYY-MM-DD</li> <li>● <b>Age:</b> 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.</li> </ul> <p>The example below gives more details: (Assume Current date: 21 NOV 2012)</p> <p>Case1:</p> <p>(i) if age = 1Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2011 (includes both days)</p> <p>(ii) if age = 5Y: Date range would be considered: 21 NOV 2012 to 21 NOV 2007 (includes both days)</p> <p>Case2:</p> <p>(i) if age = 1M: Date range would be considered: 21 NOV 2012 to 21 OCT 2012 (includes both days)</p> <p>(ii) if age = 5M: Date range would be considered: 21 NOV 2012 to 21 JUN 2012 (includes both days)</p> <p>Case3:</p> <p>(i) if age = 1D: Date range would be considered: 21 NOV 2012 to 20 NOV 2012 (includes both days)</p> <p>(ii) if age = 5D: Date range would be considered: 21 NOV 2012 to 16 NOV 2012 (includes both days)</p> <p>(iii) if age = 0D: Date range would be considered: 21 NOV 2012 to 21 NOV 2012 (that is, current date only)</p> <p><b>Note:</b> 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.</p>
Status	<p>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 <code>comma</code>.</p>
JobIds	<p>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 <code>comma</code>.</p> <p><b>Note:</b> This property is specific to alerts only. This should not be specified for cases.</p>
ThresholdSetIds	<p>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 <code>comma</code>.</p> <p><b>Note:</b> This property is specific to alerts only. This should not be specified for cases.</p>

## 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.

---

## 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.

---

---

**Note:** 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.

---

### ***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.

### **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.

#### **Example 1**

If user specifies only one rule 'xyz' for purging alerts and assume 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.

### Example 2

If 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 de-scribed 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.

## Managing Batch Control Utility

The Batch Control Utility enables you to manage and record the beginning and ending of a Behavior Detection 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* on page 87 for more information).

This section covers the following topics:

- [Batches in Behavior Detection](#)
- [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).

---

## Batches in Behavior Detection

Except for the behavior detection subsystem, batches govern all other activity in the Behavior Detection 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.

Behavior Detection 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.

## Directory Structure

Table 31 provides the directory structure for the Batch Control Utility, in <OFSAAI Installed Directory>/database/db\_tools/:

**Table 31. Batch Control Utility Directory Structure**

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

## 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* on page 87, and Appendix A, *Logging*, on page 199, for more information.

## 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

### 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 40).
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.
```

**Figure 46. 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 40 on page 134, for more information).

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

### Setting Up Batches

OFSBD delivers with a default batch called DLY. The `KDD_PRCNSG_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_PRCNSG_BATCH_SRC` table.

The following table provides the contents of the `KDD_PRCNSG_BATCH` table.

**Table 32. KDD\_PRCNSG\_BATCH Table Contents**

Column Name	Description
<code>PRCSNG_BATCH_NM</code>	Name of the batch , such as DLY.
<code>PRCSNG_BATCH_DSPLY_NM</code>	Readable name for the batch, such as Daily.
<code>PRCSNG_ORDER</code>	Relative order of a batch run within processing.

**Table 32. KDD\_PRCNSG\_BATCH Table Contents**

EOD_BATCH_NM	Name of the batch that is this batch's end-of-day. This name is the same as the name for PRCNSG_BATCH_NM if the row represents an end-of-day batch.
PRCSNG_BATCH_NM	Description of this processing batch.

Each row in the KDD\_PRCNSG\_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](#)

### Single Batch

In this example, the KDD\_PRCNSG\_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\_PRCNSG\_BATCH table may look similar to the example in Table 33.

**Table 33. Sample KDD\_PRCNSG\_BATCH Table with Single Batch**

PRCSNG_BATCH_NM	PRCSNG_BATCH_DSPLY_NM	PRCSNG_ORDER	EOD_BATCH_NM
DLY	Daily Batch	1	DLY

### 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 34 describes.

**Table 34. Sample KDD\_PRCNSG\_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*, on page 132, 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.

### 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\_PRCNSG\_BATCH\_SRC FSDM table. The following columns are available in this table:

**Table 35. KDD\_PRCNSG\_BATCH\_SRC FSDM 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 36 describes.

**Table 36. Sample KDD\_PRCNSG\_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.

### Starting a Batch Process Manually

To start a batch manually, follow these steps:

1. Verify that the Behavior Detection 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. Refer to “*Configuring Console Output*,” for more information.

### 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\_PRCNSG\_BATCH table (Table 36).
2. The utility determines whether a batch is running by querying the KDD\_PRCNSG\_BATCH\_CONTROL table. The following table describes the KDD\_PRCNSG\_BATCH\_CONTROL table.

**Table 37. KDD\_PRCNSG\_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_PRCNSG_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\_PRCNSG\_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\_PRCNSG\_BATCH\_HIST table.

**Table 38. KDD\_PRCNSG\_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 ( <i>RUN</i> ) or has finished ( <i>FIN</i> ).

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 47. 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 47. Sample KDD\_PRCSNG\_BATCH\_HIST Table—Batch Start Status**

### 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 32). To stop a batch process manually, follow these steps:

1. Verify that the Behavior Detection 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.

### 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 37 on page 129).
2. Records information about the batch in the KDD\_PRCSNG\_BATCH\_HIST table (refer to Table 38 on page 129). This table contains a history of all batches that appear by start date and end date. Figure 48 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 48. Sample KDD\_PRCNSG\_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\_PRCNSG\_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.

## Identifying a Running Batch Process

**Caution:** 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.

### 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.

### Obtaining a Batch Name

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

1. The utility retrieves the name of the currently running batch from the KDD\_PRCNSG\_BATCH\_CONTROL table (refer to Table 37 on page 129).

The utility returns the batch name to standard output.

## 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 OFSBD 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

### Directory Structure

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

**Table 39. 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.

### Logs

As the utility updates the calendars in the OFSBD 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* on page 87, and Appendix A, *Logging*, on page 199, for more information.

### 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 26 and Table 27). These tables contain calendar information that an Oracle client has provided regarding observed holidays and non-business days.

## 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](#)

## 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 40). 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.

### 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.

### Starting the Utility Manually

To start the Calendar Manager Utility, follow these steps:

1. Verify that the Behavior Detection 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. refer to *Configuring Console Output*, on page 138, for more information.

### Updating the `KDD_CAL` Table

The Calendar Manager Utility retrieves information that it needs for updating OFSBD 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 40. `KDD_CAL` Table Contents**

Column Name	Description
<code>CLNDR_NM</code>	Specific calendar name.
<code>CLNDR_DT</code>	Date in the range between the lookback and lookforward periods.
<code>CLNDR_DAY_AGE</code>	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 <code>CLNDR_DAY_AGE</code> of 20061128 = 1, and 20061130 = -1.

**Table 40. KDD\_CAL Table Contents (Continued)**

Column Name	Description
BUS_DAY_FL	<p>Flag that indicates whether the specified date is a valid business day (set the flag to Y).</p> <p>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.</p>
BUS_DAY_AGE	<p>Number of business days ahead or behind the provided date.</p> <p>If BUS_DAY_FL is N, BUS_DAY_AGE receives the value of the previous day's BUS_DAY_AGE.</p>
DAY_OF_WK	<p>Value that represents the day of the week: Sunday=1, Monday=2, Tuesday=3, ... Saturday=7.</p>
WK_BNDRY_CD	<p>Week's start day (SD) and end day (ED).</p> <ul style="list-style-type: none"> <li>● 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&lt;x&gt;, where &lt;x&gt; is a numeric counter with the start/end of the week that the provided date is in = 0.</li> <li>● 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&lt;x&gt;.</li> </ul> <p>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.</p>
MNTH_BNDRY_CD	<p>Month's start day (SD) and end day (ED).</p> <ul style="list-style-type: none"> <li>● 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&lt;y&gt;, where y is a numeric counter with the start/end of the month that the provided date is in = 0.</li> <li>● 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&lt;y&gt;.</li> </ul> <p>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.</p>
BUS_DAY_TYPE_CD	<p>Indicates the type of business day:</p> <ul style="list-style-type: none"> <li>● N = Normal</li> <li>● C = Closed</li> <li>● S = Shortened</li> </ul>
SESSN_OPN_TM	<p>Indicates the opening time of the trading session for a shortened day. The format is HHMM.</p>
SESSN_CLS_TM	<p>Indicates the closing time of the trading session for a shortened day. The format is HHMM.</p>

**Table 40. KDD\_CAL Table Contents (Continued)**

Column Name	Description
SESSN_TM_OFFST_TX	Indicates the timezone offset for SESSN_OPN_TM and SESSN_CLS_TM. The format is HH:MM.
QRTR_BNDRY_CD	<p>Quarter's start day (SD) and end day (ED).</p> <ul style="list-style-type: none"> <li>• 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 &lt;y&gt;, where y is a numeric counter with the start/end of the quarter that the provided date is in = 0.</li> <li>• 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 &lt;y&gt;.</li> </ul> <p>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.</p>

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_PRCNG_BATCH_CONTROL.DATA_DUMP_DT` field.

### 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.

## Managing Data Retention Manager

Behavior Detection 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 39 on page 86). 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:

- [Directory Structure](#)
- [Logs](#)
- [Processing Flow](#)
- [Using the Data Retention Manager](#)
- [Utility Work Tables](#)

## Directory Structure

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

**Table 41. Data Retention Manager Directory Structure**

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

## 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*, on page 87, and Appendix A, *Logging*, on page 199, for more information.

## Processing Flow

Figure 49 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.

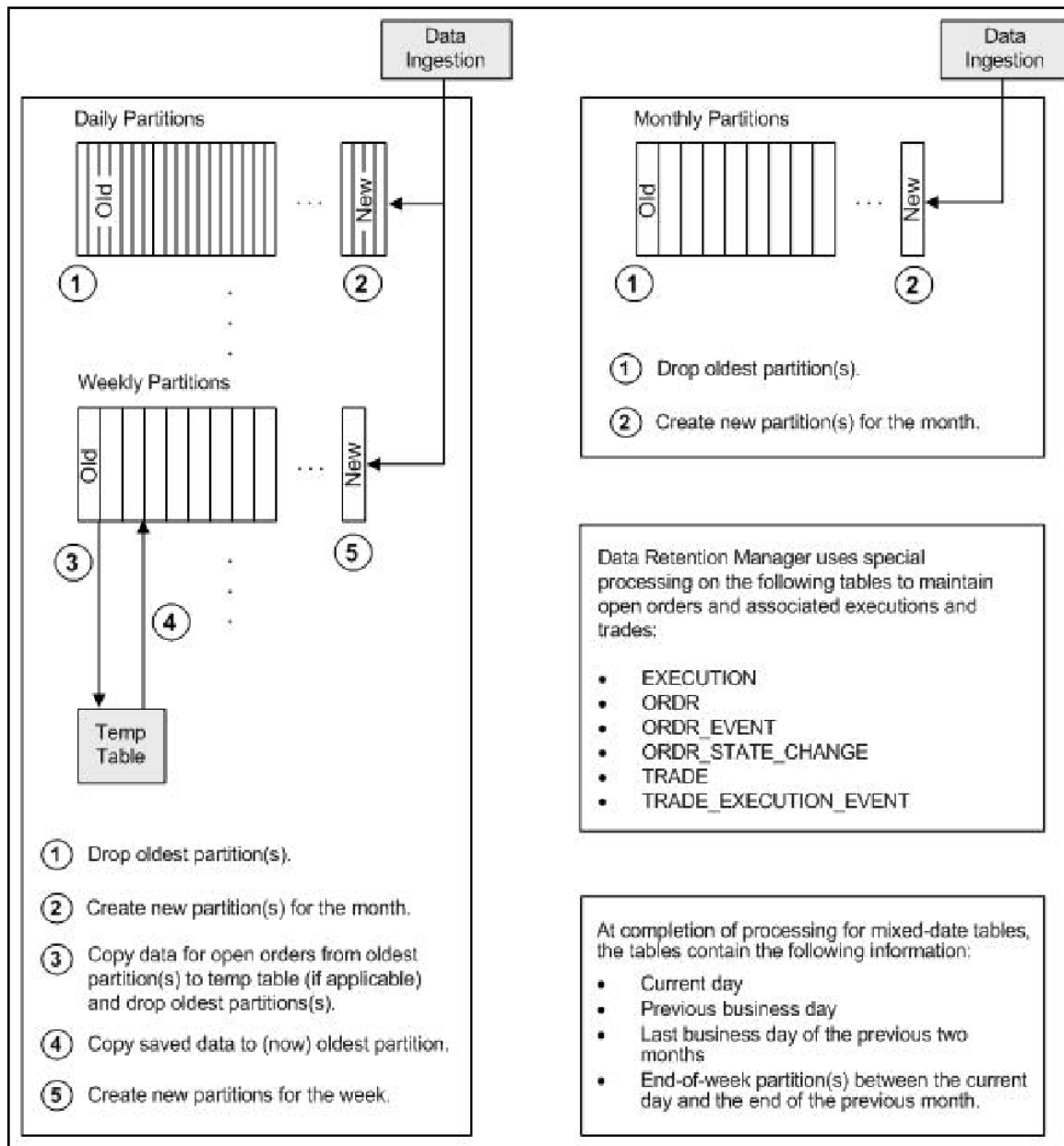


Figure 49. Database Partitioning Process

## 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](#)

## 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 40 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 50. 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 42. 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)
```

### 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.

**Note:** 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.

**Note:** Be sure to 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).

---

### Running the Data Retention Manager

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

1. Verify that the Behavior Detection 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 with the parameters in Table 42:

```
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
```

### Creating Partitions

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

**Table 43. Partition Name Formats**

Partition Type	Format and Description
Monthly	<p>PYYYYMM</p> <p>where YYYY is the four-digit year and MM is the two-digit month for the data in the partition.</p> <p>For example:            Data for November 2006 resides in partition P200611.</p> <hr/> <p><b>Note:</b> The Data Retention Manager uses information in the <code>KDD_CAL</code> table to determine end-of-week and end-of-month boundary dates.</p>
Weekly or Daily	<p>PYYYYMMDD</p> <p>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.</p> <p>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.</p> <hr/> <p><b>Note:</b> The Data Retention Manager uses information in the <code>KDD_CAL</code> table to determine end-of-week and end-of-month boundary dates.</p>

**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.

### 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.

## 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 42.

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 44 on page 144, for more information.

---

## 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*, on page 132, 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.

## 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.

---

## 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:

### Adding a Monthly Database Partition

To add a monthly partition, run the utility's shell script as follows (refer to Table 42 for parameters):

```
run_drm_utility.sh AM M BUSINESS <object> N
```

where AM is the `drm_operation` parameter that implies adding a monthly partition.

### Dropping a Monthly Database Partition

To drop a monthly partition, run the utility's shell script as follows (refer to Table 42 for parameters):

```
run_drm_utility.sh DM M BUSINESS <object> N
```

where, DM is the `drm_operation` parameter that implies dropping a partition.

## 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.

## 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

### 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 44. 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: <ul style="list-style-type: none"><li>● A: Analyze</li><li>● RI: Rebuild Indexes</li><li>● P: Partition</li><li>● RV: Recompile Views</li></ul>
OWNER	Identifies an owner or user of the utility.
TABLE_NM	Identifies a database table.
PARTN_TYPE_CD	Indicates the partition type: <ul style="list-style-type: none"><li>● D: Daily</li><li>● W: Weekly</li><li>● M: Monthly</li><li>● X: Mixed Date</li></ul>

**Table 44. BUSINESS .KDD\_DR\_MAINT\_OPRTN Table Contents (Continued)**

Column Name	Description
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).
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.

**Caution:** For weekly partitioned tables, do not set the value to Y.

### 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 45. 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: <ul style="list-style-type: none"> <li>● RUN: Running</li> <li>● FIN: Finished successfully</li> <li>● ERR: An error occurred</li> <li>● WRN: Finished with a warning</li> </ul>

### 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 46. 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.
RESULT_CD	Result of the process: <ul style="list-style-type: none"><li>● FIN: Finished successfully</li><li>● ERR: An error occurred</li><li>● WRN: Finished with a warning</li></ul>
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 40 for contents of the KDD\_CAL table.

## Database Statistics Management

The system uses a script to manage Oracle database statistics. These statistics determine the appropriate execution path for each database query.

### 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.

## 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 Behavior Detection 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.

---

**Note:** 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.

## ***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.

---

**Note:** The log for these scripts is written in the run\_stored\_procedure.log file under the <OFSAAI Installed Directory>/database/db\_tools/logs directory.

---

**Note:** 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.

## Running Threshold Analyzer

To run the threshold analyzer, follow these steps:

1. Go to ATOMIC schema and execute below 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 -42 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.

## Managing Truncate Manager

The data management subsystem calls the `run_truncate_manager.sh` script to truncate tables that require complete replacement of their data.

### 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
```

### 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 BD 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.

---



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](#)

## 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.

## 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.

## 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 id>DS.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

## Logs

The Scenario Migration Utility produces two log files (Figure 51 on page 154): `load.log` and `extract.log`. These files reside in the following location:

```
<OFSAAI Installed Directory>/database/db_tools/logs
```

## 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

### 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`. The `install.cfg` file contains common configuration information that Scenario Migration and other utilities require for processing. Figure 51 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'.
load.ignore.custom.patterns=N

(Continued on next page)
```

```
#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/ofsaapp/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=${database.driverName}
extract.conn.url=jdbc:oracle:thin:@ofss220074.in.oracle.com:1521:Ti1011L56
#Source System Id
extract.system.id=
# Specify the schema names for Extract
extract.schema.mantas=${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

#Specify the full path in which to place extracted scenarios
extract.dirname=/scratch/ofsaapp/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/ofsaapp/OFSBD 8.0.2/OFSBD
8.0.2_B06/BDP62_B06/database/db_tools/data/temp

(Continued on next page)
```

*(Continued from previous page)*

```
#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=Ti1011L56

# 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}

#Directory where scenario migration files reside for loading
load.dirname=/scratch/ofsaapp/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 51. 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.

---



## 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.

### Configuring General Scenario Migration

The following table describes general scenario migration parameters.

**Table 47. 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.

**Caution:** 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.

### Configuring Scenario Extraction

The following table describes scenario extraction parameters.

**Table 48. 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.

**Table 48. Scenario Extraction Parameters (Continued)**

Parameter	Description
<code>extract.dirname</code>	Full path to the target directory where the utility writes extracted metadata (<OFSAAI Installed Directory>/database/db_tools/data).
<code>extract.backup.dir</code>	Full path to the target directory where the utility writes backups of the extracted metadata (<OFSAAI Installed Directory>/database/db_tools/data/temp).
<code>extract.product.range.only</code>	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.
<code>extract.product.range.check</code>	(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.

### Configuring Scenario Load

The following table describes scenario load parameters.

**Table 49. Scenario Load Parameters**

Parameter	Description
<code>load.conn.driver</code>	Database connection driver that the utility is to use (oracle.jdbc.driver.OracleDriver).
<code>load.conn.url</code>	Database connection string that the Scenario Migration Utility is to use.
<code>load.ignore.custom.patterns</code> =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.
<code>load.schema.mantas</code>	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
<code>load.schema.business</code>	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
<code>load.schema.market</code>	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
<code>load.user.miner</code>	ATOMIC schema owner in the database in which loading of the scenario occurs (ATOMIC).
<code>load.miner.password</code>	Password for the above user.
<code>load.threshold.update</code>	Threshold values from the incoming scenario. <ul style="list-style-type: none"> <li>• Selecting N retains the threshold values from the target environment.</li> <li>• Selecting Y updates thresholds in the target environment to values from the incoming file.</li> </ul>
<code>load.system.id</code>	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.
<code>load.dirname</code>	Directory from which the system loads scenario, network, and dataset XML files.

**Table 49. Scenario Load Parameters (Continued)**

Parameter	Description
verify.target.system	<p>Check target name upon loading metadata files.</p> <ul style="list-style-type: none"> <li>● 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.</li> <li>● 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.</li> </ul>

### 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:

1. Navigate to the
 

```
cd <OFSAAI Installed Directory>/db_tools directory
```
2. 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
3. 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.

### 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
```

2. *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
3. Copy the XML files you plan to load into the directory that the `load.dirname` specifies in the `install.cfg` file.
  4. 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.

### 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.

---

**Caution:** 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.

## 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

## 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.

### *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.

Conflicting sequence IDs are often the cause errors when you migrate a scenario, so it is important to separate the sequence ID range.

### *Separating Sequence ID ranges*

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.

## Sequences to Modify

You should set these sequences before doing any work on scenarios, thresholds, or scoring rules.

Table 50 lists sequences involved and sample values for the Development environment.

**Table 50. 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_SEQUENC E	10000000	10000000	19999999
KDD_JOB	JOB_ID_SEQ	200000000	10000000	19999999
KDD_LINK_ANALYS_NTWRK_ DEFN	NTWRK_DEFN_ID_SEQ	10000000	10000000	19999999
KDD_LINK_ANALYS_TYPE_C D	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

**Table 50. Environment 1 (Development)**

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 51 lists sequences involved and sample values for the Test/UAT environment.

**Table 51. 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_SEQUENC E	20000000	20000000	29999999
KDD_JOB	JOB_ID_SEQ	20000000	20000000	29999999
KDD_LINK_ANALYS_NTWRK_ DEFN	NTWRK_DEFN_ID_SEQ	20000000	20000000	29999999
KDD_LINK_ANALYS_TYPE_C D	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
KDD_TSHLD_HIST	HIST_SEQ_ID_SEQ	20000000	20000000	29999999
KDD_TSHLD_SET	TSHLD_SET_ID_SEQ	20000000	20000000	29999999

Table 52 lists sequences involved and sample values for the Production environment.

**Table 52. 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

**Table 52. Environment 3 (PROD) (Continued)**

TABLE_NM	SEQUENCE_NAME	CURRENT_VALUE	MIN_VALUE	MAX_VALUE
KDD_LINK_ANALYS_NTWRK_DEFN	NTWRK_DEFN_ID_SEQ	30000000	30000000	39999999
KDD_LINK_ANALYS_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.

## ***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 *Oracle Financial Services Scenario Manager User Guide*.

---

**Note:** Changing scenario threshold values can generate significantly more or less 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 *Trade-Based Anti Money Laundering Technical Scenario Description*(.

### **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, on page 164 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.

---

## 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

### 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 53 shows how mutually exclusive thresholds work in two different situations.

**Table 53. 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).

### 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.

## About the Threshold Editor Screen Elements

The following screen elements display in the Threshold Editor:

- Search Bar
- <Scenario–Threshold Set> Area

### Search Bar

The search bar allows you to search for threshold values by selecting a specific scenario and threshold set (Figure 52).

Figure 52. 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.

### <Scenario–Threshold Set> Area

The <Scenario-Threshold Set> Area displays the list of threshold values for a selected scenario and threshold set (Figure 53). 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 53. <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*, on page 163 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.

**Figure 54. Scenario Test Execution window**

The Scenario Test Execution window displays the following fields:

**Table 54. 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. <b>Note:</b> 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. <b>Note:</b> If a Batch with the selected Processing Batch Name and Date is already running, then the following error message is displayed: <i>A Batch with the selected Processing Batch Name and Date is already running. Please wait till the Batch completes.</i>

- **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.

## 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:

- Changing a Scenario Threshold
- Resetting a Scenario Threshold to the Sample Values
- Viewing a Scenario Threshold’s History
- Viewing Expanded Comments

### 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>.

### 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.

### 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.

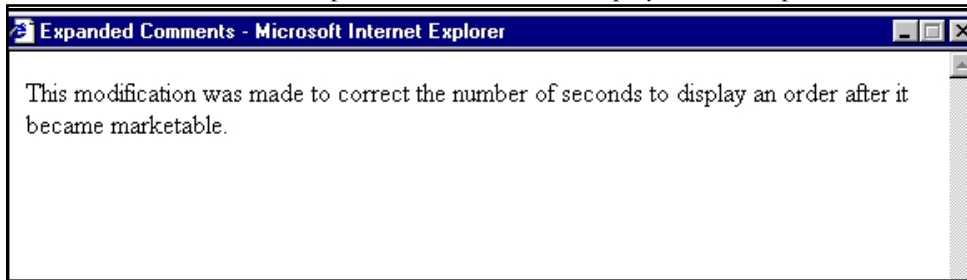
2. Click **Contract** next to the threshold to hide the Threshold History inset.

### 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 55).



**Figure 55. Example Expanded Comment Dialog Box**

2. Click **X (Close button)** on the top right corner to close the dialog box.

This chapter explains how to import the .dxi files into the Enterprise Data Quality (EDQ) application, run the EDQ jobs, and change the EDQ URL for the TBAML application.

This chapter focuses on the following topics:

- [About EDQ](#)
- [EDQ Configuration Process Flow](#)
- [General EDQ Configurations](#)

## About EDQ

The Oracle Financial Services TBAML application is built using EDQ as a platform. EDQ provides a comprehensive data quality management environment that is used to understand, improve, protect and govern data quality. EDQ facilitates best practices such as master data management, data integration, business intelligence, and data migration initiatives. EDQ provides integrated data quality in customer relationship management and other applications.

EDQ has the following key features:

- Integrated data profiling, auditing, and cleansing and matching
- Browser-based client access
- Ability to handle all types of data (for example, customer, product, asset, financial, and operational)
- Connection to any Java Database Connectivity (JDBC) compliant data sources and targets
- Multi-user project support (Role-based access, issue tracking, process annotation, and version control)
- Representational State Transfer Architecture (ReST) support for designing processes that may be exposed to external applications as a service
- Designed to process large data volumes
- A single repository to hold data along with gathered statistics and project tracking information, with shared access
- Intuitive graphical user interface designed to help you solve real world information quality issues quickly
- Easy, data-led creation and extension of validation and transformation rules
- Fully extensible architecture allowing the insertion of any required custom processing

---

**Note:** For more information on EDQ, see [Oracle Enterprise Data Quality Documentation](#).

---

## EDQ Configuration Process Flow

The following image shows the EDQ configuration process flow:

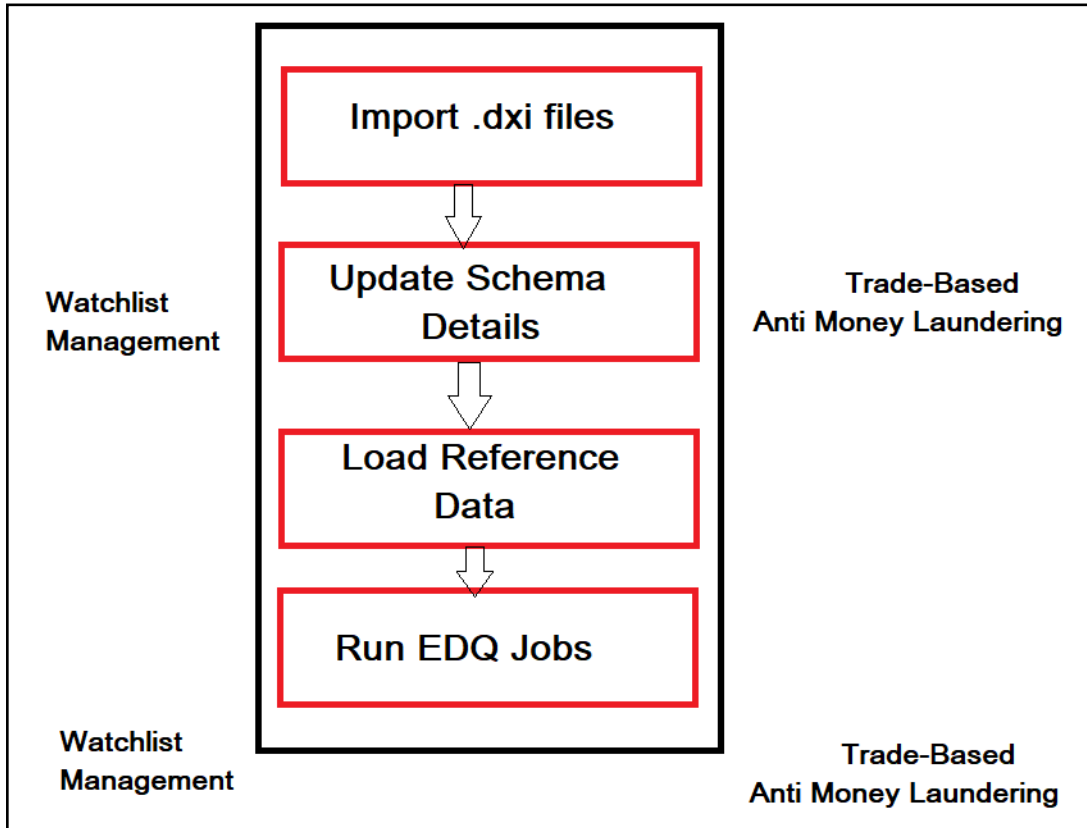


Figure 56. EDQ Configuration Process Flow

To configure the EDQ, follow these steps:

1. Import the TBAML 8.0.6.0.0.dxi/ Watchlist\_Management-8.0.6.0.1.dxi files from the <FIC\_HOME> path.



2. Enter the organization-specific Atomic schema details as shown below:

The screenshot shows a dialog box titled "Edit Data Store" with a close button (X) in the top right corner. Inside the dialog, there is a section titled "Oracle Configuration" containing several input fields and a dropdown menu:

- Database host: whf00bls.in.oracle.com
- Port: 1521
- Database name: ORCLDB
- Name type: SID (dropdown menu)
- User name: tfit\_atomic
- Password: masked with seven dots
- Schema: (empty field)

Below the input fields, there is a note: "The schema need not be entered if it is the default for the user". At the bottom right of the dialog, there are three buttons: "Test...", "O", and "Cancel".

**Figure 57. Updating the Schema Details**

3. Load the Reference data. For more information on Reference data, see [Configuring Port, Goods, Name and Address Screening](#).
4. Run the following jobs under the Watchlist Management project:
  - Analyze Reference data quality
  - Download Prepare & filter export list data
  - Generate StopPhrases
5. Run the TBAML jobs under the TBAML project.
6. Change the EDQ URL in the TBAML\_APPLN\_PARAMS table. This is done the first time you set up the TBAML application as the application needs to know the location of the EDQ. See the *TBAML Installation Guide* for configuration steps.
7. Configure the message and screening parameters, if required.

## General EDQ Configurations

This section consists of the following topics:

- [Importing TBAML Projects](#)
- [Configuring Watch List Management](#)
- [Filtering Watch List Data](#)
- [Port, Goods, Name and Address Screening](#)

## Importing TBAML Projects

See *OFS TBAML Installation Guide* for information about importing TBAML projects.

## Configuring Watch List Management

The TBAML distribution contains two Run Profiles for configuring watch list management and screening:

- `watchlist-management.properties`
- `TBAML.properties`

Run Profiles are optional templates that specify a number of 'override' configuration settings for externalized options when a Job is run. They offer a convenient way of saving and reusing a number of configuration overrides, rather than specifying each override as a separate argument.

Run Profiles may be used when running jobs either from the Command Line Interface, using the 'runopsjob' command, or in the Server Console UI.

The `watchlist-management.properties` Run Profile controls the following:

- Which watch lists are downloaded, and the configuration of the download process
- Whether filtering is applied to the watch lists
- Whether Data Quality Analysis is applied to the watch lists

Additionally, the `TBAML.properties` Run Profile controls the following:

- Real-Time and Batch Screening set up
- Screening reference ID prefixes and suffixes
- Watch list routing
- Configuration of match rules
- Port standardization
- Port Screening
- Goods Screening
- Name and Address Screening

This section consists of the following topics:

- [Preparing Watch List Data](#)
- [Setting Up Private Watch List](#)
- [Showing Watch List Staged Data/Snapshots in the Server Console UI](#)
- [Configuring Match Rules](#)
- [Configuring a Job](#)

## Preparing Watch List Data

TBAML is pre-configured to handle reference data from the following sources:

- HM Treasury
- OFAC
- EU consolidated list
- UN consolidated list
- World-Check
- Dow Jones Watchlist
- Dow Jones Anti-Corruption List
- Accuity Reference Data

For information on the watch lists, see *Appendix A, “Watch Lists,”*.

## Setting Up Private Watch List

TBAML is pre-configured to work with a number of commercially-available and government-provided watch lists. However, you can also screen against your own private watch lists. On installation, screening is configured to run against a sample private watch list with minimal additional configuration, allowing the installation to be validated quickly. The sample private watch list is provided in two files - `privateindividuals.csv` and `privateentities.csv` - in the `config/landingarea/Private` folder.

## The OEDQ Config Folder

Your OEDQ instance's config folder might not be named 'config'. The choice of the config folder's name is made when OEDQ is installed - in some cases a name is automatically allocated. OEDQ release 11g and later has both a 'base' and a 'local' config folder. The base config folder is often called 'oedqhome', and the local config folder is often called 'oedqlocalhome'. In some cases, dots or underscores may be inserted into these names (for example: 'oedq\_local\_home'). Whenever you see a file path in this document that begins with config, this always refers to your OEDQ instance's local config folder.

The first step in screening against your own private watch list is to replace the data in the supplied files with your own data. To do this:

1. Transform your private watch list data into the format specified by the Private List Interface. For more information on Private Watch Lists, see [PLI Reference Data](#).
2. Replace the data in the `privateindividuals.csv` and `privateentities.csv` files with your transformed private watch list data.

---

**Note:** The files must be saved in UTF-8 format.

---

**Note:** To screen against multiple private watch lists, consolidate them into the files: `privateindividuals.csv` and `privateentities.csv`. These two files can also be used to hold data from external watch lists that TBAML is not pre-configured to work with.

---

The second and final step is to enable the staging and preparation of the private watch list in the `watchlist-management.properties` Run Profile. To stage your private watch list set the following value to **Y**:

```
phase.PRIV\ -\ Stage\ reference\ lists.enabled
```

Once you have done this, set the following value to **Y** to prepare the private watch list without filtering:

```
phase.PRIV\ -\ Prepare\ without\ filtering.enabled
```

Or set both of the following values to **Y** to prepare the private watch list with filtering:

```
phase.PRIV\ -\ Prepare\ with\ filtering\ (Part\ 1).enabled
```

```
phase.PRIV\ -\ Prepare\ with\ filtering\ (Part\ 2).enabled
```

## Showing Watch List Staged Data/Snapshots in the Server Console UI

Certain types of staged data and snapshots are hidden in the Server Console UI by default. These are:

- Watch list snapshots
- Intermediate filtered watch list staged data
- Centralized Reference Data staged data/snapshots

To display this data, set the corresponding visibility property value(s) in the relevant Run Profile(s) to **Y**.

For example, to make all HM Treasury watch list snapshots generated during Watchlist Management visible, set the following properties in the `watchlist-management.properties` Run Profile:

- `stageddata.ACY\ Sources.visible = Y`
- `stageddata.ACY_All.visible = Y`
- `stageddata.ACY_Sources.visible = Y`

## Configuring Match Rules

Match rules - and also match clusters - can be configured and controlled by adding a property to the `TBAML.properties` Run Profile.

For example, to disable the port screening for batch, add the following property to the Run Profile:

```
phase.Port\ Screening.enabled = N
```

---

**Note:** Capitalization must be respected and characters must be escaped as required.

---

The `*` character denotes a wildcard, and therefore specifies that this rule applies to all phases and all processes. For example, if disabling the rule for Batch screening only, the property would read:

```
phase.Batch\ screening.process.*.[I0100]\ Exact\ name\ only.san_rule_enabled = false
```

---

**Note:** For further details on tuning Match rules, please refer to the *Oracle Financial Services Transaction Filtering Matching Guide*.

---

## Configuring a Job

A job has to be configured in the `.properties` file and on the Admin screen to enable or disable the webservice. For more information, see *Setting Filtering Options in the Run Profiles*.

## Filtering Watch List Data

### Enabling Watch List Filtering

Watch list data is filtered either during List Management, Screening, or both.

To enable filtering for a specific watch list, set the **Prepare Filtering** phase(s) in the appropriate Run Profile to **Y**, and the **Prepare Without Filtering** phase(s) to **N**.

### Configuring Watch List Filtering

Watch list filtering is controlled by configuring reference data in the Watch list projects.

**Note:** Once data is filtered out, it is not possible to filter it back in. For example, if all entities are filtered out in Watch List Management, even if the TBAML project is configured to include entities, they will not show up in results data.

The top level of filtering is controlled by editing the **Filter - Settings** reference data:

List Key	List Sub Key	List/sub-lis...	Individuals...	Entities (Pr...	Vessels (P...	All origins ...	All origin r...	All origin s...	All name ty...
ACY	ACY-SAN	Y	Y	Y	Y	Y	Y	Y	Y
ACY	ACY-PEP	Y	Y	Y	Y	Y	Y	Y	Y
ACY	ACY-EDD	Y	Y	Y	Y	Y	Y	Y	Y
HMT	HMT-CONS	Y	Y	Y	Y	Y	Y	Y	Y
HMT	HMT-IB	Y	Y	Y	Y	Y	Y	Y	Y
EU	EU	Y	Y	Y	Y	Y	Y	Y	Y
DJW	DJW-SAN	Y	Y	Y	Y	Y	Y	Y	Y
DJW	DJW-PEP	Y	Y	Y	Y	Y	Y	Y	Y
DJW	DJW-EDD	Y	Y	Y	Y	Y	Y	Y	Y
OFAC	OFAC-SDN	Y	Y	Y	Y	Y	Y	Y	Y
OFAC	OFAC-NS-PLCY	Y	Y	Y	Y	Y	Y	Y	Y
UN	UN-ALQ	Y	Y	Y	Y	Y	Y	Y	Y
UN	UN-TAL	Y	Y	Y	Y	Y	Y	Y	Y
WC	WC-SAN	Y	Y	Y	Y	Y	Y	Y	Y
WC	WC-PEP	Y	Y	Y	Y	Y	Y	Y	Y
WC	WC-EDD	Y	Y	Y	Y	Y	Y	Y	Y
PRIV		Y	Y	Y	Y	Y	Y	Y	Y
DJAC	DJAC-SAN	Y	Y	Y	Y	Y	Y	Y	Y
DJAC	DJAC-PEP	Y	Y	Y	Y	Y	Y	Y	Y
DJAC	DJAC-EDD	Y	Y	Y	Y	Y	Y	Y	Y

**Table 55. Filter Settings**

All the reference data filters are set to **Y** by default, except Linked Profiles which is set to **N**. Unless these settings are changed, no actual filtering is performed on watch list data.

**Note:** In the **Filter - Settings** reference data, a value of **Y** indicates that all records should be included - in other words, no filter should be applied.

Broadly speaking, watch list filtering falls into four categories:

- By list and list sub key

- By list record origin characteristics
- By list profile record characteristics
- By linked profiles

### Primary and Secondary Filtering, and Linked Records

- Primary filtering - These filters are used to return all profiles that match the criteria specified.
- Linked Profiles - If this value is set to Y, then all profiles linked to those captured by Primary filters are also captured; an example of use is a filter configured to capture all Sanctions and their related PEPs.
- Secondary filtering - These filters are applied to further filter any linked profiles that are returned.

---

**Note:** Only the World-Check and DJW watch lists can provide Linked Profiles.

---

### Setting Multiple Values for Primary and Secondary Filters

The following filter options require further configuration in additional reference data:

- Origins
- Origin Regions
- Origin Statuses
- Primary and Secondary Name Qualities
- Primary and Secondary Name Types
- Primary and Secondary PEP Classifications

To filter using one or more of these options, set the relevant value in the Filter - Settings reference data to **N**, and then make further changes to the corresponding reference data.

---

**Note:** The effect of setting a value in the **Filter - Settings** reference data to **N** is that only records that match values set in the corresponding reference data will be included. For example, if you set the value of **All name qualities (Primary)?** to **N** in **Filter - Settings**, then, in the **Filter - Primary Name Qualities** reference data you could determine which name qualities should be included for each watch list. For instance, if you include a row for High quality names in the EU watch list, but you do not include rows for medium and low quality names for this watch list, then only records with high quality names will be included for this watch list.

---

Some of these reference data sets are pre-populated with rows, to be edited or removed as required. These rows contain data (generally, but not always) supplied by each watch list provider, and are all contained within the Watchlist Management project.

For example, to view all possible keywords for World-Check data, open the **WC Keyword** reference data in the Watchlist Management project. See the following example for further details.

### Filtering World Check Data

This example describes configuring filtering on the World-Check Sanctions list in the Watchlist Management project, and setting further filters in the TBAML project.

- Enabling filtering in the Run Profiles

- Configuring the Primary filters in the Watchlist Management project to return only active records for sanctioned individuals (not entities) originating from the EU list
- Enabling the filtering of Linked Profiles in the Watchlist Management project
- Configuring the Secondary filters in the Transaction Filtering project to further filter out all Linked Profiles of deceased individuals

## Setting Filtering Options in the Run Profiles

In the `watchlist-management.properties` Run Profile, set the World-Check filtering phases as follows:

- `phase.WC\ -\ Prepare\ without\ filtering.enabled = N`
- `phase.WC\ -\ Prepare\ with\ filtering\ (Part\ 1).enabled = Y`
- `phase.WC\ -\ Prepare\ with\ filtering\ (Part\ 2).enabled = Y`

In the `TBAML.properties` Run Profile, set the World-Check filtering phases as follows:

- `phase.WC\ -\ Load\ without\ filtering.enabled = N`
- `phase.WC\ -\ Load\ with\ filtering\ (Part\ 1).enabled = Y`
- `phase.WC\ -\ Load\ with\ filtering\ (Part\ 2).enabled = Y`

## Setting Primary Filters and Linked Profiles in the Watchlist Management project

To set primary filters, follow these steps:

1. In Director, open the Watchlist Management project and expand the Reference Data node.
2. Locate the **Filter - Settings** reference data, and double-click to open it.
3. Ensure the **List/sub-list (Primary)?** value in the **WC-SAN** row is set to **Y**.
4. Set the **Entities (Primary)?** value in the **WC-SAN** row to **N**.
5. Set the **Inactive (Primary)?** value in the **WC-SAN** row to **N**.
6. Set the **All Origins (Primary)?** value in the **WC-SAN** row to **N**.
7. Ensure all other values in the **WC-SAN** row are set to **Y**.
8. Click **OK** to close the reference data and save changes.
9. Locate the **Filter - Origins** reference data and double-click to open it.
10. Add a new row with the following values:
  - List Key - WC
  - List Sub Key - WC-SAN
  - Origin - EU
11. Change the **Linked Profiles?** value in the **WC-SAN** row to **Y**.
12. Click **OK** to close the Filter Settings reference data and save changes.

## Setting Secondary Filters in the TBAML project

To set secondary filters, follow these steps:

1. Open the TBAML project, and expand the reference data link.
2. Locate the **Filter - Settings** reference data file, and double-click to open it.
3. Set the **Deceased (Secondary)?** value in the **WC-SAN** row to **N**.
4. Click **OK** to close the reference data and save changes.

## Screening All Data Using Sanctions Rules

By default, watch list records are routed to the different screening processes depending on their record type, that is, SAN, PEP or EDD. This allows different rules, and hence different levels of rigor, to be applied to the list data according to risk appetite.

However, if you want to use the same screening logic for all list records, and do not want the overhead of maintaining separate rule sets, the system can be configured to reroute all list records to the SAN screening processes. To do this, set the **phase.\*.process.\*.Screen\ all\ as\ SAN?** value in the `watchlist-screening.properties` Run Profile to **Y**.

## Port, Goods, Name and Address Screening

This section consists of the following topics:

- [Configuring Port, Goods, Name and Address Screening](#)
- [Extending Prohibition Screening](#)

## Configuring Port, Goods, Name and Address Screening

This section consists of the following topics:

- [Bad BICs Reference Data](#)
- [Blacklisted Cities Reference Data](#)
- [Blacklisted Countries Reference Data](#)
- [Stop Keywords Reference Data](#)
- [Goods Prohibition Reference Data](#)
- [Ports Prohibition Reference Data](#)

### **Bad BICs Reference Data**

The following columns are available in the template for BICs:

- **Record ID:** This column displays the record serial number for the blacklisted BIC. The record ID is unique for every BIC.
- **BIC:** This column displays the name of the BIC.
- **Details of BIC:** This column displays the details of the BIC.
- **Data Source:** This column displays the source of the data for the BIC.
- **Risk Score:** This column displays the risk score for the BIC.



### Sample Data for Sanctioned BICs

The following table provides examples based on BICs:

**Table 56. Sample Data for BICs**

Record ID	BIC	Details of BIC	Data Source	Risk Score
1	SIIBSYDA	NA	OFAC (Office of Foreign Assets Control)	85
2	FTBDKPPY	NA	OFAC (Office of Foreign Assets Control)	90
3	DCBKKPPY	NA	OFAC (Office of Foreign Assets Control)	85
4	ROSYRU2P	NA	OFAC (Office of Foreign Assets Control)	90
5	INAKRU41	NA	OFAC (Office of Foreign Assets Control)	90
6	SBBARUMM	NA	OFAC (Office of Foreign Assets Control)	90

### Blacklisted Cities Reference Data

The following columns are available in the template for blacklisted cities:

- Record ID: This column displays the record serial number for the blacklisted city. The record ID is unique for every city.
- Country: This column displays the name of the country of the blacklisted city.
- City: This column displays the name of the blacklisted city.
- ISO City Code: This column displays the ISO code of the blacklisted city.
- Data Source: This column displays the source of the data for the blacklisted city.
- Risk Score: This column displays the risk score for the blacklisted city.

### Sample Data for Sanctioned Cities

The following table provides examples for blacklisted cities:

**Table 57. Sample Data for Blacklisted Cities**

Record ID	Country	City	ISO City Code	Data Source	Risk Score
1	IRAQ	ARBIL	ABL	OFAC (Office of Foreign Assets Control)	90

Record ID	Country	City	ISO City Code	Data Source	Risk Score
2	IRAQ	ABU AL FULUS	ALF	OFAC (Office of Foreign Assets Control)	90
3	IRAQ	AMARA (AL-AMARA H)	AMA	OFAC (Office of Foreign Assets Control)	85
4	IRAQ	ARAK	ARK	OFAC (Office of Foreign Assets Control)	90

### **Blacklisted Countries Reference Data**

The following columns are available in the template for blacklisted countries:

- Record ID: This column displays the record serial number for the blacklisted country. The record ID is unique for every country.
- Country: This column displays the name of the blacklisted country.
- ISO Country Code: This column displays the ISO code of the blacklisted country.
- Country Synonyms: This column displays the synonyms of the blacklisted country.
- Data Source: This column displays the source of the data for the blacklisted country.
- Risk Score: This column displays the risk score for the blacklisted country.

### **Sample Data for Sanctioned Countries**

The following table provides sample data for blacklisted countries:

**Table 58. Sample Data for Blacklisted Countries**

Record ID	Country	ISO Country Code	Country Synonyms	Data Source	Risk Score
1	IRAQ	IQ	IRAK, REPUBLIC OF IRAQ, AL JUMHURIYAH AL IRAQIYAH, AL IRAQ	OFAC (Office of Foreign Assets Control)	90
2	DEMOCRATIC REPUBLIC OF THE CONGO	CD	CONGO, THE DEMOCRATIC REPUBLIC OF THE	OFAC (Office of Foreign Assets Control)	90
3	AFGHANISTAN	AF	NA	ITAR (International Traffic in Arms Regulations)	85
4	ZIMBABWE	ZW	NA	ITAR (International Traffic in Arms Regulations)	90

Record ID	Country	ISO Country Code	Country Synonyms	Data Source	Risk Score
5	CENTRAL AFRICAN REPUBLIC	CF	NA	EAR (Export Administration Regulations)	85
6	BELARUS	BY	NA	EAR (Export Administration Regulations)	80

### **Stop Keywords Reference Data**

The following columns are available in the template for keywords:

- Record ID: This column displays the record serial number for the keyword.
- Stop keyword: This column displays the keyword.
- Risk Score: This column displays the risk score for the keyword.

### **Sample Data for Sanctioned Stop Keywords**

The following table provides examples based on keywords:

**Table 59. Sample Data for Stop Keywords**

Record ID	Stop KeyWords	Risk Score
1	EXPLOSIVE	80
2	DIAMOND	90
3	TERROR	80
4	TERRORIST	85
5	ARMS	80
6	NUCLEAR	90

### **Goods Prohibition Reference Data**

The following columns are available in the template for prohibited goods:

- Record ID: This column displays the record serial number for the prohibited good. The record ID is unique for every good.
- Good Code: This column displays the code of the prohibited good.
- Good Name: This column displays the name of the prohibited good.
- Good Description: This column displays the description of the prohibited good.
- Good Synonym: This column displays other names which may be used to describe the prohibited good.
- Import: This column displays the County ISO code where the goods are imported from
- Export : This column displays the County ISO code where the goods are exported from

*Sample Data for Prohibited Goods*

The following table provides sample data for prohibited goods.

**Table 60. Sample Data for Prohibited Goods**

Record ID	Good Code	Good Name	Good Description	Good Synonym	Import	Export
1	0207 43 00	Ammonium nitrate	Ammonium nitricum	Nitram, Ammonium saltpeter;	PK	AZ
2	0208 90 10	Ivory	Ornamental Jewlery		PK	TD
3	0209 10 00	Ceramic tableware	NA	terracotta, kerameikos	US	EG
4	3057100	Shark fins	NA		CA	US
5	4302 19 40	Tiger-Cat skins	Pelts		IN	RU

**Ports Prohibition Reference Data**

The following columns are available in the template for prohibited ports:

- Record ID: This column displays the record serial number for the prohibited port. The record ID is unique for every port.
- Country: This column displays the Country ISO of the country where the prohibited port is located.
- Port Name: This column displays the name of the prohibited port.
- Port Code: This column displays the code of the prohibited port.
- Port Synonyms: This column displays the synonym of the prohibited port.

*Sample Data for Prohibited Ports*

The following table provides sample data for prohibited ports:

**Table 61. Sample Data for Prohibited Ports**

Record ID	Country	Port Name	Port Code	Port Synonyms
1	IRAN, ISLAMIC REPUBLIC OF	KHORRAMS HAHR	IR KHO	KHORRAMSHAHR Port
2	RUSSIA	Sevastopol	SMTTP	Sebastopol,Port of Sevastopol
3	New Zealand	Dunedin	NZ ORR	Otago Harbour
4	New Zealand	Ravensbourne	NZ ORR	Otago Harbour

### ***Extending Prohibition Screening***

TBAML, as delivered, allows for prohibition screening against Nationality and Residency for Individuals and [country of] Operation and [country of] Registration for Entities. Additional prohibition types can be added as follows:

- Create new entries in the prohibition reference data with a new Prohibition Type name, for example "Employment Country".
- [Batch screening only] Extend the customer data preparation process to create a new attribute, for example `dnEmploymentCountryCode`.
- Edit the appropriate screening process(es), to create the necessary match rules and clusters for the new attribute.

For more information about Watch List matching, see *Oracle Financial Services Transaction Filtering Matching Guide*.



You can add new SWIFT message types and configure the messages by uploading a JSON for a given message type followed by few configurations using admin UI screen. A new JSON is required for each new SWIFT message type and for editing any existing message type. JSON follows SWIFT message standards given in SWIFT document. JSON file should be .txt or .json extensions only.

This chapter provides information on how to create a JSON for SWIFT messages with sequences and for SWIFT messages without sequences. This chapter covers the following topics:

- [Structure of a JSON](#)
- [Creating JSON for SWIFT Messages with Sequences](#)
- [Creating JSON for SWIFT Messages without Sequences](#)
- [Creating JSON for SWIFT messages with the List of Values \(LOV\) Attribute](#)

---

**Note:** For information on how to upload a JSON, see [Adding or Updating a New Message Type](#).

---

For more information about using the SWIFT Parser, refer to the *Oracle Financial Services SWIFT Parser User Guide*.

## Structure of a JSON

An example of a JSON is shown below:

```
{
  "message": [
    {
      "attr": {
        "id": "t1",
        "field": "Basic Header Block",
        "status": "",
        "fieldName": "",
        "expression": "",
        "editable": "N"
      },
      "children": [
        {
          "attr": {
            "id": "t1:1",
            "field": "",
            "status": "",
            "fieldName": "Block Identifier",
            "expression": "",
            "editable": "Y",
            "size": "1"
          }
        }
      ]
    }
  ]
}
```

Each JSON should start with a "message" element. Every "message" element is a list of "attr" elements.

Each field/tag in the JSON should be represented by "attr". Every "attr" element in the JSON can have the below mentioned properties.

- id: A unique value that identifies each element
- field: Name of the element as per the Swift document, used at parent level.
- status: It can hold either "M" or "O" ("M" - mandatory ,"O" - optional)



- `fieldName`: Name of the element as per the Swift document, used at child level.
- `expression`: Swift expression as per the Swift document
- `editable`: It can hold either "Y" or "N" ("Y" - editable in Admin UI,"N" - non editable in Admin UI)
- `size`: This property is applicable for Swift Block 1, Swift Block 2 where data is only positional i.e there is no swift expression for the element

For example:

- An *attr* element which represents the Swift Block Name is shown below:

```
{
"attr":
{
"field":"Basic Header Block",
"status":"",
"fieldName":"",
"expression":"",
"editable":"N"
}
}
```

- An *attr* element which represents the Swift Block Tag with a *size* property is shown below:

---

**Note:** The *expression* property should be blank for elements that are positional.

---

```
{
"attr":
{
"field":"",
"status":"",
"fieldName":"Block Identifier",
"expression":"",
"editable":"Y",
"size":"1"
}
}
```

An *attr* element which represents the Swift Block Tag with an *expression* property is shown below:

```
{
"attr":
{
"id":"t4:1:2:5:2:1",
"field":"",
"status":"",
"fieldName":"Party Identifier",
"expression":"35x",
"editable":"Y"
}
}
```

Each *attr* element in the JSON can have one or more child attributes. *Children* is used as a notation to identify the children of a particular *attr* element.

```
{
"attr": {
  "id": "t1",
  "field": "Basic Header Block",
  "status": "",
  "fieldName": "",
  "expression": "",
  "editable": "N"
},
"children": [
  {
    "attr": {
      "id": "t1:1",
      "field": "",
      "status": "",
      "fieldName": "Block Identifier",
      "expression": "",
      "editable": "Y",
      "size": "1"
    }
  },
  .....
}
```

```
]
}
```

## Creating JSON for SWIFT Messages with Sequences

To create a JSON, follow these steps:

1. Create Message Elements.
2. Configure SWIFT Message Blocks

### Creating Message Elements

To create a message element, use the sample code below:

```
{
  "message": [
    {
      Requires tags ...
    }
  ]
}
```

### Configuring SWIFT Message Blocks

To configure a SWIFT message block, follow these steps:

1. Configure the Basic Header Block. See *Configuring the Basic Header Block*.
2. Configure the Application Header Block. See *Configuring the Application Header Block*.
3. Configure the User Header Block. See *Configuring the User Header Block*.
4. Configure the Text Block. See *Configuring the Text Block*.
5. Configure the Trailer Block. See *Configuring the Trailer Block*.

### Configuring the Basic Header Block

To configure a User Header Block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *Basic Header Block* and *editable* property as *N*.
2. Create a *children* element with the required *attr* elements that should be part of *Basic Header Block*.

```
{
  "attr": {
    "id": "t1",
    "field": "Basic Header Block",
```

```
    "status": "",
    "fieldName": "",
    "expression": "",
    "editable": "N"
  },
  "children": [
    {
      "attr": {
        "id": "t1:1",
        "field": "",
        "status": "",
        "fieldName": "Block Identifier",
        "expression": "",
        "editable": "Y",
        "size": "1"
      }
    },
    .....
  ]
}
```

### Configuring the Application Header Block

To configure an Application Header Block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *Application Header Block* and *editable* property as *N*.
2. Create a *children* element with two *attr* elements with *fieldName* property as *Application Header - Input* and *Application Header - Output* and *editable* property as *N*.
3. Create a *children* element with the required *attr* elements that should be part of *Application Header - Input* and *Application Header - Output*.

```
{
  "attr": {
    "id": "t2",
    "field": "Application Header Block",
    "status": "",
    "fieldName": "",
    "expression": "",
    "editable": "N"
  },

```

```
"children": [  
  {  
    "attr": {  
      "id": "t2:1",  
      "field": "Application Header - Input",  
      "status": "",  
      "fieldName": "",  
      "expression": "",  
      "editable": "N"  
    },  
    "children": [  
      {  
        "attr": {  
          "id": "t2:1:1",  
          "field": "",  
          "status": "",  
          "fieldName": "Block Identifier",  
          "expression": "",  
          "editable": "Y",  
          "size": "1"  
        }  
      },  
      .....  
    ]  
  },  
  {  
    "attr": {  
      "id": "t2:2",  
      "field": "Application Header - Output",  
      "status": "",  
      "fieldName": "",  
      "expression": "",  
      "editable": "N"  
    },  
    "children": [  
      {  
        "attr": {
```

```
        "id": "t2:2:1",
        "field": "",
        "status": "",
        "fieldName": "Block Identifier",
        "expression": "",
        "editable": "Y",
        "size": "1"
    }
},
.....
]
}
]
}
```

### Configuring the User Header Block

To configure a User Header Block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *User Header Block* and *editable* property as *N*.
2. Create a *children* element with the required *attr* elements that should be part of *User Header Block*.

```
{
  "attr": {
    "id": "t3",
    "field": "User Header Block",
    "status": "",
    "fieldName": "",
    "expression": "",
    "editable": "N"
  },
  "children": [
    {
      "attr": {
        "id": "t3:1",
        "field": "",
        "status": "",
        "fieldName": "Block Identifier",
        "expression": "",
        "editable": "Y"
      }
    }
  ]
}
```

```

    },
    .....
  ]
}

```

### Configuring the Text Block

To configure a Text Block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *Text Block* and *editable* property as *N*.
2. Create a *children* element with *attr* element having *fieldName* property as *Sequences* and *editable* property as *N*.
3. Create a *children* element with the required *attr* elements that represent individual Sequence (that is, Sequence <X>, where X can be A, B, or C) that should be part of Sequences.

```

{
  "attr": {
    "id": "t4",
    "field": "Text Block",
    "status": "",
    "fieldName": "",
    "expression": "",
    "editable": "N"
  },
  "children": [
    {
      "attr": {
        "id": "t4:1",
        "field": "Sequences",
        "status": "",
        "fieldName": "",
        "expression": "",
        "editable": "N"
      },
      "children": [
        {
          "attr": {
            "id": "t4:1:1",
            "field": "Sequence A",
            "status": "",
            "fieldName": "",

```

```
    "expression": "",
    "editable": "N"
  },
  "children": [
    {
      "attr": {
        "id": "t4:1:1:1",
        "field": "20",
        "status": "M",
        "fieldName": "Sender's Reference",
        "expression": "16x",
        "editable": "Y"
      }
    },
    .....
  ],
  {
    "attr": {
      "id": "t4:1:2",
      "field": "Sequence B",
      "status": "",
      "fieldName": "",
      "expression": "",
      "editable": "N"
    },
    "children": [
      {
        "attr": {
          "id": "t4:1:2:1",
          "field": "21",
          "status": "M",
          "fieldName": "Transaction Reference",
          "expression": "16x",
          "editable": "Y"
        }
      },
      .....
    ]
  }
}
```



```

        }
      ]
    }
  ]
}

```

### Configuring the Trailer Block

To configure the Trailer Block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *Trailer Block* and *editable* property as *N*.
2. Create a *children* element with the required *attr* elements that should be part of *Trailer Block*.

```

{
  "attr": {
    "id": "t5",
    "field": "Trailer Block",
    "status": "",
    "fieldName": "",
    "expression": "",
    "editable": "N"
  },
  "children": [
    {
      "attr": {
        "id": "t5:1",
        "field": "CHK",
        "status": "M",
        "fieldName": "Checksum",
        "expression": "",
        "editable": "Y"
      }
    }
  ],
  .....
}

```

## Example of MT101 with Sequences

To see examples of MT101 with sequences, see [MOS Document 2329509.1](#).

## Creating JSON for SWIFT Messages without Sequences

To create a JSON, follow these steps:

1. Create Message Elements.
2. Configure SWIFT Message Blocks

### Creating Message Elements

To create a message element, use the sample code below:

```
{
  "message": [
    {
      Requires tags ...
    }
  ]
}
```

### Configuring SWIFT Message Blocks

To configure a SWIFT message block, follow these steps:

1. Configure the Basic Header Block. See [Configuring the Basic Header Block](#).
2. Configure the Application Header Block. See [Configuring the Application Header Block](#).
3. Configure the User Header Block. See [Configuring the User Header Block](#).
4. Configure the Text Block. See [Configuring the Text Block](#).
5. Configure the Trailer Block. See [Configuring the Trailer Block](#).

### Configuring the Text Block

To configure the text block, follow these steps:

1. Create an *attr* element node with *fieldName* property as *Text Block* and *editable* property as *N*.
2. Create a *children* element with the required *attr* elements that should be part of *Text Block*.

```
{
  "attr": {
    "id": "t4",
    "field": "Text Block",
    "status": "",

```

```
"fieldName": "",
"expression": "",
"editable": "N"
},
"children": [
  {
    "attr": {
      "id": "t4:1",
      "field": "20",
      "status": "M",
      "fieldName": "Sender's Reference",
      "expression": "16x",
      "editable": "Y"
    }
  },
  .....
]
```

### Example of MT101 without Sequences

To see examples of MT101 with sequences, see [MOS Document 2329509.1](#).

## ***Creating JSON for SWIFT messages with the List of Values (LOV) Attribute***

According to SWIFT standards, if there is a tag which contains predefined codes, then we must prepare a List of Values (LOV) attribute for the SWIFT tag. An example of a JSON with an LOV attribute is shown below:

```
{
  "attr": {
    "id": "t4:14:2:2",
    "field": "",
    "status": "",
    "fieldName": "Code",
    "expression": "14x",
    "regex": "",
    "editable": "Y",
    "lov": [
      "BY ACCEPTANCE",
      "BY DEF PAYMENT",
```

```
        "BY MIXED PYMT",  
        "BY NEGOTIATION",  
        "BY PAYMENT"  
    ]  
}  
}
```

This appendix describes the mechanism that TBAML uses when logging system messages.

- [About System Log Messages](#)
- [Message Template Repository](#)
- [Logging Levels](#)
- [Logging Message Libraries](#)
- [Logging Configuration File](#)

## 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; section *Log File Sizes* on page 251 provides instructions. The default value for this property is 10 MB. The maximum file size should not exceed two gigabytes (2000000000 bytes).

---

## 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.

## **Logging Levels**

Table 62 outlines the logging levels that the Common Logging component supports.

**Table 62. Logging Levels**

<b>Severity (Log Level)</b>	<b>Usage</b>
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.

## **Logging Message Libraries**

Some Oracle subsystems produce log output files in default locations. The following sections describe these subsystems.

### **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`

### **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.

## 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.

## 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.

## 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.

## Alert Management

The following logs contain the message library for the Alert Management application:

### Web Server Logs

The following file is the message library for the Web server logs:

```
$FIC_WEB_HOME/logs/UMMService.log
```

### Application Server logs

The following file is the message library for the Application Server logs:

```
$FIC_APP_HOME/common/ficserver/logs/RevAppserver.log
```

### 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.

## Ingestion Manager

The following file is the message library for the Ingestion Manager:

```
<OFSAAI Installed Directory>/ingestion_manager/config/message.dat
```

## 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 63: Logging Configuration Files**

Subsystem	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: <ul style="list-style-type: none"> <li>● DEBUG</li> <li>● INFO</li> <li>● WARN</li> <li>● SEVERE</li> <li>● FATAL</li> </ul>
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: <ul style="list-style-type: none"> <li>● DEBUG</li> <li>● INFO</li> <li>● WARN</li> <li>● SEVERE</li> <li>● FATAL</li> </ul>
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

## 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" filePattern="@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" target="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 58. Sample Logging Configuration File

## Configurable Logging Properties

Table 64 identifies the configurable properties for logging in an Oracle client's environment.

**Table 64. 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 <i>Short Introduction to log4j</i> guide ( <a href="http://logging.apache.org/log4j/docs/manual.html">http://logging.apache.org/log4j/docs/manual.html</a> ) 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.
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.<category_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 <code>categories.cfg</code> 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; <i>false</i> 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. <b>Note:</b> Setting value to true (synchronous) may have performance impact
log.warning	true	Indicates enabling of warning logging; <i>false</i> indicates that warning logging is not enabled.

**Table 64. Configurable Parameters for Common Logging (Continued)**

Property	Sample Value	Description
<code>log.warning.synchronous</code>	false	Indicates that warning level logging should happen asynchronously; true indicates warning level logging should happen synchronously. <b>Note:</b> Setting value to true (synchronous) may have performance impact
<code>log.notice</code>	true	Indicates enabling of notice logging; <i>false</i> indicates that notice logging is not enabled.
<code>log.notice.synchronous</code>	false	Indicates that notice level logging should happen asynchronously; true indicates notice level logging should happen synchronously. <b>Note:</b> Setting value to true (synchronous) may have performance impact
<code>log.diagnostic</code>	false	Indicates that diagnostic logging is not enabled; <i>true</i> indicates enabling of diagnostic logging.
<code>log.diagnostic.synchronous</code>	false	Indicates that diagnostic level logging should happen asynchronously; true indicates diagnostic level logging should happen synchronously. <b>Note:</b> Setting value to true (synchronous) may have performance impact
<code>log.trace</code>	false	Indicates that trace logging is not enabled; <i>true</i> indicates enabling of trace logging.
<code>log.trace.synchronous</code>	true	Indicates that trace level logging should happen asynchronously; true indicates trace level logging should happen synchronously. <b>Note:</b> Setting value to true (synchronous) may have performance impact
<code>log.syslog.hostname</code>	hostname	Indicates the host name of syslog for messages sent to syslog.
<code>log.time.zone</code>	US/Eastern	Indicates the time zone that is used when logging messages.

## 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.



This appendix describes the application of software updates in Oracle Financial Services TBAML:

- [Oracle Software Updates - Hotfix](#)
- [Hotfix Effect on Customization](#)

## **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:

- The 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

## **Hotfix Effect on Customization**

When a hotfix is installed it can affect your customizations on the *User Interface* and *Scenarios*.

### **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.

## **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.

This appendix describes the user administration of Oracle Financial Services TBAML.

- [Managing User Groups and User Roles](#)
- [Managing User Groups](#)
- [Defining User Access Properties and Relationships](#)

## ***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 Group for TBAML is TBAMLADMINISTRATORGRP.

For more information on creating a new user group and mapping it to an existing role, see *Oracle Financial Services Analytical Applications Infrastructure User Guide* in Identity Management section.

**Note:** While creating a new User Group, you can set precedence as 5001 or greater.

## ***Managing User Groups***

The following sections describe how to manage User Groups:

- [Defining User Group Maintenance Details](#)
- [Adding New User Group Details](#)
- [Mapping Users to User Groups](#)
- [Mapping User Group\(s\) to Domain\(s\)](#)
- [Mapping a User to a Single User Group](#)

## ***Defining User Group Maintenance Details***

For more information on defining user group maintenance details, see *Oracle Financial Services Analytical Applications Infrastructure User Guide* in the Identity Management section.

## Adding New User Group Details

For more information on adding new user group details, see *Oracle Financial Services Analytical Applications Infrastructure User Guide* in the Identity Management section.

## Mapping Users to User Groups

---

**Note:** 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 *Oracle Financial Services Analytical Applications Infrastructure User Guide* in the Identity Management section.

## Mapping User Group(s) to Domain(s)

To map user group or groups to domain or domains see *Oracle Financial Services Analytical Applications Infrastructure User Guide* in the Identity Management section.

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, *Configuration Guide*, and refer to section *Working with Alert Action Settings*.

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.

## 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 *Oracle Financial Services Analytical Applications Infrastructure User Guide* in the Identity Management section.

## 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.

## 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**.

### Mapping a Function to a Role

The following list of functions must be mapped to appropriate TBAML 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 65. 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.

## 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 OFSECM 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.

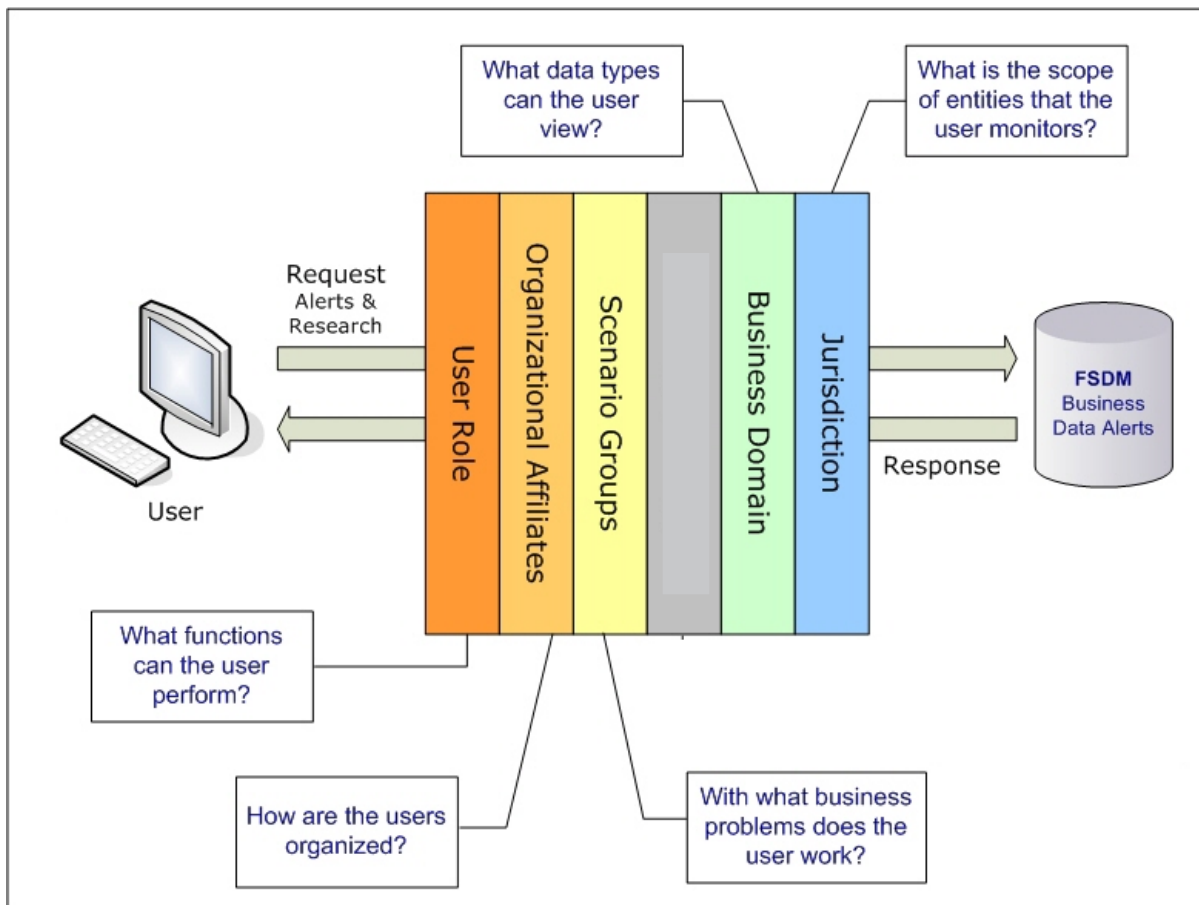


Figure 59. User Authorization Model

The following table provides the relationships between the data points that Figure 3 illustrates.

**Table 66. Relationships between Data Points**

Data Point	Relationship
Organization	Root of a BD 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
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 <code>mantasLineOrgMember</code> )
Users (Admin Tools)	Should be mapped only to mantas Admin Role.
Scenario Group	Associated to 0..n users
	Associated with Scenarios referenced in <code>KDD_SCNRO</code> table.
Business Domains	Associated to 0..n users
	Business domain <i>key</i> must be in the <code>KDD_BUS_DMN</code> table
Jurisdiction	Associated to 0..n users
	Jurisdiction <i>key</i> must exist in the <code>KDD_JRSDCN</code> table



This appendix covers the following topics:

- [FSDF/Hive CSA Ingestion](#)
- [Flat File Ingestion](#)

## CSA Ingestion

This section refers to Common Staging Area (CSA) ingestion and covers the following topics:

- [CSA Datamaps](#)
- [List of Data Quality Group Names and T2T Names](#)

## 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.

---

**Note:** Ensure that you run the Country and Customer data files before you run the other files.

---

**Table 67. CSA Datamaps**

Group	Logical Table Name	
1.	Country	Customer
2.	Account Phone Watch List Account Email Address Insurance Product Insurance Policy Insurance Transaction Insurance Policy Balance Front Office Transaction Account Customer Role Organization Insurance Policy Feature Market Center	Insurance Policy To Customer Market Index Daily Loan Issuer Loan Daily Activity Market Index Online Account Service Team Insurance Seller Service Team Member Insurance Seller To License Customer Credit Rating Customer Identification Document

**Table 67. CSA Datamaps**

Group	Logical Table Name	
3.	Account To Peer Group Account Group Peer Group Security Firm Daily	Market Index Member Security Security Market Daily Security Customer
4.	Account Watch List Entry Loan Product Employee	Front Office Transaction Party Organization Relationship Restriction List Automated Quote
5.	Managed Account Account To Customer Account Group Member Account To Correspondent Account Balance Account Address Customer To Markets Served Customer To Products Offered Customer To Customer Relationship Anticipatory Profile Customer Phone Customer Email Address Customer Country Customer Address Online Account To Account	Controlling Customer Employee To Account Account Position Security Trading Restriction Employee Trading Restriction Employee Phone Employee Email Address Employee Address Security Group Member Security Investment Rating Structured Deal Account Profit And Loss Account Investment Objective Account Position Pair Mutual Fund Breakpoint Market News Event
6.	Borrower Account Restriction Back Office Transaction	Investment Advisor Settlement Instruction Loan Origination Document Print Log

**List of Data Quality Group Names and T2T Names**

The following table provides the FSDM logical table names and T2T names, and the corresponding data quality group names:

**Table 68. Data Quality Group Names and Related T2T Names**

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Account	ACCT	<ul style="list-style-type: none"> <li>● t2t_Account.STG_ANNUITY_CONTRACTS</li> <li>● t2t_Account.STG_CARDS</li> <li>● t2t_Account.STG_CASA</li> <li>● t2t_Account.STG_CORRESPONDENT_ACCOUNT</li> <li>● t2t_Account.STG_MERCHANT_CARDS</li> <li>● t2t_Account.STG_MM_CONTRACTS</li> <li>● t2t_Account.STG_OD_ACCOUNTS</li> <li>● t2t_Account.STG_REPO_CONTRACTS</li> <li>● t2t_Account.STG_RETIREMENT_ACCOUNTS</li> <li>● t2t_Account.STG_SWAPS_CONTRACTS</li> <li>● t2t_Account.STG_TD_CONTRACTS</li> <li>● t2t_Account.STG_TRADING_ACCOUNT</li> <li>● t2t_Account.STG_TRUSTS</li> </ul>
Account	ACCT	<ul style="list-style-type: none"> <li>● t2t_Account.STG_LEASES_CONTRACTS</li> <li>● t2t_Account.STG_LOAN_CONTRACTS</li> </ul>
Account Address	ACCT_ADDR	<ul style="list-style-type: none"> <li>● t2t_AccountAddress</li> </ul>
Account Balance	ACCT_BAL_PO SN_SMRY	<ul style="list-style-type: none"> <li>● t2t_AccountBalance.STG_ANNUITY_CONTRACTS</li> <li>● t2t_AccountBalance.STG_CARDS</li> <li>● t2t_AccountBalance.STG_CASA</li> <li>● t2t_AccountBalance.STG_CORRESPONDENT_ACCOUNT</li> <li>● t2t_AccountBalance.STG_LEASES_CONTRACTS</li> <li>● t2t_AccountBalance.STG_LOAN_CONTRACTS</li> <li>● t2t_AccountBalance.STG_OD_ACCOUNTS</li> <li>● t2t_AccountBalance.STG_RETIREMENT_ACCOUNTS</li> <li>● t2t_AccountBalance.STG_TD_CONTRACTS</li> <li>● t2t_AccountBalance.STG_TRADING_ACCOUNT</li> <li>● t2t_AccountBalance.STG_TRUSTS</li> </ul>
Account Customer Role	CUST_ACCT_ROLE	t2t_AccountCustomerRole
Account Email Address	ACCT_EMAIL_ADDR	t2t_AccountEmailAddress
Account Group	ACCT_GRP	t2t_AccountGroup
Account Group Member	ACCT_RLSHP	t2t_AccountGroupMember

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Account Investment Objective	ACCT_NVSM_T_OBJ	t2t_AccountInvestmentObjective
Account Phone	ACCT_PHON	t2t_AccountPhone
Account Position	ACCT_POSN	t2t_AccountPosition.STG_ACCOUNT_POSITION
Account Position Pair	ACCT_POSN_PAIR	t2t_AccountPositionPair
Account Profit and Loss		t2t_AccountProfitAndLoss
Account Restriction	ACCT_RSTRN	t2t_AccountRestriction
Account to Correspondent	ACCT_INSTN_MAP_STAGE	t2t_AccountToCorrespondent
Account to Customer	CUST_ACCT	t2t_AccountToCustomer
Account To Peer Group	ACCT_PEER_GRP	t2t_AccountToPeerGroup
Anticipatory Profile	NTCPTRY_PROF_L	<ul style="list-style-type: none"> <li>● t2t_AnticipatoryProfile.STG_ACCT_ANTICIPATORY_PROFILE</li> <li>● t2t_AnticipatoryProfile.STG_CUST_ANTICIPATORY_PROFILE</li> </ul>
Back Office Transaction	BACK_OFFICE_TRXN	<ul style="list-style-type: none"> <li>● t2t_BackOfficeTransaction.STG_ANNUITY_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_CARDS_PAYMENT_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_CASA_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_CORRESPONDENT_ACCT_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_LEASES_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_LOAN_CONTRACT_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_MERCHANT_CARDS_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_MM_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_OD_ACCOUNTS_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_REPO_TRANSACTIONS</li> <li>● t2t_BackOfficeTransaction.STG_RETIREMENT_ACCOUNTS_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_SWAP_ACCOUNT_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_TERMDEPOSITS_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_TRADING_ACCOUNT_TXNS</li> <li>● t2t_BackOfficeTransaction.STG_TRUSTS_TXNS</li> </ul>
Borrower	BORROWER	t2t_Borrower
Branch CTR Conductor	BRANCH_CTR_CNDTR	t2t_BranchCTRConductor



Interface File Name	Data Quality Group Name	Corresponding T2T Name
Branch CTR Summary	BRANCH_CTR_SMRY	t2t_BranchCTRSummary
Branch CTR Transaction	BRANCH_CTR_TRXN	t2t_BranchCTRTransaction
Controlling Customer	CNTRL_CUST	t2t_ControllingCustomer
Corporate Action	CORP_ACTN	t2t_CorporateAction
Country	GEOGRAPHY	t2t_Country.STG_COUNTRY_MASTER
Currency Transaction	CURRENCY_TRXN	<ul style="list-style-type: none"> <li>● t2t_CurrencyTransaction.STG_ANNUITY_TXNS</li> <li>● t2t_CurrencyTransaction.STG_CARDS_PAYMENT_TXNS</li> <li>● t2t_CurrencyTransaction.STG_CASA_TXNS</li> <li>● t2t_CurrencyTransaction.STG_LOAN_CONTRACT_TXNS</li> <li>● t2t_CurrencyTransaction.STG_RETIREMENT_ACCOUNTS_TXNS</li> <li>● t2t_CurrencyTransaction.STG_SWAP_ACCOUNT_TXNS</li> <li>● t2t_CurrencyTransaction.STG_TERMDEPOSITS_TXNS</li> <li>● t2t_CurrencyTransaction.STG_TRADING_ACCOUNT_TXNS</li> </ul>
Customer	CUST	t2t_Customer.STG_PARTY_MASTER
Customer Address	CUST_ADDR	t2t_CustomerAddress
Customer Country	CUST_CNTRY	<ul style="list-style-type: none"> <li>● t2t_CustomerCountry</li> <li>● t2t_CustomerCreditRating</li> </ul>
Customer E-mail Address	CUST_EMAIL_ADDR	t2t_CustomerEmailAddress
Customer Identification Document	CUST_ID_DOC	t2t_CustomerIdentificationDocument
Customer Phone	CUST_PHON	t2t_CustomerPhone
Customer Supplemental Attribute	CUST_SUPPLEMENTAL_ATTR	t2t_CustomerSupplementalAttribute
Customer to Customer Relationship	CUST_CUST	t2t_CustomerToCustomerRelationship
Customer to Markets Served	CUST_MKT_SERVED	t2t_CustomerToMarketsServed

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Customer to Products Offered	CUST_PRODU CT	t2t_CustomerToProductsOffered
Employee	EMP	t2t_Employee.STG_EMPLOYEE_MASTER
Employee Address	EMP_ADDR	t2t_EmployeeAddress
Employee Email Address	EMP_EMAIL_A DDR	t2t_EmployeeEmailAddress
Employee Phone	EMP_PHON	t2t_EmployeePhone
Employee to Account	EMP_ACCT	t2t_EmployeeToAccount
Employee Trading Restriction	EMP_SCRTY_R STRN_LIST	t2t_EmployeeTradingRestriction.STG_EMPLOYEE_TRD_RESTRICTION
External Party Stage		t2t_ExternalPartyStage.STG_PARTY_MASTER
Front Office Transaction	FO_TRXN_STA GE	<ul style="list-style-type: none"> <li>● t2t_FrontOfficeTransaction.STG_ANNUITY_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_CARDS_PAYMENT_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_CASA_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_CORRESPONDENT_ACCT_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_LEASES_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_LOAN_CONTRACT_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_MERCHANT_CARDS_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_MM_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_OD_ACCOUNTS_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_REPO_TRANSACTIONS</li> <li>● t2t_FrontOfficeTransaction.STG_RETIREMENT_ACCOUNTS_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_SWAP_ACCOUNT_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_TERMDEPOSITS_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_TRADING_ACCOUNT_TXNS</li> <li>● t2t_FrontOfficeTransaction.STG_TRUSTS_TXNS</li> </ul>
Front Office Transaction Party	FO_TRXN_PAR TY_STAGE	t2t_FrontOfficeTransactionParty
Inside Quote	BBO_STAGE	t2t_InsideQuote
Insurance Policy	INSURANCE_P OLICY	t2t_InsurancePolicy.STG_LIFE_INS_CONTRACTS
Insurance Policy Balance	INSURANCE_P OLICY_BAL	t2t_InsurancePolicyBalance.STG_LIFE_INS_CONTRACTS

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Insurance Policy Feature	INSURANCE_POLICY_FEATURE	t2t_InsurancePolicyFeature
Insurance Policy To Customer	INSURANCE_POLICY_CUST	t2t_InsurancePolicyToCustomer
Insurance Product	INSURANCE_PRODUCT	t2t_InsuranceProduct
Insurance Seller	INSURANCE_SELLER	t2t_InsuranceSeller
Insurance Seller To License	INSURANCE_SELLER_LICENSE	t2t_InsuranceSellerToLicense.STG_INS_SELLER_LICENSE
Insurance Transaction	INSURANCE_TRANSACTION	t2t_InsuranceTransaction.STG_LIFE_INS_POLICY_TXNS
Investment Advisor	NVSMT_MGR	t2t_InvestmentAdvisor
Issuer	ISSUER	t2t_Issuer.STG_PARTY_MASTER
Loan	LOAN	<ul style="list-style-type: none"> <li>● t2t_Loan.STG_CARDS</li> <li>● t2t_Loan.STG_LOAN_CONTRACTS</li> </ul>
Loan Daily Activity	LOAN_SMRY_MNTH_STAGE	t2t_LoanDailyActivity.STG_LOAN_CONTRACTS
Loan Origination Document Print Log	LOAN_ORIG_DOC_PRINT_LOG	t2t_LoanOriginationDocumentPrintLog
Loan Product	LOAN_PRODUCT	t2t_LoanProduct
Managed Account	MANGD_ACCT	t2t_ManagedAccount
Market Center	MARKET_CENTER	t2t_MarketCenter.STG_MARKET_CENTER_MASTER
Market Center Quote	QUOTE_STAGE	t2t_MarketCenterQuote
Market Index	MKT_IDX	t2t_MarketIndex
Market Index Daily	MKT_IDX_DAILY	t2t_MarketIndexDaily
Market Index Member Security	MKT_IDX_MBR_SCRTY	t2t_MarketIndexMemberSecurity
Mutual Fund Breakpoint	MFUND_BRKPT	t2t_MutualFundBreakpoint

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Online Account	ONLINE_ACCT	t2t_OnlineAccount
Online Account To Account	ONLINE_ACCT_ACCT	t2t_OnlineAccountToAccount
Open Order Stage		t2t_OpenOrderStage.STG_OPEN_TRADE_ORDER
Order Stage		t2t_OrderStage.STG_TRADE_ORDER
Organization	ORG	<ul style="list-style-type: none"> <li>● t2t_Organization.STG_GEOGRAPHY_MASTER</li> <li>● t2t_Organization.STG_ORG_STRUCTURE_MASTER</li> <li>● t2t_Organization.STG_ORG_UNIT_MASTER</li> <li>● t2t_Organization.STG_TRADING_DESK_MASTER</li> </ul>
Organization Relationship	ORG_RLSHP	t2t_OrganizationRelationship.STG_ORG_STRUCTURE_MASTER
Party Identification Document		t2t_PartyIdentificationDocument.STG_CUSTOMER_IDENTIFCTN_DOC
Party to Party Relationship		t2t_PartytoPartyRelationship.STG_PARTY_PARTY_RELATIONSHIP
Peer Group	PEER_GRP	t2t_PeerGroup
Reported Market Sale	REPORTED_SALE_STAGE	t2t_ReportedMarketSale
Restriction List	RSTRN_LIST	t2t_RestrictionList
Security	SCRTY	t2t_Security.STG_INSTRUMENT_CONTRACT_MASTER
Security Firm Daily	SCRTY_FIRM_DAILY	t2t_SecurityFirmDaily
Security Group Member	RLTD_SCRTY	t2t_SecurityGroupMember
Security Investment Rating	SCRTY_NVSMRTNG	t2t_SecurityInvestmentRating
Security Market Daily	SCRTY_MKT_DAILY	t2t_SecurityMarketDaily.STG_INSTRUMENT_MARKET_PRICES
Security Trading Restriction	SCRTY_RSTRN_LIST	t2t_SecurityTradingRestriction.STG_INST_TRADE_RESTRICTION

Interface File Name	Data Quality Group Name	Corresponding T2T Name
Service Team	ACCT_SRVC_T EAM	t2t_ServiceTeam
Service Team Member	ACCT_SRVC_T EAM_MEMBER	t2t_ServiceTeamMember
Settlement Instruction	INSTRUCTION	t2t_SettlementInstruction.STG_SETTLEMENT_INSTRUCTION
Structured Deal	DEAL	t2t_StructuredDeal
Trade Execution Event Stage		t2t_TradeExecutionEventStage.STG_TRADE_EXECUTION
Trusted Pair		t2t_TrustedPair
Watch List	WATCH_LIST_S OURCE	t2t_WatchList
Watch List Entry	WATCH_LIST	t2t_WatchListEntry

## 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.

## Flat File Ingestion

This section refers to Behavior Detection (BD) Ingestion Flat Files and covers the following topics:

- [BDF.xml File Parameters](#)
- [Behavior Detection Flat File Interface](#)

### 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.

**Table 69. Parameters Related to Processing DIS Files**

Property Name	Description	Default
DIS.Source	Indicates the source of DIS records. Valid values are: <ul style="list-style-type: none"> <li>• FILE for a DIS file</li> <li>• FSDW for CSA table loading</li> <li>• FILE-EXT for loading DIS file using an external table</li> </ul>	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
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

Property Name	Description	Default
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).	<ul style="list-style-type: none"> <li>● CHECK</li> <li>● CHECK-ACH</li> </ul>
CustProfileLookBack	Indicates the parameter which is used to look back at the days in Customer Summary Daily for Customer Summary Month recalculation.  <b>Note:</b> 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

## BD 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 70. BD Ingest DIS Data Files By Group**

Group	Data Files
1.	Account Phone Watch List Account Emal IAddress Insurance Product Insurance Policy Insurance Transaction Insurance Policy Balance Front Office Transaction Account Customer Role Organization Insurance Policy Feature Market Center Insurance Policy To Customer Market Index Daily Loan Issuer Loan Daily Activity Market Index Online Account Service Team Insurance Seller Service Team Member Insurance Seller To License Country
2.	Account To Peer Group Account Group Peer Group Security Firm Daily Market Index Member Security Security Market Daily Security
3.	Account Customer Watch List Entry Loan Product Employee Front Office Transaction Party Organization Relationship Restriction List Automated Quote

**Table 70. BD Ingest DIS Data Files By Group**

Group	Data Files																																
4.	<table border="0"> <tr> <td>Managed Account</td> <td>Controlling Customer</td> </tr> <tr> <td>Account To Customer</td> <td>Employee To Account</td> </tr> <tr> <td>Account Group Member</td> <td>Account Position</td> </tr> <tr> <td>Account To Correspondent</td> <td>Security Trading Restriction</td> </tr> <tr> <td>Account Balance</td> <td>Employee Trading Restriction</td> </tr> <tr> <td>Account Address</td> <td>Employee Phone</td> </tr> <tr> <td>Customer To Markets Served</td> <td>Employee Email Address</td> </tr> <tr> <td>Customer To Products Offered</td> <td>Employee Address</td> </tr> <tr> <td>Customer To Customer Relationship</td> <td>Security Group Member</td> </tr> <tr> <td>Anticipatory Profile</td> <td>Security Investment Rating</td> </tr> <tr> <td>Customer Phone</td> <td>Structured Deal</td> </tr> <tr> <td>Customer Email Address</td> <td>Account Profit And Loss</td> </tr> <tr> <td>Customer Country</td> <td>Account Investment Objective</td> </tr> <tr> <td>Customer Address</td> <td>Account Position Pair</td> </tr> <tr> <td>Online Account To Account</td> <td>Mutual Fund Breakpoint</td> </tr> <tr> <td></td> <td>Market News Event</td> </tr> </table>	Managed Account	Controlling Customer	Account To Customer	Employee To Account	Account Group Member	Account Position	Account To Correspondent	Security Trading Restriction	Account Balance	Employee Trading Restriction	Account Address	Employee Phone	Customer To Markets Served	Employee Email Address	Customer To Products Offered	Employee Address	Customer To Customer Relationship	Security Group Member	Anticipatory Profile	Security Investment Rating	Customer Phone	Structured Deal	Customer Email Address	Account Profit And Loss	Customer Country	Account Investment Objective	Customer Address	Account Position Pair	Online Account To Account	Mutual Fund Breakpoint		Market News Event
Managed Account	Controlling Customer																																
Account To Customer	Employee To Account																																
Account Group Member	Account Position																																
Account To Correspondent	Security Trading Restriction																																
Account Balance	Employee Trading Restriction																																
Account Address	Employee Phone																																
Customer To Markets Served	Employee Email Address																																
Customer To Products Offered	Employee Address																																
Customer To Customer Relationship	Security Group Member																																
Anticipatory Profile	Security Investment Rating																																
Customer Phone	Structured Deal																																
Customer Email Address	Account Profit And Loss																																
Customer Country	Account Investment Objective																																
Customer Address	Account Position Pair																																
Online Account To Account	Mutual Fund Breakpoint																																
	Market News Event																																
5.	<table border="0"> <tr> <td>Borrower</td> <td>Investment Advisor</td> </tr> <tr> <td>Account Restriction</td> <td>Settlement Instruction</td> </tr> <tr> <td>Back Office Transaction</td> <td>Loan Origination Document Print Log</td> </tr> </table>	Borrower	Investment Advisor	Account Restriction	Settlement Instruction	Back Office Transaction	Loan Origination Document Print Log																										
Borrower	Investment Advisor																																
Account Restriction	Settlement Instruction																																
Back Office Transaction	Loan Origination Document Print Log																																
6.	<table border="0"> <tr> <td>OpenOrder</td> <td>TradeExecutionEvent</td> </tr> <tr> <td>Order</td> <td></td> </tr> </table>	OpenOrder	TradeExecutionEvent	Order																													
OpenOrder	TradeExecutionEvent																																
Order																																	

## Behavior Detection Flat File Interface

The following tables describe the Ingestion Flat File details for products within the BD 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 [List of Data Quality Group Names and T2T Names](#)

The Staging Representation column indicates whether this file requires a Staging source.

The following table describes the Group 1 Ingestion Flat File details.

**Table 71. Group 1 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Account Phone	X		X	X	X						BD Datamaps	Yes	Yes
Account Email Address	X		X	X	X						BD Datamaps	Yes	Yes
Insurance Policy	X		X	X							BD Datamaps	Yes	Yes
Insurance Policy Balance	X		X								BD Datamaps	Yes	Yes
Account Customer Role	X		X		X	X					BD Datamaps	Yes	Yes



Table 71. Group 1 Interface Ingestion Flat Files

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Insurance Policy Feature	X		X								BD Datamaps	Yes	Yes
Insurance Policy to Customer	X		X	X							BD Datamaps	Yes	Yes
Loan	X		X								BD Datamaps	Yes	Yes
Loan Daily Activity	X		X								BD Datamaps	Yes	Yes
Online Account	X		X								BD Datamaps	Yes	Yes
Insurance Seller	X		X								BD Datamaps	Yes	Yes
Insurance Seller to License	X		X								BD Datamaps	Yes	Yes
Country	X		X		X						BD Datamaps	Yes	Yes
Watch List	X		X	X							BD Datamaps	Yes	Yes
Insurance Product	X		X	X							BD Datamaps	Yes	Yes
Insurance Transaction	X		X								BD Datamaps	Yes	Yes
Front Office Transaction	X		X								BD Datamaps	Yes	Yes
Organization						X	X		X		BD Datamaps	Yes	No
Market Center							X				BD Datamaps	Yes	No
Market Index Daily							X				BD Datamaps	Yes	No
Issuer							X				BD Datamaps	Yes	No
Market Index							X				BD Datamaps	Yes	No
Service Team Member									X		BD Datamaps	Yes	No
Service Team									X		BD Datamaps	Yes	No
CTR Transaction	X		X			X					runDP/runDL	No	No
Account Realized Profit and Loss									X		runDP/runDL	No	No
Letter of Intent									X		runDP/runDL	No	No

**Table 71. Group 1 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Collateral Value-Currency									X		runDP/runDL	No	No
Collateral Value-Product									X		runDP/runDL	No	No
Commission Product									X		runDP/runDL	No	No
Compliant Registration									X		runDP/runDL	No	No
Complaint Type Rating									X		runDP/runDL	No	No
Employee to Insurance Policy									X		runDP/runDL	No	No
Investment Guideline									X		runDP/runDL	No	No
Investment Guideline to Account									X		runDP/runDL	No	No
System Logon Type									X		runDP/runDL	No	No
Registered Representative Complaint									X		runDP/runDL	No	No
Energy And Commodity Instrument										X	runDP/runDL	No	No

The following table describes the Group 2 Ingestion Flat File details.

**Table 72. Group 2 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Account to Peer Group	X		X	X							BD Datamaps	Yes	Yes
Account Group	X		X								BD Datamaps	Yes	Yes
Peer Group	X		X	X							BD Datamaps	Yes	Yes
Security Market Daily							X				BD Datamaps	Yes	No
Security Firm Daily							X				BD Datamaps	Yes	No
Security							X	X			BD Datamaps	Yes	No

**Table 72. Group 2 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Market Index Member Security							X				BD Datamaps	Yes	No
Security Market State Change							X				BD Datamaps	Yes	No
Matched Entity	X		X								runDP/runDL	No	No
Trusted Pair	X		X								BD Datamaps	Yes	No
Firm Account Position Pair							X		X		runDP/runDL	No	No
Natural Gas Flow										X	runDP/runDL	No	No

The following table describes the Group 3 Ingestion Flat File details.

**Table 73. Group 3 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Account	X		X	X	X	X					BD Datamaps	Yes	Yes
Customer	X		X	X	X	X					BD Datamaps	Yes	Yes
Watch List Entry	X		X	X							BD Datamaps	Yes	Yes
Loan Product	X		X								BD Datamaps	Yes	Yes
Employee	X		X								BD Datamaps	Yes	Yes
Front Office Transaction Party	X		X								BD Datamaps	Yes	Yes
Organization Relationship						X	X		X		BD Datamaps	Yes	No
Restriction List							X				BD Datamaps	Yes	No
Automated Quote							X				BD Datamaps	No	No
Account Supplemental Attribute				X							runDP/runDL	No	No
Customer Supplemental Attribute				X							runDP/runDL	No	Yes

**Table 73. Group 3 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Market Trading Session							X				runDP/runDL	No	No
Account GroupAddress	X		X								runDP/runDL	No	No
Account Group Investment Objective									X		runDP/runDL	No	No
Account Group IOS Member									X		runDP/runDL	No	No
Account Group Member Experience									X		runDP/runDL	No	No
Loan Origination Action									X		runDP/runDL	No	No
Mail Handling Instruction Activity									X		runDP/runDL	No	No
Banker To Officer									X		runDP/runDL	No	No
Reference Table Detail									X		runDP/runDL	No	No
General Usage List									X		runDP/runDL	No	No
Loan Origination Product									X		runDP/runDL	No	No
Organization To Mortgage Type									X		runDP/runDL	No	No
Securities License									X		runDP/runDL	No	No
Service Vendor									X		runDP/runDL	No	No
Energy and Commodity Trade										X	runDP/runDL	No	No

The following table describes the Group 4 Ingestion Flat File details.

**Table 74. Group 4 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Market News Event							X				BD Datamaps	No	No
Managed Account	X		X	X							BD Datamaps	Yes	No
Account To Customer	X		X	X	X	X					BD Datamaps	Yes	Yes

Table 74. Group 4 Interface Ingestion Flat Files

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Branch CTR Transaction						X					BD Datamaps	Yes	No
Branch CTR Conductor						X					BD Datamaps	Yes	No
Branch CTR Summary						X					BD Datamaps	Yes	No
Account Group Member	X		X								BD Datamaps	Yes	Yes
Account To Correspondent	X		X								BD Datamaps	Yes	Yes
Account Balance	X		X	X	X						BD Datamaps	Yes	Yes
Account Address	X		X	X	X						BD Datamaps	Yes	Yes
Customer Identification Document	X		X	X	X						BD Datamaps	Yes	Yes
Customer To Markets Served	X		X	X							BD Datamaps	Yes	Yes
Customer To Products Offered	X		X	X							BD Datamaps	Yes	Yes
Customer To Customer Relationship	X		X	X	X						BD Datamaps	Yes	Yes
Anticipatory Profile	X		X	X							BD Datamaps	Yes	Yes
Customer Phone	X		X	X	X	X					BD Datamaps	Yes	Yes
Customer Email Address	X		X	X	X	X					BD Datamaps	Yes	Yes
Customer Country	X		X	X							BD Datamaps	Yes	Yes
Customer Address	X		X	X	X	X					BD Datamaps	Yes	Yes
Online Account to Account	X		X	X							BD Datamaps	Yes	No
Controlling Customer	X		X								BD Datamaps	Yes	No
Employee To Account	X		X								BD Datamaps	Yes	Yes
Account Position							X		X		BD Datamaps	Yes	No
Security Trading Restriction							X	X			BD Datamaps	Yes	No

Table 74. Group 4 Interface Ingestion Flat Files

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Employee Trading Restriction							X	X			BD Datamaps	Yes	No
Employee Phone							X				BD Datamaps	Yes	Yes
Employee Email Address							X	X			BD Datamaps	Yes	Yes
Employee Address							X				BD Datamaps	Yes	Yes
Outside Business Activity											BD Datamaps	Yes	No
Private Security Transaction											BD Datamaps	Yes	No
Security Group Member							X				BD Datamaps	Yes	No
Security Investment Rating							X				BD Datamaps	Yes	No
Structured Deal							X				BD Datamaps	Yes	No
Account Profit and Loss									X		BD Datamaps	Yes	No
Account Position Pair									X		BD Datamaps	Yes	No
Account Investment Objective									X		BD Datamaps	Yes	No
Mutual Fund Breakpoint									X		BD Datamaps	Yes	No
Account Feature									X		runDP/runDL	No	No
Access Events			X								runDP/runDL	No	No
Customer Balance			X								runDP/runDL	No	No
Front Office Transaction Remittance Document	X		X								runDP/runDL	No	No
Related Front Office Transaction Information	X		X								runDP/runDL	No	No
Account To Organization							X		X		runDP/runDL	No	No
Firm Account Position							X		X		runDP/runDL	No	No
External Investment Account Position								X	X		runDP/runDL	No	No
Employee To Organization								X	X		runDP/runDL	No	No
Security Select List Entry							X				runDP/runDL	No	No

Table 74. Group 4 Interface Ingestion Flat Files

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Account Fees									X		runDP/runDL	No	No
Account Profile Stage									X		runDP/runDL	No	No
Account Qualification Agreement									X		runDP/runDL	No	No
Account Representative Position									X		runDP/runDL	No	No
Account Asset Allocation									X		runDP/runDL	No	No
Account Scheduled Event									X		runDP/runDL	No	No
Account Identifier Change History									X		runDP/runDL	No	No
Account Position Profile And Loss									X		runDP/runDL	No	No
Uncovered Option Account Position									X		runDP/runDL	No	No
Account Collateral									X		runDP/runDL	No	No
Mail Handling Instruction									X		runDP/runDL	No	No
Mutual Fund Family Letter of Intent									X		runDP/runDL	No	No
Employee Disciplinary Action									X		runDP/runDL	No	No
Employee Exam History									X		runDP/runDL	No	No
Employee Firm Transfer History									X		runDP/runDL	No	No
Employee Securities License State Registration									X		runDP/runDL	No	No
Employee Supervision List									X		runDP/runDL	No	No
Employee To Manager History									X		runDP/runDL	No	No
Employee To Securities License									X		runDP/runDL	No	No
Employment History									X		runDP/runDL	No	No
System Logon									X		runDP/runDL	No	No
Plan of Solicitation									X		runDP/runDL	No	No
Mutual Fund Family Configuration									X		runDP/runDL	No	No

**Table 74. Group 4 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Energy And Commodity Market Daily										X	runDP/runDL	No	No
Energy And Commodity Firm Daily										X	runDP/runDL	No	No
Energy And Commodity Reported Market Sale										X	runDP/runDL	No	No
Energy And Commodity Market Trading Session										X	runDP/runDL	No	No
Energy And Commodity Market Center										X	runDP/runDL	No	No
Energy And Commodity Location										X	runDP/runDL	No	No
Energy Flow Mode										X	runDP/runDL	No	No
Energy and Commodity Instrument Position										X	runDP/runDL	No	No

The following table describes the Group 5 Ingestion Flat File details.

**Table 75. Group 5 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Borrower	X		X								BD Datamaps	Yes	No
Back Office Transaction	X		X								BD Datamaps	Yes	Yes
Account Restriction				X			X				BD Datamaps	Yes	No
Investment Advisor							X				BD Datamaps	Yes	No
Investment Guideline Override											BD Datamaps	Yes	No
Settlement Instruction							X				BD Datamaps	Yes	No
Loan Origination Document Print Log									X		BD Datamaps	Yes	No
Change Log	X		X	X	X						runDP/runDL	No	No



**Table 75. Group 5 Interface Ingestion Flat Files**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Options Violation									X		runDP/runD L	No	No
Loan Origination Condition									X		runDP/runD L	No	No
Loan Origination Fee Detail									X		runDP/runD L	No	No
Loan Origination Note									X		runDP/runD L	No	No
Loan Origination To Service									X		runDP/runD L	No	No
Investment Guideline Override									X		runDP/runD L	No	No
Loan Origination Condition Type									X		runDP/runD L	No	No
System Logon To System Logon Type									X		runDP/runD L	No	No
System Logon To Organization									X		runDP/runD L	No	No
Registered Representative Account Commission									X		runDP/runD L	No	No
Registered Representative Account Commission Prior Year									X		runDP/runD L	No	No
Registered Representative Commission Monthly Profile									X		runDP/runD L	No	No
Registered Representative Commission Product									X		runDP/runD L	No	No
Currency Transaction						X					BD Datamaps	Yes	No

The following table describes the Group 6 Ingestion Flat File details.

**Table 76. Group 6 Interface Ingestion for Market Data**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
Inside Quote							X				BD Datamaps	Yes	No
Market Center Quote							X				BD Datamaps	Yes	No
ReportedMarketSale							X				BD Datamaps	Yes	No
InsideQuote_Derived							X				BD Datamaps	Yes	No
MarketCenterQuote_Derived							X				BD Datamaps	Yes	No
ReportedMarketSale_Derived							X				BD Datamaps	Yes	No

The following table describes the Group 7 Ingestion Flat File details.

**Table 77. Group 7 Interface Ingestion for Trade Finance Data**

Interface File Name	AML	TBAML	Fraud	KYC	FATCA	CTR	TC	PTA	BC	ECTC	Current Ingestion	Staging Representation	T2T
TradeFinanceContractEventAcknowledgement							X				BD Datamaps	Yes	No
TradeFinanceContractAmendmentStatus							X				BD Datamaps	Yes	No
TradeFinanceContract							X				BD Datamaps	Yes	No
TradeFinancetoAccount							X				BD Datamaps	Yes	No
TradeFinanceDocument							X				BD Datamaps	Yes	No
TradeFinanceGoodorService							X				BD Datamaps	Yes	No
TradeFinanceParty							X				BD Datamaps	Yes	No
DocCollectionContractAcknowledgementStage							X				BD Datamaps	Yes	No
DocumentaryCollectionContractAcceptanceStage							X				BD Datamaps	Yes	No
DocumentaryCollectionDiscrepancyDetail							X				BD Datamaps	Yes	No
DocumentaryCollectionContractEvent							X				BD Datamaps	Yes	No

**Note:** The AccountAverageNetWorth file is an exceptional case, and is only intended to be run once before any other files have been loaded. The average net worth amount in the account profile table is built up over time as transactions are ingested. This file allows this value to be set as a starting point before any transactions have been ingested. After transactions are ingested, this file should no longer be used.

**Note:** The following derived datamaps must be run after running the corresponding BD scripts.

CurrencyTransaction\_ExemptFlagUpd  
SecurityInvestmentRating\_PrevInvestmentUpd  
AutomatedQuote\_SecurityUpd

For Example:

AutomatedQuote\_SecurityUpd should be run after <OFSAAI Installed Directory>/BDF/scripts/execute.sh AutomatedQuote as <OFSAAI Installed Directory>/BDF/scripts/execute.sh AutomatedQuote\_SecurityUpd

## Pre-processing & Loading Directory Structure

Data for Pre-processing & Loading are organized in subdirectories below the ingestion\_manager root level. Figure 60 illustrates the subdirectories that the ingestion\_manager directory contains.

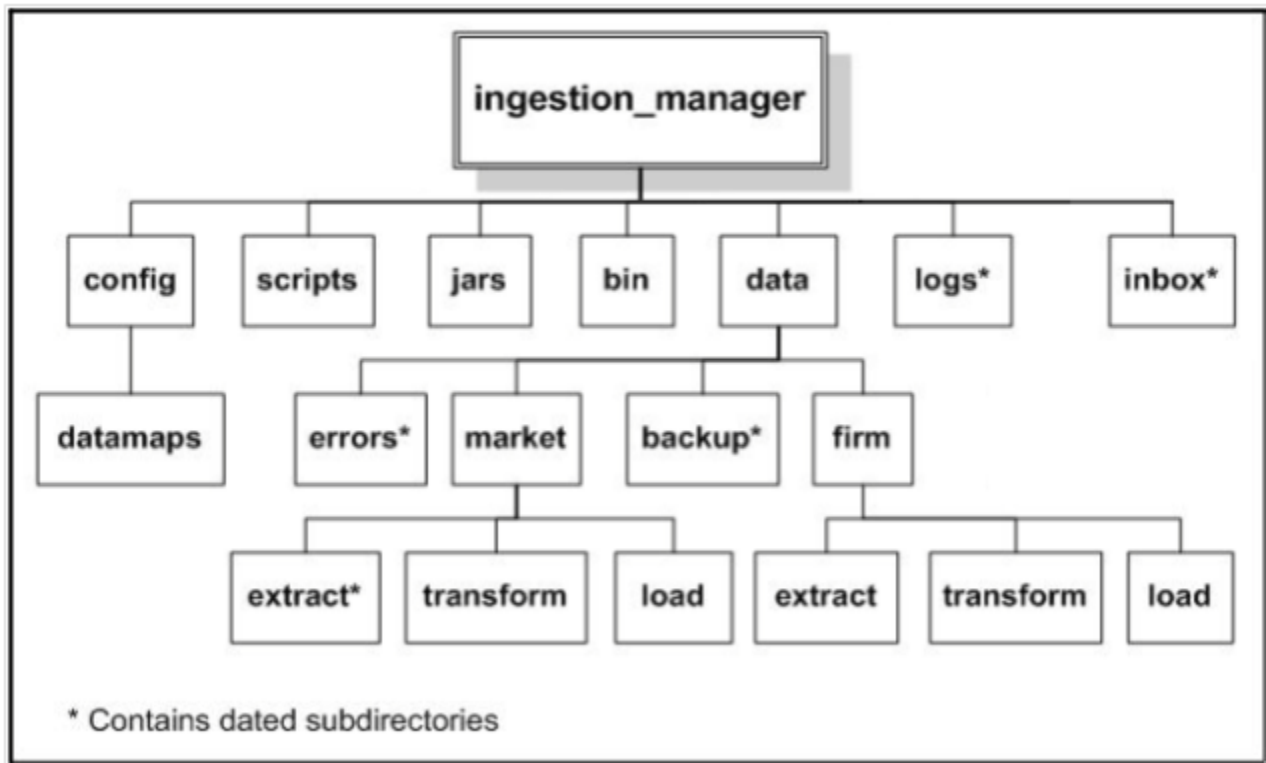


Figure 60. Data Management Subsystem Directory Structure

## Directory Structure Descriptions

The following table lists important subdirectories that compose the <OFSAAI Installed Directory>/ingestion\_manager directory structure.

**Table 78. Data Management Directory Structure Description**

Directory Name	Description
config	Contains files used to configure the Data Management components (see <i>config Subdirectory</i> for more information).
data/backup	Contains backup files for the various Data Management components (see <i>data/backup Subdirectory</i> for more information).
data/errors	Contains error files for various Data Management components (see <i>data/errors Subdirectory</i> for more information).
data/firm	Contains Oracle client data files that Data Management components write (see <i>data/firm Subdirectory</i> for more information).
inbox	Contains data files that the Oracle client provides (see <i>inbox Subdirectory</i> for more information).
jars	Contains the Java Archive (JAR) files to run Java Data Management components implemented in Java (see <i>jars Subdirectory</i> for more information).
logs	Contains log files that Data Management components write (see <i>logs Subdirectory</i> for more information).
scripts	Contains all the shell scripts for running Data Management components (see <i>scripts Subdirectory</i> for more information).
/inbox/<yyyymmdd>	Backup of input files (for restart purposes, if necessary).
/data/<firm or market>/load	<ul style="list-style-type: none"><li>• for loading into the database as &lt;data type&gt;_&lt;yyyymmdd&gt;_&lt;batch_name&gt;_&lt;N&gt;.XDP.</li><li>• Load control files.</li></ul>
/logs/<yyyymmdd>	Pre-processing and load status, and error messages.
/data/errors/<yyyymmdd>	Records that failed validation. The file names are the same as those of the input files.
/data/firm/transform	TC trading data files that the FDT processes.

This section covers the following topics:

- jars Subdirectory
- scripts Subdirectory
- data Subdirectory
- extract Subdirectory
- transform Subdirectory
- load Subdirectory
- inbox Subdirectory
- logs Subdirectory

### jars Subdirectory

The jars subdirectory within the ingestion\_manager directory contains Java programs that Ingestion Manager uses. A run script in the scripts subdirectory launches each program (see *scripts Subdirectory* for more information).

## scripts Subdirectory

The scripts subdirectory within the ingestion\_manager directory contains the UNIX Bourne Shell scripts to run runtime components. Executing a run script runs a new instance of a component. If an application component terminates successfully, a script returns a zero return code. If the component fails to terminate successfully, the script returns a non-zero status (normally 1). The following table defines the run scripts for starting each component and any special instructions.

**Table 79. Run Scripts by Component**

Script Names	Description or Special Instructions
runDP.sh <data type>	Launches an instance of the data Pre-processor (runDP.sh). For example: runDP.sh Customer To run a specific Data Pre-processor, specify a valid input component that the run script recognizes. If the script does not recognize the input component, it exits with an error and identifies the valid list of parameters. For valid component names, see Figure 61
runFDT.sh	Launches the FDT. This script stops after it processes all qualifying files that it finds in the /data/firm/transform directory at the time the process starts. The system processes an input file if the processing data and batch name are correct. You can stop the FDT immediately by using the UNIX kill command to stop the process ID for the Java process that is a child of the runFDT.sh process.
runDL.sh <data type>	Launches an instance of the data loader (runDL.sh). For example: runDL.sh Customer To run a specific data loader, specify a valid component that the run script recognizes. If the script does not recognize the component, it exits with an error and identifies the valid list of parameters. For valid component names, see Figure 61.
runRebuildIndexes.sh	Launches a process to rebuild the indexes of the given component. Processing requires this script only during use of a live market feed. A valid <component> value is one of InsideQuote, ReportedMarketSale, or MarketCenterQuote.
process_firm_summary.sh	Calls a database procedure to build summary statistics about the Oracle client (firm) data.
process_market_summary.sh	Calls a database procedure to build summary statistics about the Market data.
market_analyze.sh	Calls a database procedure to create internal database statistics for Market tables.
firm_analyze.sh	Calls a database procedure to create internal database statistics for Oracle client (firm) tables.
runIMC.sh	Launches the Ingestion Manager Cleaner (IMC) utility. The utility terminates after it finishes removing old data subdirectories and their contents.
env.sh	Contains common configuration settings required to run Data Management subsystem components. The run*.sh scripts use this script.
truncate_table.sh <schema.tablename>	Truncates a specified table in the database. Processing runs this script prior to loading reference data when an Oracle client wants to perform a full refresh of the data.
runUtility.sh <datatype>	Launches a Java based utility to derive the contents of a given database table. You must run runDL.sh <data type> after this script has executed successfully. For example: runUtility.sh AccountDailySecurityProfile runDL.sh AccountDailySecurityProfile

The run scripts in Table 80 configure the executing environment for the Java component, and then execute it. All run scripts invoke the `env.sh` script to define environment variables that the components require. The run scripts also start the Java program with appropriate command line parameters, which Table 80 describes.

**Table 80. Environment Variable Descriptions**

Parameter	Description
<code>classpath</code>	Directs the Java Runtime Environment (JRE) to the location of Java programs and supporting Java classes.
<code>Djava.security.policy</code>	Sets the location of the policy file that provides directory and network access rights to the component.
<code>server</code>	Instructs Java JRE to optimize for server-based processing.
<code>Xms&lt;NNNN&gt;*</code>	Indicates the minimum number of megabytes (as NNNN) to reserve for Java memory allocation.
<code>Xmx&lt;NNNN&gt;*</code>	Indicates the maximum number of megabytes (as NNNN) to reserve for Java memory allocation. Note: Setting <code>Xmx</code> too small may result in component failure.

**Note:** Default values that are appropriate to the operating system in use, such as Linux or Solaris, are automatically set in the `env.sh` file:

- For 64-bit operating systems, the maximum value should not be greater than 3500 MB.
- For 32-bit operating systems, the maximum value should not be greater than 1800 MB.

Minimum values vary by component; the `env.sh` file specifies these values.

### config Subdirectory

The `config` subdirectory within the `data_ingest` directory contains the application configuration files, as Table 81 describes:

- `DataIngestCustom.xml` (see section *Data Ingest XML Configuration File* for more information).
- `DataIngest.properties` (see section *Data Ingest Properties Configuration File* for more information).
- `DataIngest.xml` (see section *Data Ingest XML Configuration File* for more information).

The `DataIngest.properties` and `DataIngest.xml` files contain settings for IP addresses, port numbers, file paths, file extensions, and other runtime settings including an application's performance tuning parameters. Property files within the `config` subdirectory contain database user IDs and encrypted passwords.

The `config/datamaps` subdirectory also contains XML data maps for parsing input data and mapping processed data to fields in files and in databases. The XML data maps are preset and do not require any modifications.

**Table 81. Application Configuration Files**

File Name	Description
<code>DataIngest.properties</code>	Property file that contains settings that are configured at installation. These settings are of the most interest to an Oracle client regarding modification (see Table 82).
<code>DataIngest.xml</code>	XML configuration file that contains settings that normally remain as is (see Table 83).
<code>DataIngestCustom.xml</code>	XML configuration file that contains overridden settings from <code>DataIngest.xml</code> .

The following sections describe each of these configuration files:

## Data Ingest Properties Configuration File

The following table describes the parameters for the `DataIngest.properties` configuration file.

**Table 82. DataIngest.properties File Configuration Parameters**

Property Name	Description	Example
<code>DB.Connection.URL</code>	Database URL for JDBC connections made by Java ingestion components. The content and format of this value is specific to the database vendor and the vendor database driver. Oracle recommends that you use Thin Driver.	<code>jdbc:oracle:thin:@ofss220074.in.Oracle.com:1521:Ti1O11L56</code>
<code>DB.Connection.Instance</code>	Database instance to connect to on the database servers. Typically, the instance name matches the database name portion of the <code>DB.Connection.URL</code> .	D1O9L2
<code>DB.Connection.User</code>	Database user name that Java ingestion components uses when connecting to the database. The database user must have been assigned the appropriate privileges that Data Management requires for interacting with the database.	ATOMIC
<code>DB.Connection.Password</code>	Password that Java Ingestion components use when connecting with the database. This is set by the Password Manager Utility.	
<code>DB.Type</code>	The type of database being used.	Oracle
<code>MANTAS.DBSchema</code>	Schema name for the ATOMIC database schema. Data Management accesses the ATOMIC schema when allocating sequence IDs to ingested records.	ATOMIC
<code>MARKET.DBSchema</code>	Schema name for the ATOMIC database schema. Data Management stores market data related records in the ATOMIC schema.	ATOMIC
<code>BUSINESS.DBSchema</code>	Schema name for the ATOMIC database schema. Data Management stores market data related records in the ATOMIC schema.	ATOMIC

## Data Ingest XML Configuration File

The following table describes the parameters for the `DataIngest.xml` configuration file.

**Caution:** Default values for properties in this file are suitable for most deployments. Use caution when changing any default values.

**Table 83. DataIngest.xml File Configuration Parameters**

Property Name	Description	Example
<b>ProcessingBatch:</b> Specifies batch settings that override settings in the database. Overrides are primarily useful during testing.		
<code>ProcessingBatch.Name</code>	Sets the current batch name. Ingestion components process only input files that contain this value in the batch name portion of the file name. This property should be blank during normal operation.	
<code>ProcessingBatch.Date</code>	Sets the current processing date. Ingestion components process only input files that contain this value in the processing date portion of the file name. This property should be blank during normal operation. The date format is YYYYMMDD.	
<code>ProcessingBatch.Last</code>	Identifies the flag that indicates processing of the last batch of the day to Data Management. This property should be blank during normal operation.	
<b>Miscellaneous</b>		
<code>DefaultSourceSystem.value</code>	Indicates the default value to use for source system when manufacturing reference data records.	MTS
<code>BufferSize.value</code>	Specifies the buffer size in kilobytes for I/O byte buffers that the MDS and FDT processes create to read input files.  Use care when changing this parameter due to impact on performance and memory requirements.	1024
<code>DirectBufferSize.value</code>	Specifies the buffer size in kilobytes for Java NIO direct byte buffers that the MDS, MDT, and FDT processes create to read input files.  Use care when changing this parameter due to impact on performance and memory requirements	1024
<code>DefaultCurrency.value</code>	Indicates the value to use as the issuing currency when manufacturing security records from order or trade execution records.	USD
<code>UseDirectBuffers.value</code>	Specifies whether to make use of Java NIO's direct buffer mechanism.	TRUE



**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
<code>Separator.value</code>	Specifies the delimiter that separates fields in data file records.	~
<b>Log:</b> Specifies properties used to configure the common logging module.		
<code>Log.UseDefaultLog</code>	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
<code>Log.UseDateLog</code>	Specifies whether to put default log file for a component in a date subdirectory. Otherwise, it is placed directly under the logs directory.	TRUE
<code>Log.InitDir</code>	Specifies the location of the properties file for configuring the common logging module ( <code>install.cfg</code> ).	<code>../config</code>
<b>DB:</b> Specifies properties related to database access.		
<code>DB.Connection.Driver</code>	Indicates the JDBC driver class name.	<code>oracle.jdbc.driver.OracleDriver</code>
<code>DB.Connection.InitialConnections</code>	Specifies the number of connections initially to allocate in the connection pool.	1
<code>DB.Connection.MaximumConnections</code>	Indicates the maximum number of connections in the connection pool. You should correlate this parameter to the number of configured threads for the component.	10
<code>DB.Connection.Timeout</code>	Identifies the number of seconds to wait before timing out on a database connection attempt.	10
<code>DB.Connection.NumRetries</code>	Specifies the maximum number of times to attempt to connect to a database before failing.	5
<b>BUSINESS:</b> Specifies properties related to data loaded into the ATOMIC schema.		
<code>BUSINESS.ExtractDir</code>	Identifies the parent directory for intermediate files that Pre-processors produce that are applicable to the ATOMIC schema in the database.	<code>../data/firm/extract</code>
<code>BUSINESS.TransformDir</code>	Specifies the working directory for the FDT component which transforms BUSINESS trade-related data.	<code>../data/firm/transform</code>
<code>BUSINESS.LoadDir</code>	Indicates the parent directory for directories that store ATOMIC schema bound data files prior to loading with the Java data loader component. Control files for native loaders also reside below this directory.	<code>../data/firm/load</code>

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
<b>MANTAS:</b> Specifies properties related to data loaded into the ATOMIC schema.		
MANTAS.ExtractDir	Specifies the parent directory for intermediate files that Pre-processors produce that are applicable to the ATOMIC schema in the database.	../data/mantas/extract
MANTAS.TransformDir	Specifies the working directory for intermediate files that utilities produce that are applicable to the ATOMIC schema in the database.	../data/mantas/transfor m
MANTAS.LoadDir	Specifies the parent directory for directories that store ATOMIC schema bound data files prior to loading with the Java data loader component. Control files for native loaders also reside below this directory.	../data/mantas/load
<b>Directory:</b> Specifies properties used to define directory locations.		
Directory.Log	Specifies the parent directory for log file directories and log files that Java ingestion components create.	../logs
Directory.Inbox	Specifies the input directory where Java ingestion components find files that the Oracle client submits. Processing creates subdirectories in the /inbox directory for each day of data, to contain a copy of the input data file.	../inbox
Directory.Error	Specifies the parent directory for error directories that contain error data files that Java ingestion components create. Each error data file contains records that were not processed due to error.	../data/errors
Directory.Archive	Specifies the parent directory for directories that contain backup copies of intermediate files that Java ingestion components create.	../data/backup
Directory.Config	Specifies the directory containing configuration files for Java ingestion server.	../config
Directory.FuzzyMatcher	Specifies the directory containing files related to fuzzy matcher.	../fuzzy_match
Directory.DataMap	Specifies the directory that contains XML data map files.	../config/datamaps
<b>FileExtension:</b> Specifies properties used to define extensions for various types of files.		
FileExtension.Log	Specifies the file name extension for log files.	.log
FileExtension.Checkpoint	Specifies the file name extension for checkpoint files. Many of the Java ingestion components create checkpoint files as an aid to recovery when restarted after exiting prematurely.	.cp

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
<code>FileExtension.Temporary</code>	Specifies the file name extension for temporary files that Java ingestion components create.	<code>.tmp</code>
<code>FileExtension.Error</code>	Specifies the file name extension for error files that Java ingestion components create.	<code>.err</code>
<code>FileExtension.Data</code>	Specifies the file name extension for input data files that the Oracle client submits. The default value of <code>.dat</code> is in accordance with the DIS.	<code>.dat</code>
<b>Security:</b> Specifies properties used to produce security reference data.		
<code>Security.AdditionalColumns</code>	Specifies additional columns of data that ingestion components must populate when manufacturing security records.	<code>SCRTY_SHRT_NM,</code> <code>SCRTY_ISIN_ID,</code> <code>PROD_CTGRY_CD,</code> <code>PROD_TYPE_CD,</code> <code>PROD_SUB_TYPE_CD</code>
<b>Symbol:</b> Specifies properties used for looking up security reference data by security short name.		
<code>Symbol.DbTableName</code>	Specifies the name of the database table to use when looking up security records by security short name.	<code>SCRTY</code>
<code>Symbol.KeyColumn</code>	Specifies the column name to use when looking up security records by security short name.	<code>SCRTY_SHRT_NM</code>
<code>Symbol.ValueColumn</code>	Specifies the column to use for retrieving the Behavior Detection assigned identifier for a security.	<code>SCRTY_INTRL_ID</code>
<code>Symbol.Category</code>	Specifies the category of data for the security table. The category is a key for mapping to the database schema in which the security table resides.	<code>BUSINESS</code>
<b>SecurityISIN:</b> Specifies properties used for looking up security ISINs.		
<code>SecurityISIN.DbTableName</code>	Specifies the name of the table to use when looking up a security using the Behavior Detection assigned security identifier.	<code>SCRTY</code>
<code>SecurityISIN.KeyColumn</code>	Specifies the column name to use when looking up security records by Behavior Detection assigned security identifier.	<code>SCRTY_INTRL_ID</code>
<code>SecurityISIN.ValueColumn</code>	Specifies the column to retrieve when looking up a security using the Behavior Detection assigned security identifier.	<code>SCRTY_ISIN_ID</code>
<code>SecurityISIN.Category</code>	Specifies the category of data in which the security table resides. The category is a key for mapping to the database schema in which the security table resides.	<code>BUSINESS</code>
<b>FDT:</b> Specifies properties used to configure the FDT component.		

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
FDT.NumberOfThreads.Value	Specifies the number of worker threads that the FDT uses when processing data.	4
FDT.LowerDisplayLimit.Value	Specifies the quantity below which orders are exempt from display.	100
FDT.UpperDisplayLimit.Value	Specifies the quantity above which orders are exempt from display.	10000
FDT.OrderPriceLimit.Value	Specifies the dollar value above which orders are exempt from display.	200000
FDT.SequenceBatchSize.OrderEvent	Specifies the batch size when retrieving sequence IDs for OrderEvent records (during end-of-day processing).	1000
FDT.SequenceBatchSize.Order	Specifies the batch size when retrieving sequence IDs for Order records.	10000
FDT.SequenceBatchSize.Trade	Specifies the batch size when retrieving sequence IDs for Trade records.	10000
FDT.SequenceBatchSize.Execution	Specifies the batch size when retrieving sequence IDs for Execution records.	10000
FDT.SequenceBatchSize.DerivedTrade	Specifies the batch size when retrieving sequence IDs for DerivedTrade records.	10000
FDT.MarketDataSource.Value	Specifies the source of market data. Valid values are File for file based access or RMI for access using an RMI server (not recommended for performance reasons).	File
FDT.CalculateDisplayability.Value	Specifies whether to calculate displayability states.	FALSE
FDT.ExplainableCancelCodes.Value	Specifies a comma-separated list of explainable cancellation codes.	
FDT.BufferSize.value	Allows an override to the BufferSize.value property for FDT.	
FDT.LookForFutureEventTimes.value		
FDT.UsePrevailingSale.value	Specifies whether to use the prevailing reported market sales price as an execution's expected print price when no comparable market sales occur during the order's marketable periods.	FALSE
Data Management uses the following three parameters when calculating the expected print price for executions. A reported market sale is comparable to an execution when its size is in the same tier.		
FDT.ExecutionSizeThresholds.FirstTierMax	Specifies the maximum size for the first tier.	1000
FDT.ExecutionSizeThresholds.SecondTierMax	Specifies the maximum size for the second tier.	5000
FDT.ExecutionSizeThresholds.ThirdTierMax	Specifies the maximum size for the third tier. Any size bigger than this value is considered part of the fourth tier.	10000
Data Management uses the next five parameters when calculating the marketable time with reasonable size attributes for an order. Processing divides orders into small, medium, and large based on their remaining unit quantities.		

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
FDT.OrderSizeMarketability.MaxSmallSize	Specifies the maximum size for an order to be considered small.	1000
FDT.OrderSizeMarketability.MaxMediumSize	Specifies the maximum size for an order to be considered medium.	5000
FDT.OrderSizeMarketability.SmallMinPercentAtBest	Specifies the minimum percent of a small order's remaining unit quantity that must be available at the best price for execution to be considered reasonable. The minimum percentage value must be represented in its decimal equivalent (for example 1.0 = 100%).	1.0
FDT.OrderSizeMarketability.MediumMinPercentAtBest	Specifies the minimum percent of a medium order's remaining unit quantity that must be available at the best price for execution to be considered reasonable. The minimum percentage value must be represented in its decimal equivalent (for example 1.0 = 100%).	1.0
FDT.OrderSizeMarketability.LargeMinPercentAtBest	Specifies the minimum percent of a large order's remaining unit quantity that must be available at the best price for execution to be considered reasonable. The minimum percentage value must be represented in its decimal equivalent (for example 1.0 = 100%).	1.0
FDT.TradePurposeFilter.value	Specifies a comma-separated list of trade purpose codes. Processing does not consider trades with one of these purpose codes in firm reference price derivations.	IFADM, OFEA, CONB, CLNT, BTBX
FDT.RunBatchesSeparately.value	Specifies whether the FDT treats batches as distinct from one another. TRUE: Three defined batches originate from different geographical areas in which the data in each batch does not overlap (that is, an execution in batch A does not occur against an order in batch B). FALSE: Processing does not separate data in each batch into a distinct time interval (that is, an event in batch A occurred at 10am and an event in batch B occurred at 9am, and batch B arrived after batch A).	TRUE
FDT.RegNMSExceptionCodes	Identifies the Order Handling Codes that should be considered as Reg NMS executions.	ISO, BAP, BRD, BOP, SOE, SHE
FDT.TreatLostEventsAsErrors.value	Identifies whether lost events found by the FDT (see <a href="#">Rejection During the Transformation Stage</a> , for a discussion of lost events) should be treated as errors (TRUE) or as lost events to be read in on the next run of FDT (false).	TRUE

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
FDT.OpenOrderFileExpected.value	Identifies whether an OpenOrder file will be provided by the client during an end of day batch (TRUE) or whether it will not be provided (FALSE).	TRUE
FDT.NonExecutionTradePurposeCodes.value	Specifies a comma-separated list of trade purpose codes. For Trade Execution records that refer to an Order and have one of these codes, the FDT will create a Trade record rather than an Execution record.	CLNT, BTBX
FDT.DeriveTradeBlotter.value	Specifies whether or not the FDT will create a Trade Blotter file.	FALSE
FDT.EnableMIFID.value	Identifies whether MiFid related data will be provided (TRUE) or not (FALSE).	FALSE
FDT.IgnoreFutureMarketRefs.value	Identifies whether the FDT will use Reported Market Sales records that occur later in time than a given trade when calculating the market reference price for that trade (FALSE) or not (TRUE).	FALSE
FDT.MaxFutureMarketRefCompTime.value	Specifies the number of seconds from the time a trade occurs during which any reported sales records cannot have the same price and quantity as the given trade to be considered as a market reference price. -1 means that this condition will not apply, 0 means the condition applies to reported sales with the same time, 5 means the condition applies to reported sales within 5 seconds of the trade, and so on. This parameter is only used if FDT.IgnoreFutureMarketRefs.value = FALSE.	-1
<p>The next four parameters are used to generate records in the TRADE_TRXN_CORRECTION table, which record when a correction to a field of an execution, trade, or order occurs. The fields to be checked for corrections should be specified in a comma separated list of business field names. Business field names can be found in the corresponding XML data map file in the datamaps directory.</p>		
FDT.DeriveCorrectionFields.Trade	Specifies which fields of a trade are monitored for corrections.	UnitQuantity, PriceIssuing
FDT.DeriveCorrectionFields.Execution	Specifies which fields of an execution are monitored for corrections.	UnitQuantity, PriceIssuing
FDT.DeriveCorrectionFields.DerivedTrade	Specifies which fields of a derived trade are monitored for corrections.	YieldPercentage, YieldMethodCode
FDT.DeriveCorrectionFields.Order	Specifies which fields of an order are monitored for corrections.	LimitPriceIssuing, UnitQuantity
<b>XDP:</b> Specifies properties used to configure the Pre-processor (XDP) component.		
XDP.Default.ArchiveFlag	Specifies whether to archive data files. The system copies input files to the backup directory (TRUE) or deletes input files (FALSE).	TRUE

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
XDP.Default.ErrorLimit	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
XDP.Default.TargetDir	Specifies the directory in which to place the resulting output file. If this is blank (the default), output files reside in the corresponding load directory (a subdirectory of <code>market/load</code> or <code>firm/load</code> depending on the schema of the data being processed).	
XDP.Default.SequenceBatchSize	Specifies the batch size when retrieving sequence IDs.	100000
XDP.Default.AdditionalOutput	Specifies a directory to contain the output file in addition to the target directory.	
XDP.Default.DoFileLookups	Specifies whether to do reference data lookups for fields that arrive as part of an input file (TRUE) or not do them (FALSE).	FALSE
XDP.Default.DiscardLookupFailures	Specifies whether to discard records that fail a reference data lookup (TRUE) or just log a message (FALSE).	FALSE
XDP.Default.ValidatorClass	Specifies the Java class used to validate records of a given data type. Use of subclasses occurs when the general functionality of <code>AbstractValidator</code> is not enough for a given data type.	<code>AbstractValidator</code>
XDP.Default.AdapterClass	Specifies the Java class used to process records of a given data type. Use of subclasses occurs when the general functionality of <code>BaseFileAdapter</code> is not enough for a given data type.	<code>BaseFileAdapter</code>
XDP.Default.NumberOfThreads	Specifies the number of worker threads to be used when Pre-processing a file	2
XDP.Default.BufferSize	Allows an override to the <code>BufferSize.value</code> property for the XDP.	100
XDP.Default.InputFileCharset	Specifies the character set of the DIS input files provided by the client. Currently, the only supported character sets are those that are compatible with ASCII.	UTF8
XDP.Default.SupplementalType	Specifies an additional file type that a given XDP will create when it processes a file of the given type.	<code>TrustedPairMember</code>
XDP.Account.DeriveAccountToPeerGroup	When processing Account records, specifies whether to derive an <code>AccountToPeerGroup</code> record if the <code>AccountPeerGroupIdentifier</code> field is populated.	

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
XDP.EmployeeTradingRestriction.DescendOrgTree	If an Employee Trading Restriction record contains an Organization Identifier, then it specifies: <ul style="list-style-type: none"> <li>• Whether to create Employee Trading Restriction records for all employees in the organization and all the related child organizations defined in the Organization Relationship file (TRUE)</li> </ul> or <ul style="list-style-type: none"> <li>• Whether to create records only for employees in the specified organization (False).</li> </ul>	FALSE
XDP.<Data Type>.<Property>	Overrides the given property for the given Pre-processor instance.	
<b>XDL:</b> Specifies properties used to configure the Data Loader (XDL) component.		
XDL.Default.FullRefresh	Is valid for data types that have a load operation of <i>Overwrite</i> as defined in the DIS. This parameter specifies replacement of the entire table (TRUE) or provision of deltas (FALSE).	TRUE
XDL.Default.DataFileExts	Specifies the possible file extensions for an input file.	.XDP, .FDT, .MDT, .XDT
XDL.Default.CommitSize	Specifies the number of records to update or insert before committing (not used when Direct=TRUE).	500
XDL.Default.ErrorLimit	Specifies the number of rejected records to allow before exiting with an error. If left blank (the default), processing sets no limit.	
XDL.Default.DbErrorCodes	Specifies a comma-separated list of database vendor-specific 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
The following properties apply only to the Oracle adapter.		
XDL.Default.MaxBindSize	Specifies the maximum number of bytes (integer) to use in the bind array for loading data into the database.	4194304
XDL.Default.Direct	Specifies whether to use direct path loading (TRUE) or conventional path loading (FALSE).	FALSE
XDL.Default.Parallel	Specifies whether a direct path load will be done in parallel (TRUE). This will be the case when multiple loaders for the same data type are run in parallel, such as with multiple ingestion instances.	FALSE



**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
XDL.Default.Unrecoverable	Specifies whether a direct path load does not use redo logs (TRUE) or uses redo logs (FALSE).	FALSE
XDL.Default.Partitioned	Specifies whether a direct path load uses the current date partition (TRUE) or any partition (FALSE).	FALSE
XDL.Default.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 Data Loader.	FALSE
XDL.Default.SkipIndexErrorCode	Specifies a database vendor specific error code that occurs in the log file when skipping indexes.	26025
XDL.Default.IndexParallelLevel	Specifies the parallel level of an index rebuild (that is, number of concurrent threads for rebuilding an index).	4
XDL.Default.DoAnalyze	Specifies whether to run a stored procedure to analyze a database table after loading data into it.	FALSE
XDL.Default.DoImportStatistics	Specifies whether to run a stored procedure to import statistics for a database table after loading data into it.	FALSE
XDL.Default.ImportStatisticsType	Specifies the type of statistic import to perform if DoImportStatistics has a value of True.	DLY_POST_LOAD
XDL.Default.ImportStatisticsLogDir	Saves the directory to which the stored procedure writes the log file if DoImportStatistics has a value of True. This log directory must reside on the server that hosts the database.	/tmp
XDL.Default.TableDoesNotExistErrorCode	Specifies the database error code that indicates a database table does not exist.	942
XDL.Default.UseUpdateLoader	Specifies whether JDBC updates should be used instead of a delete/insert when updating a database record. This is only valid for data types that have a load operation of Update.	FALSE
XDL.<Data Type>.<Property>	Overrides the specified property for a given Data Loader instance.	
<b>IMC:</b> Specifies properties for configuring the Directory Cleanup (IMC) component.		
Directory[1..N].Name	Identifies the directory to clean up. The system removes date subdirectories (in YYYYMMDD format) from this directory.	../data/backup
Directory[1..N].DaysToKeep	Specifies the number of days to keep for this directory. The system does not delete date subdirectories with the latest dates.	3

**Table 83. DataIngest.xml File Configuration Parameters (Continued)**

Property Name	Description	Example
<b>DBUtility:</b> Specifies properties used to configure various utility processes. Valid utility names are SecurityMarketDaily, SecurityFirmDaily, AccountChangeLogSummary, CustomerChangeLogSummary, AccountToCustomerChangeLogSummary.		
<UtilityName>.NumberOfThreads	Specifies the number of worker threads that the give component uses when processing data.	4
<UtilityName>.SequenceBatchsize	Specifies the batch size when retrieving sequence IDs for records generated by given component.	10000
<b>Watch List Service:</b> Specifies properties used to configure the Scan Watch List Web Service.		
Timeout.value	Specifies how many seconds a call to the Watch List Service made through the scanWatchList.sh script will wait for the service request to finish. This value should be set to the longest wait time expected based on the volume of data and system configuration. Setting it very high will not affect performance since the call will return as soon as it is complete.	600
Log.UseDateLog	Overrides the default Log.UseDateLog property.	FALSE
WatchListScannerClass.value	Identifies the Java class used to scan a watch list for a given name.	MantasWatchListScanner
NameMatcherClass.value	Identifies the Java class used to match a name against a list of names.	FuzzyNameMatcher
FuzzyMatcher.SecondToPoll	Identifies the number of seconds to wait between querying the WATCH_LIST table for new names that are added by the Watch List Management Utility.	
FuzzyMatcher.MaximumAddedNames	Identifies the maximum number of names that can be added to the Watch List Service after it is initialized. If additional names must be added, the service must be re-initialized.	

### Data Ingest Custom XML Configuration File

Oracle clients can modify the DataIngest.xml file to override default settings that the system provides. However, this file is subject to change in future OFSBD releases. Therefore, upon installation of a newer OFSBD version the client must reapply any modifications in the current DataIngest.xml file to the newer DataIngest.xml file.

To simplify this process, the DataIngestCustom.xml file is available for use. This file holds all site-specific changes to the DataIngest.xml file. The client can override any settings in DataIngest.xml by placing the modifications in DataIngestCustom.xml. After installing a newer OFSBD version, the client can copy the older DataIngestCustom.xml file to DataIngestCustom.xml in the new installation.

## data Subdirectory

The `data` subdirectory within the `ingestion_manager` directory contains additional subdirectories for organizing Market data files and Oracle client data files. The system creates these files during the Pre-processing, transformation and data-loading stages of the ingestion process. The Market data and Oracle client data files appear in subdirectories that are indicative of the processing stages (or workflow steps) that the Data Management subsystem components perform. The following sections describe the contents of each subdirectory and the components that read or write to each subdirectory.

**Note:** Processing date stamps should appear as YYYYMMDD for Data Management directories and subdirectories. The system provides this processing date to the `set_mantas_date.sh` shell script when starting the first batch for the day.

### data/errors Subdirectory

The `errors` subdirectory within the `data` subdirectory stores error files that Data Management subsystem components create or move upon detection of errors during file processing. The system places error files in subdirectories within the `errors` subdirectory. These error file subdirectories are name-based on the processing date for the files that they contain. The date has the format YYYYMMDD, where YYYY is the four-digit year, MM is the two-digit month, and DD is the two-digit day. The files in the `errors` subdirectory have the same name as the file in which the error was detected. However, the component that identified the errors appends its extension to the end of the file.

The following table identifies the error file signatures that each component can output to the `errors` subdirectory.

**Table 84. Error File Signatures Output by Component**

Component	Error File
Pre-processor	<data type>_*.XDP.err
Data Loader	<data type>_*.XDL.err
FDT	Order_*.FDT.err TradeExecution_*.FDT.err
MDS	InsideQuote_*.MDS.err MarketCenterQuote_*.MDS.err ReportedMarketSale_*.MDS.err

The IMC utility, `runIMC.sh`, cleans up the `errors` subdirectory. The IMC's configuration file defines the number of days that error files age before their removal.

### data/backup Subdirectory

The `backup` subdirectory stores files that Data Management subsystem components processed and require no further processing. That is, they are considered to be in a final form after successful processing.

- Transformers back up files that they receive and create.
- Loaders back up files that they finished loading. Each file in the backup directory appears in a subdirectory with the date as its name. The name is in the format YYYYMMDD, where YYYY is the four-digit year, MM is the two-digit month, and DD is the two-digit day.

The IMC component, `runIMC.sh`, cleans up the backup subdirectory. The IMC's configuration file defines the number of days that backup files age before removal. The following table references the files that the system writes to the backup subdirectory, by component.

**Table 85. Backed Up Files by Component**

Component	Data Files
FDT	*.XDP
Data Loader	*.XDP, *.FDT

### data/firm Subdirectory

The `firm` subdirectory within the `data` subdirectory contains the `extract`, `transform` and `load` subdirectories that correspond directly to the workflow steps that Firm data moves through during Data Management. The following sections describe each subdirectory.

#### extract Subdirectory

The `extract` subdirectory within the `firm` subdirectory contains checkpoint data and working files for each Pre-processor during Pre-processing.

Each Pre-processor also maintains checkpoint files that enable it to recover after a failure and without the loss of data integrity; an FDT removes the files after it successfully Pre-processes its data. When finished, each Pre-processor moves its final Pre-processed files to either the `transform` subdirectory for processing by FDT, or to the `load` subdirectory for loading into the database.

The `.XDP` file type identifies files that the Pre-processor creates.

#### transform Subdirectory

The `transform` subdirectory within the `firm` subdirectory contains the FDT's checkpoint data and working files during transformation. When finished, the FDT moves its final transformed Firm data files to the `load` subdirectories for loading into the database. The system writes the transformed data to files and then moves the files to the `load` subdirectory. The `.FDT` file type identifies the files that the FDT creates.

The FDT also maintains several checkpoint files that allow it to recover after a failure, without the loss of data integrity.

#### load Subdirectory

The `load` subdirectory within the `firm` subdirectory contains additional subdirectories that contain Pre-processed and transformed Firm data that the system queues for loading into the database. Each loader component monitors its respective subdirectory (that is, data queue) looking for data to load into the database—a subdirectory exists for each kind of Oracle client data that processing loads into the database. After loading data files into the database, each loader moves the processed files to the backup subdirectory.

#### inbox Subdirectory

The `inbox` subdirectory within the `ingestion_manager` directory is an electronic mailbox or queue in which the Oracle client writes its data files for subsequent processing by Data Management subsystem Data Pre-processor components. Each Market or Firm Data Pre-processor retrieves the file it is assigned to process from the `inbox`

subdirectory and then moves the file to the appropriate extract subdirectory for Pre-processing. The DIS describes the naming convention and content of each data file that an Oracle client provides.

## logs Subdirectory

The `logs` subdirectory contains a log file for each component running on a host computer. Each log file in the `logs` subdirectory appears in a subdirectory with the date as its name, in the format `YYYYMMDD`, where `YYYY` is the four-digit year, `MM` is the two-digit month, and `DD` is the two-digit day. The subdirectory's date is based on the processing date for data to which the log files pertain.

The IMC utility, `runIMC.sh`, cleans up the `logs` subdirectory. The IMC utility's configuration file defines the number of days that log files age before their removal. The following table identifies log files for each component, based on the file name's prefix.

**Table 86. Log Files Output by Component**

Prefix	Component
XDP	Pre-processor
XDL	Data loader
FDT	File Data Transformer
IMC	IMC

## BD Directory Structure

The BD Datamap component is organized as subdirectories below the `<OFSAAI_Installed_Directory>/bdf` file. The following table provides details about each subdirectory..

**Table 87. Directory Structure Description**

Directory Name	Description
<code>scripts</code>	Shell scripts for running BD components, setting the environment, and changing passwords
<code>logs</code>	Log files containing status and error messages produced by BD components
<code>config</code>	Files used to configure BD components
<code>config/datamaps</code>	XML files containing data map definitions for individual BD components
<code>jars</code>	Java Archive (JAR) files used to run BD components
<code>data/errors</code>	Files containing error records produced by BD components
<code>data/temp</code>	Temporary files produced by BD components
<code>inbox</code>	Data files provided by the Oracle client in DIS format
<code>fuzzy_match</code>	C++ library files used for the purpose of fuzzy matching names

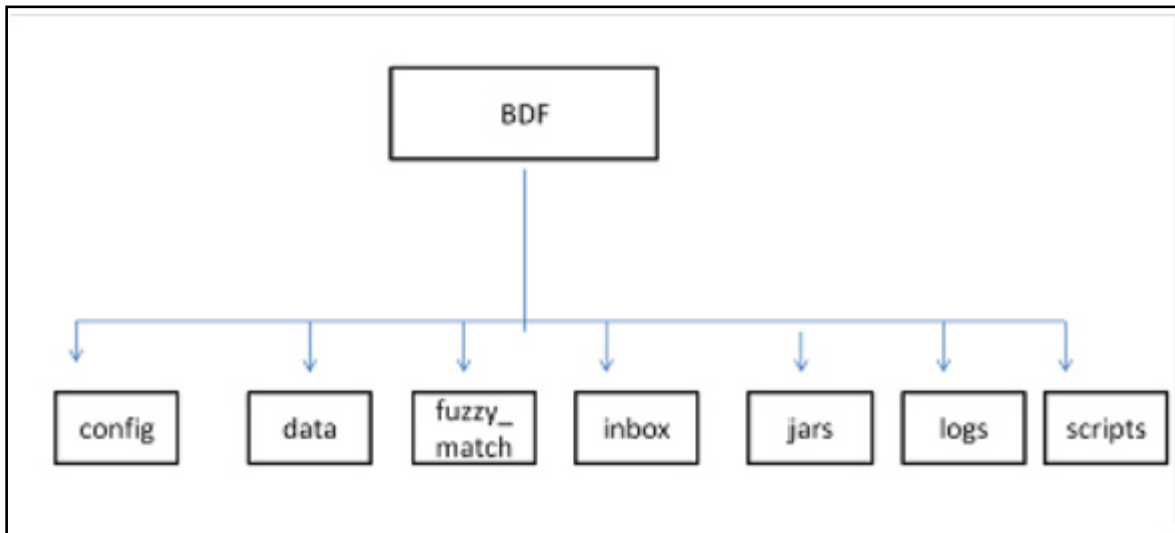


Figure 61. BD Subsystem Directory Structure

The following sections describe the BD directory structure.

## Scripts

The scripts folder contains the following files:

- **changePassword.sh** - Changes passwords used during the execution of BD components. Refer to the *Installation Guide* for more information.
- **env.sh** - Sets up the shell environment of BD components
- **execute.sh** - Executes BD 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 BD 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 BD subsystem improves performance time.

## 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
```

## Parameters

Parameters in BD 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
```

## 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 BD 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 BD components.
- The derivations subdirectory contains SQL derivations for individual fields.
- The queries subdirectory contains SQL queries for individual data maps.



## BDF.xml Configuration Parameters

The following table describes the BD properties configurations mentioned in the <OFSAAI Installed Directory>/bdf/config/BDF.xml file.

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
<b>MISCELLANEOUS</b>		
NumberOfThreads	The number of worker threads used by some BD 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.	~
<b>DB:</b> Parameters related to database access.		
DB.Connection.Driver	The JDBC driver class name.	oracle.jdbc.O racleDriver
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
<b>Directory:</b> 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 BD components will reside. This includes log files, error files, and any temporary processing files.	..
<b>Log:</b> Parameters used to configure the common logging module		
Log.Format	Identifies the log formatting string.	%d [%t] %p - %m%n
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 BD log file (logs/YYYYMMDD/Component.log). If the value is not provided, considers the default BD log location.	false

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
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 BD log file (logs/YYYYMMDD/Component.log).  If the value is not provided, considers the default BD 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 BD log file (logs/YYYYMMDD/Component.log). If the value is not provided, considers the default BD 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 BD log file (logs/YYYYMMDD/Component.log).	
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 BD log file (logs/YYYYMMDD/Component.log).	
Log.FATAL.Synchronous	Specify whether logging for a particular level should be performed synchronously or asynchronously.	false
<b>Load:</b> Parameters used to configure common Loading data		
Load.FullRefresh	For DIS files defined as Overwrite, whether to fully replace FSDM 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

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
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
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
<b>DIS:</b> Parameters related to processing DIS files		
DIS.Source	The mechanism used to load DIS data.  <b>FILE:</b> DIS files will be provided and will be loaded using SQL*Loader processes running on the application server.  <b>FILE-EXT:</b> DIS files will be provided and will be loaded using external tables with the DIS files accessed directly by the database.  <b>FSDW:</b> 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

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
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
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
<b>Parameters used by queries defined in the data maps:</b>		
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 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
BankCodeListWithIA	<p>Defines the List of Financial Institution Identifier Types, these are type of unique identifiers which are used to represent the financial institutions.</p> <p>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</p> <ul style="list-style-type: none"> <li>● BIC: SWIFT Bank Identifier Code (BIC)</li> <li>● CHU: CHIPS Participant User Identifier</li> <li>● CO: Corporate Identifier</li> <li>● CHP: CHIPS Participant Identifier</li> <li>● FED: Federal Reserve Routing (ABA) Number</li> <li>● CU: Customer Identifier</li> <li>● GL: General Ledger Account</li> <li>● IA: Internal Account Identifier</li> </ul>	<p>BIC,FED,CH P,CHU, DTC,CDL,EP N,KID, CBI,CSN,OT F,BLZ,I BAN,ABLZ,B SB,CP AP, SDIC, HEBIC, BCHH, NSC, IFSC, IDIC, PNCC, RCBIC, UKDSC, Swiss BC, Swiss SIC,IA</p>
BankCodeList	<p>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).</p> <p>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</p> <ul style="list-style-type: none"> <li>● BIC: SWIFT Bank Identifier Code (BIC)</li> <li>● CHU: CHIPS Participant User Identifier</li> <li>● CO: Corporate Identifier</li> <li>● CHP: CHIPS Participant Identifier</li> <li>● FED: Federal Reserve Routing (ABA) Number</li> <li>● CU: Customer Identifier</li> <li>● GL: General Ledger Account</li> </ul>	<p>BIC,FED,CH P,CHU, DTC,CDL,EP N,KID, CBI,CSN,OT F,BLZ,I BAN,ABLZ,B SB,CP AP, SDIC, HEBIC, BCHH, NSC, IFSC, IDIC, PNCC, RCBIC, UKDSC, Swiss BC, Swiss SIC</p>
IdRiskWinLevel	<p>Defines the Risk level to calculate Effective Risks for internal parties (Account/ Customer).</p> <p>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.</p>	<p>1</p>
InternalAccountCodeList	<p>Codes to define types of Internal Entities with client, for example:</p> <ul style="list-style-type: none"> <li>● IA: Internal Account Identifier</li> <li>● GL: General Ledger Account</li> </ul>	<p>IA, GL</p>
ExternalEntityCodeList	<p>Codes to define types of External Entities with client, for example:</p> <ul style="list-style-type: none"> <li>● XA: External Account Identifier</li> <li>● CO: Corporate Identifier</li> <li>● DL: Driver License</li> <li>● IBAN: International Bank Account Number</li> </ul>	<p>XA,CC,CO,D L,GM, GP,LE,MC,N D,NR, PP,SS,TX,AR ,OT,IB AN</p>

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
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.	
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.	

**Table 88. BDF.xml File Configuration Parameters**

Parameter Name	Description	Example
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

### **BD 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 89. BD Datamap Configuration Parameters**

Property Name	Description	Example
DB.Connection.URL	Database URL for JDBC connections made by BD 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 <code>bdf/scripts/changepassword.sh</code>	
DB.Schema.MANTAS	Schema name for the Oracle ATOMIC database schema. BD 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 as:

`<OFSAAI Installed Directory>/bdf/config/<component>.xml`

And can also be created in the following:

`<OFSAAI Installed Directory>/bdf/config/custom/<component>.xml`



# *Processing Derived Tables and Fields*

This appendix covers the following topics:

- [Ingestion through Batches](#)
- [Derivations](#)
- [Ingestion Timeline - Intra-Day Ingestion Processing](#)
- [Guidelines for Duplicate Record Handling](#)
- [Data Rejection During Ingestion](#)
- [Alternatives to Standard Data Management Practices](#)

## **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/ofsaapp/FCCM806
- **INGESTION\_HOME:** The path under installed area pointing to the ingestion\_manager subsystem.  
For Example: /scratch/ofsaapp/FCCM806/ingestion\_manager
- **DB\_TOOLS\_HOME:** The path under installed area pointing to database subsystem.  
For Example: /scratch/ofsaapp/FCCM806/database/db\_tools
- **BDF\_HOME:** The path under the installed area pointing to the BD subsystem.  
For Example: /scratch/ofsaapp/FCCM806/bdf

**Note:** BDF\_HOME should be exported only if Ingestion has to be run through the BD 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*.

Sample customized script for execute.sh is given below:

```
#!/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

```

The above script is used to trigger BD 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.

## 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 90. 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	
BC	MutualFundFamilyAccountPosition	MUTUAL_FUND_FAM_ACCT_PO SN	
BC	RegisteredRepresentativeCommissionProfile	RGSTD_REP_CMSN_SMRY	

**Table 90. Utilities**

Product	Utility Name	Table Name	Predecessor
BC	RegisteredRepresentativeCommissionProduct MixProfile	RGSTD_REP_CMSN_PRDCT_SM RY	
ECTC	EnergyFlowDailyProfile	ENERGY_FLOW_SMRY_DAILY	Energy Flow

### 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
```

## 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.

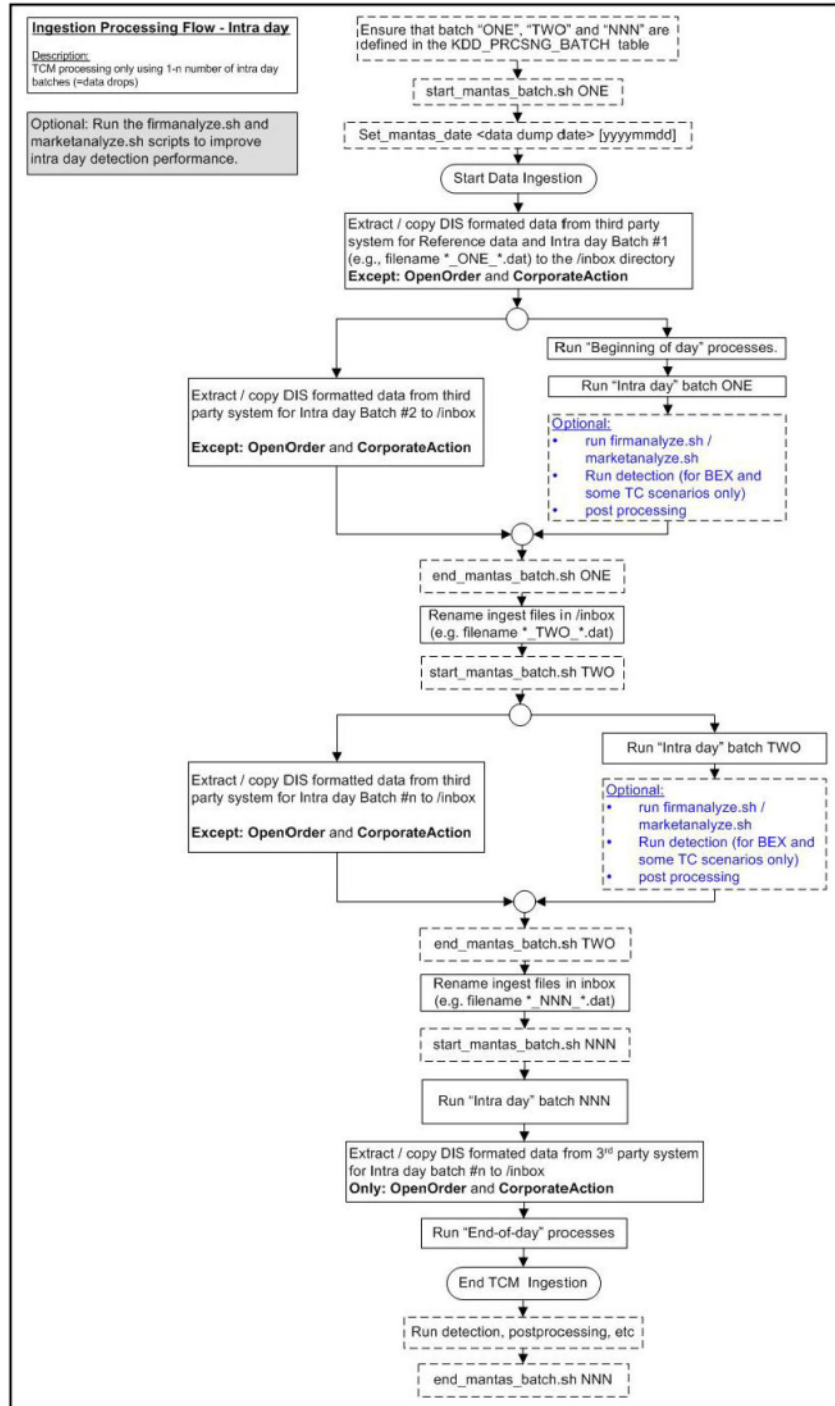


Figure 62. Intra-Day Data Management Processing

Intra-day processing references different processing groups as Figure 62 illustrates, such as beginning-of-day processing and intra-day processing. Multiple batches run throughout the day. As in Figure 62, 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_PRCNG_BATCH` table.

**Table 91. 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

## ***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.

## ***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).

## 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.

### 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.

### 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.

### 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.

### 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.

## 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.

### **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.

### **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 timestamp prior to the timestamp 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 timestamp 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.

## 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).

## *Alternatives to Standard Data Management Practices*

### 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.

### Fuzzy Name Matcher Utility

During BD 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.

### 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.

### 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 BD Datamaps (`NameMatchStaging.xml`, `RegOToBorrower.xml`) file in folder (`<OFSAAI Installed Directory>/bdf/config/datamaps`). For more information, refer *Chapter 3, 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/ofsaapp/BD805/BD805/bdf/data/temp

fuzzy_name.B.stopword_file=/scratch/ofsaapp/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/ofsaapp/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 63. 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 92. Fuzzy Name Matcher Utility Configuration Parameters**

Parameter	Description
<code>fuzzy_name.stopword_file</code>	Identifies the file that stores the stop word list. The stop word file is either corporate or personal. The <code>&lt;prefix&gt;</code> token identifies corporate as <i>B</i> and personal as <i>P</i> . Certain words such as <i>Corp, Inc, Mr, Mrs, or the</i> , do not add value when comparing names.
<code>fuzzy_name.match_threshold</code>	Indicates the score above which two names are considered to match each other. The utility uses this parameter only when the <code>match_multi</code> property has a value of <code>true</code> . The allowable range is from 0 to 100.
<code>fuzzy_name.initial_match_score</code>	Specifies the score given for matching to an initial. The allowable range is 0 to 100; the recommended default is 75.
<code>fuzzy_name.initial_match_p1</code>	Specifies the number of token picks that must be made before awarding <code>initial_match_score</code> . The value is an integer $\geq 0$ . The default value is 2.
<code>fuzzy_name.initial_match_p2</code>	Specifies the number of token picks that must be made before awarding <code>initial_match_score</code> if only initials remain in one name. The value is an integer $\geq 0$ . The default value is 1.
<code>fuzzy_name.extra_token_match_score</code>	Indicates the score given to extra tokens. The allowable range is 0 to 100; the recommended default is 50.
<code>fuzzy_name.extra_token_min_match</code>	Specifies the minimum number of matches that occur before awarding <code>extra_token_match_score</code> . The range is any integer $\geq 0$ . The recommended setting for corporations is 1; for personal names is 2.
<code>fuzzy_name.extra_token_pct_decrease</code>	Determines the value of the <code>extra_token_match_score</code> parameter in regard to extra tokens. If multiple extra tokens are present, reduction of <code>extra_token_match_score</code> occurs for each additional extra token. The utility multiplies it by this number. For example, if <code>extra_token_match_score = 50</code> , and <code>extra_pct_decrease</code> 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).
<code>fuzzy_name.first_first_match_score</code>	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 $\geq 0$ . The recommended value for corporate names is 1.0; for personal names, 0.0.
<code>fuzzy_name.match_multi</code>	Determines how to handle multiple matches above the <code>match_threshold</code> value. If set to <code>true</code> , the utility returns multiple matches. If set to <code>false</code> , it returns only the match with the highest score.
<code>fuzzy_name.file.delimiter</code>	Specifies the delimiter character used to separate each columns in the result file and target name list file.

**Table 92. Fuzzy Name Matcher Utility Configuration Parameters (Continued)**

Parameter	Description
fuzzy_name.min.intersection.first.letter.count	<p>Specifies the number of words per name whose first letters match. For example, if parameter value = 1 only the first letter of the first <b>or</b> last name would have to match to qualify. If the value = 2, the first letter of <b>both</b> the first and last name would have to match to qualify.</p> <p><b>Warning:</b> 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.</p>
fuzzy_name.default.prefix	For entries that are not specified as business or personal name, default to this configuration set.
fuzzy_name.max.names.per.process	<p>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.</p>
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.

### 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>
```

### 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.

## 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:

1. Complete the prerequisites for using control data (refer to section *Prerequisites for Using Control Data* on page 279 for more information).
2. Prepare for ingestion of the control data (refer to section *Control Data Management* on page 279 for more information).
3. Install the control data (refer to section *Loading Control Data Thresholds* on page 280 for more information).
4. Run Behavior Detection on control data to generate events (refer to section *Running Behavior Detection on Control Data* on page 280 for more information).

## Prerequisites for Using Control Data

Before you use control data to test your installation, the following prerequisites must be fulfilled:

1. 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.
2. The current day according to control data is 20151210.
3. Unless specified, set the current date as 20151210, to generate events on control data, before running TBAML.

**Note:** For more information about control data on your site, contact your Oracle Administrator.

## 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 93. Dates used by Control Data**

20141231	20151123
20150130	20151124
20150227	20151125
20150331	20151126
20150430	20151127
20150529	20151130
20150630	20151203
20150731	20151204

Table 93. Dates used by Control Data

20150831	20151208
20150930	20151209
20151030	20151210
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 (refer to section *inbox Subdirectory* for more information).
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.

## 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.

## 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.)

### Important Notes

1. Run loaded scenarios with the system date as 20151210 with the following exceptions:
  - a. For Portfolio Pumping scenario, the system date must be 20151204

- b. 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.





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](#)

**Note:** Oracle recommends all datamaps are run in the order described in the following tables.

## ***Trade Finance Datamaps***

### **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 94. Trade Finance - Pre-Watch List Datamaps**

<b>Datamap Number</b>	<b>Datamap Name</b>	<b>Predecessors</b>
60200	TradeFinanceContractEvent.xml	NA
60210	TradeFinanceContractEventAcknowledgementStage.xml	NA
60220	TradeFinanceContractAmendmentStatusStage.xml	NA
60230	TradeFinanceContractEvent_AcknowledgeUpd.xml	60200 60210

**Table 94. Trade Finance - Pre-Watch List Datamaps**

Datamap Number	Datamap Name	Predecessors
60240	TradeFinanceContractEvent_AmendmentUpd.xml	60200 60220
60250	TradeFinanceContract.xml	60200
60260	TradeFinancetoAccount.xml	NA
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
60370	TradeFinancePartyTF_DerivedAddressUpd.xml	60360 60350 60310 60300
60380	TradeFinancePartyDC_DerivedAddressUpd.xml	60370 60360 60350 60310 60300
60390	FinancialInstitution_TradeFinanceParty.xml	60300 60310
60400	DerivedEntity_TradeFinancePartyInsert.xml	60300 60310
60410	DerivedEntity_TradeFinancePartyUpd.xml	60300 60310 60400

**Table 94. 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
60590	BrokerageDistribution_FinancialInstnUpd.xml	60570 60560 60550 60390 60300 60310
60600	DerivedAddress_TradeFinanceBrokerageDistributionStageInsert.xml	60580 60570 60560 60550 60390 60300 60310

**Table 94. Trade Finance - Pre-Watch List Datamaps**

Datamap Number	Datamap Name	Predecessors
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	DocCollectionContractAcknowledgementStage.xml	NA
60650	DocumentaryCollectionContractAcceptanceStage.xml	NA
60660	DocumentaryCollectionContractEvent_AcknowledgeUpd.xml	60620 60630
60670	DocumentaryCollectionContractEvent_AcceptanceUpd.xml	60620 60640
60680	DocumentaryCollectionDiscrepancyDetail.xml	NA
60690	DocumentaryCollectionDiscrepancyDetail_DiscrDtUpd.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
	CountryTradeList	
	GoodsorService	

### 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 95. Trade Finance - Post-Watch List Datamaps**

Datamap Number	Datamap Name	Predecessors
60730	DocumentaryCollectionContract_LiquidationUpd.xml	60720
	TradeFinancePartyTF_DerivedAddressUpd	
	TradeFinancePartyDC_DerivedAddressUpd	
60740	TradeFinancePartyTF_EntityActivityRiskUpd.xml	NA
60750	TradeFinancePartyDC_EntityActivityRiskUpd.xml	NA
60760	ExternalOrganization_ExternalEntitySeqUpd	60200 60210
60770	ExternalOrganization_EntityRiskInsert	60200 60220
	TradeFinanceContractEvent_ActivityRskUpd	
	DocumentaryCollectionContractEvent_ActivityRskUpd	
	CountryTradeList	
	GoodsorService	
	TradeFinanceGoodorService_Upd	

## ***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 96. 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 96. 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
	CorrespondentBankToPeerGroup	
10330	DerivedAddress_SettlementInstructionInsert	NA
10340	DerivedAddress_SettlementInstructionUpd	NA
10350	SettlementInstruction_PhysicalDlvryAddrUpd	NA
10360	DerivedAddress_FrontOfficeTransactioPartyStageInsert	NA
10370	DerivedAddress_FrontOfficeTransactioPartyStageUpd	NA
10380	FrontOfficeTransactionParty_DerivedAddress	10360 10370
	DerivedAddress_InsuranceTransactionInsert	
	DerivedAddress_InsuranceTransactionUpd	
	InsuranceTransaction_InstitutionAddrUpd	
	DerivedEntity_InsuranceTransactionInsert	
	DerivedEntity_InsuranceTransactionUpd	

**Table 96. Watch List Datamaps (Continued)**

<b>Datamap Number.</b>	<b>Datamap Name</b>	<b>Predecessors</b>
10390	DerivedEntity_FrontOfficeTransactionPartyInsert	10080 10090
10400	DerivedEntity_FrontOfficeTransactionPartyUpd	10080 10090
10410	DerivedEntity_SettlementInstructionInsert	10220 10230 10240
10420	DerivedEntity_SettlementInstructionUpd	10220 10230 10240
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
10490	LinkStaging_FrontOfficeTransactionParty	10360 10370 10380 10390 10400 10485
	LinkStaging_InsTrxnDerivedEntDerivedAdd	

Table 96. Watch List Datamaps (Continued)

Datamap Number.	Datamap Name	Predecessors
10500	LinkStaging_InstructionDerivedEntDerivedAdd	10330 10340 10350 10410 10420
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	DerivedAddressstoInternalAccount_LinkStage	10490 10500
10570	WatchListStagingTable2_WatchListStage2AcctExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
10580	WatchListStagingTable2_WatchListStage2CBExistence	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440



**Table 96. Watch List Datamaps (Continued)**

Datamap Number.	Datamap Name	Predecessors
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
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

**Table 96. Watch List Datamaps (Continued)**

Datamap Number.	Datamap Name	Predecessors
10630	WatchListStagingTable2_AcctListMembershipUpd	10450 10460 10470 10480 10390 10400 10510 10520 10410 10420 10430 10440
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 96. 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
10700	WatchListStagingTable2_WatchListStage2SeqIdUpd	10570 10580 10590 10600 10610 10620 10630 10640 10650 10660 10670 10680 10690

**Table 96. Watch List Datamaps (Continued)**

Datamap Number.	Datamap Name	Predecessors
10710	WatchListStagingTable2_WatchListStage2IntrIdUpd	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
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

**Table 96. Watch List Datamaps (Continued)**

Datamap Number.	Datamap Name	Predecessors
10860	DerivedEntityListMembership_WatchListStage2Insert	10700 10710
10870	DerivedEntityListMembership_WatchListStage2Upd	10700 10710
10875	Account_EffectiveRiskFactorTxtUpd	10700 10701
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_WatchListStage2IntrId	10320 10700 10710
10920	BackOfficeTransaction_EffectiveAcctivityRiskUpd	10890 10900
10930	SettlementInstruction_EntityAcctivityRiskUpd	10890 10900
	FrontOfficeTransactionPartyRiskStage_EntityRisk	
	FrontOfficeTransactionPartyRiskStage_ActivityRisk	
10940	FrontOfficeTransactionPartyRiskStage_EntityActivityRiskInsert	10890 10900
	InsuranceTransaction_EntityAcctivityRiskUpd	

**Note:** Datamaps 10970,10980,10990, 11000,11010,11020 can be run in parallel.

**Table 97. AML Banking - Post-Watch List Datamaps**

<b>Datamap Number</b>	<b>Datamap Name</b>	<b>Predecessors</b>
20010	CorrespondentBank_JurisdictionUpd	10430 10440
20020	CorrespondentBank_AcctJurisdictionReUpd	10430 10440
20030	FinancialInstitution_InstNameUpd	10430 10440
	AccountGroup_InvestmentObjectiveUpd	
10960	AccountGroup_JurisdictionUpd	NA
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
11030	CashTransaction_FrontOfficeTransactionRevAdj	10970 10980 10990 11000 11010 11020
11040	MonetaryInstrumentTransaction_FrontOfficeTransactionRevAdj	10970 10980 10990 11000 11010 11020

**Table 97. AML Banking - Post-Watch List Datamaps (Continued)**

Datamap Number	Datamap Name	Predecessors
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
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

**Table 97. AML Banking - Post-Watch List Datamaps (Continued)**

Datamap Number	Datamap Name	Predecessors
	InsuranceTransaction_TrustedFlagsUpd	
11140	MonetaryInstrumentTransaction_TrustedFlagsUpd	11060 11070 11080 11090 11100 11110
11150	WireTransaction_TrustedFlagsUpd	11060 11070 11080 11090 11100 11110

## Processing BD Datamaps

The following table provides a list of datamaps and description for each datamap. These datamaps are listed in order.

**Table 98. BD Datamaps**

Datamap Number	Datamap Name	Description
10100	AccountManagementStage	This datamap identifies the relationship between accounts and the employees who have a management role on that account. Management roles include positions such as Financial Advisor, Banker, and Registered Representative.
10112	ServiceTeam_SprvsncdUpd	This datamap updates service team table with the Employee Maximum Supervision Code.
10113	InvestmentAdvisor_MangdAcctUpd	This datamap updates ManagedAccountNetworth and ActiveSubAccountCount column in InvestmentAdvisor table.
60010	PortfolioManagerPosition	The datamap is used to populate the portfolio manager positions. It reads tables (Account and Account Position), populated while executing Pre-processors and creates records to populate the PORTFOLIO_MGR_POSN table.
10114	Security_CIRRatingUpd	This datamap derives the column CIRRating and updates back to Security table.
60020	AccountGroupProductAllocation	The datamap captures the actual proportionate distribution of holdings for an account group aggregated by reporting classifications.
60030	AccountProductAllocation	The datamap captures the actual proportionate distribution of holdings for an account aggregated by product classifications.
60040	UncoveredOptionExposureDaily	This datamap derives the value from the uncvrd_optns_smry_dly table and insert/updates the records in UNCVRD_OPTNS_EXPOSURE_DLY table.



**Table 98. BD Datamaps (Continued)**

<b>Datamap Number</b>	<b>Datamap Name</b>	<b>Description</b>
60050	InvestmentAdvisorProfile	This datamap updates the Investment Manager Summary Month table from the daily activity
60060	RegisteredRepresentativeProfile	This datamap updates the Registered Representative Summary Month table with daily activity
60070	RegOToBorrower	This datamap use the fuzzy match logic to match the Regulation O list against the Borrower.
60080	InterestedPartyToEmployee	This datamap use fuzzy matcher to match Interested Parties in Account Scheduled Event table against Employee name.
11160	AccountDailyProfile-Trade	This datamap performs daily aggregation of trades from trade table , Profit Loss from Account Realized Profit Loss table.
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.
10020	FinancialInstitution_ThomsonDataInstitutionInsert	This datamap builds the many-to-one relationship in INSTN_MASTER that is the relationships between bics and feds with INSTN_SEQ_ID. The INSTN_MASTER table gets populated from BANK_REFERENCE_STAGE table.
10030	AccountToClientBank_ThomsonDataInstitutionInsert	This datamap builds the many-to-one relationship in ACCT_ID_INSTN_ID_MAP that is the relationships between bics and feds with INSTN_SEQ_ID. The ACCT_ID_INSTN_ID_MAP table gets populated from BANK_REFERENCE_STAGE table.
10040	FinancialInstitution_AIIMSPopulation	This datamap inserts new records in Financial Institution table from the ACCT_INSTN_MAP_STAGE table, the datamap creates unique identifiers for banks based on the third party vendors.
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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
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.
	LoanDailyActivity_RepCurrencyUpd	
10110	LoanProfile_LoanProfileStage	This datamap is used to populate Loan Summary from LOAN_SMRY_MNTH_STAGE table. 1) Select set of information/columns from LOAN_SMRY_MNTH_STAGE table, if the record is new insert the details in LOAN_SMRY_MNTH else update the existing record.
	BackOfficeTransaction_UnrelatedPartyCodeUpd	
10116	BackOfficeTransaction_CollateralUpd	This datamap updates Collateral Percentage , Collateral Value for that transaction.
10120	BackOfficeTransaction_OriginalTransactionReversalUpd	This datamap handles reverserals 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'.

**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
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.
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).
10170	FinancialInstitution_AnticipatoryProfile	This datamap inserts new records in Financial Institution table for the institutions found in Anticipatory Profile table, INSTN_SEQ_ID are OFSAAI generated. This datamap should be executed before AccountToClientBank_AnticipatoryProfile datamap as generated INSTN_SEQ_ID will be used to populate Anticipatory Profile table.
10180	AccountToClientBank_AnticipatoryProfile	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 NTCPTRY_PRFL from the transaction data and loads them in the INSTITUTION_MASTER. 2) This data map finds the new institutions from the NTCPTRY_PRFL data and loads them in the ACCT_ID_INSTN_ID_MAP table using OFSAAI generated INTSN_SEQ_ID from INSTITUTION_MASTER.
10190	AnticipatoryProfile_AccountToClientBank	This datamap marks all institutions with an OFSAAI generated INTSN_SEQ_ID in the Anticipatory Profile tables. It should be executed after FinancialInstitution_AnticipatoryProfile and AccountToClientBank_AnticipatoryProfile datamaps are executed.
50020	DailyAggregateStage	This datamap populates DAILY_AGG_STAGE table with aggregated TRADE Data. DAILY_AGG_STAGE table in turn is used to populate OFFSETING_ACCT_PAIRS and TRADE_DAILY_TOT_CT_STAGE tables.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
50030	OffsettingAccountPairStage	This datamap is used to populate OFFSETTING_ACCT_PAIRS table by self-joining the table DAILY_AGG_STAGE to generate offsetting trade account pairs. The accounts have the lower ACCT_INTRL_ID while the offsetting accounts have the higher ACCT_INTRL_ID.
50040	TradeDailyTotalCountStage	This datamap aggregates the total trades done by that account for the current processing day.
10200	CustomerAccountStage_FrontOfficeTransactionParty	This datamap populates the Customer Account Stage table with the Cust-Acct pairs which appears in FOTPS with Party type as IA.
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.
10220	FinancialInstitution_SettlementInstruction	This datamap inserts new records in Financial Institution records for the institutions found in INSTRUCTION that have not been previously identified, INSTN_SEQ_ID are OFSAAI generated. This datamap should be executed before AccountToClientBank_SettlementInstruction datamap.
10230	AccountToClientBank_SettlementInstruction	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 INSTRUCTION from the transaction data and loads them in the INSTITUTION_MASTER. 2) This data map finds the new institutions from the INSTRUCTION data and loads them in the ACCT_ID_INSTN_ID_MAP table using OFSAAI generated INTSN_SEQ_ID from INSTITUTION_MASTER.
10240	SettlementInstruction_AccountToClientBank	This datamap updates Destination Institution and Physical Delivery Institution in INSTRUCTION table using the values from ACCT_ID_INSTN_ID_MAP table.
40010	FinancialInstitution_InsuranceTransaction	This datamap inserts new records in Financial Institution table for the institutions found in Insurance Transactions, INSTN_SEQ_ID are OFSAAI generated. This datamap should be executed before AccountToClientBank_InsuranceTransaction datamap as generated INSTN_SEQ_ID will be used to populate Anticipatory Profile table.
40020	AccountToClientBank_InsuranceTransaction	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 INSURANCE_TRXN data and loads them in the ACCT_ID_INSTN_ID_MAP table using OFSAAI generated INTSN_SEQ_ID from INSTITUTION_MASTER.

**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
40030	InsuranceTransaction_AccountToClientBank	This datamap marks all institutions with an OFSAAI generated Institution Identifier in Insurance Transaction records. 1) Prior to this datamap execution Financial Institution and Account To Client Bank records are inserted. 2) Henceforth this datamap uses the Account To Client Bank table and updates Institution Identifier in Insurance table.
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.
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_WatchListEntryUpdate	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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
10330	DerivedAddress_SettlementInstructionInsert	This datamap inserts new addresses in the Derived Address table. It derives the addresses from the INSTRUCTION table.
10340	DerivedAddress_SettlementInstructionUpd	This datamap derives the addresses from the INSTRUCTION table. It updates addresses in the Derived Address table, if already existing.
10350	SettlementInstruction_PhysicalDlvryAddrUpd	This datamap updates Mantas Physical Delivery Address Identifier in INSTRUCTION table.
10360	DerivedAddress_FrontOfficeTransactionPartyStageInsert	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_FrontOfficeTransactionPartyStageUpd	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_DerivedAddresses	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.
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.

**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
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.
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.



Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
10550	DerivedEntitytoInternalAccount_LinkStage	This datamap writes link-stage associations to various link tables in External Entity Account Table.
10560	DerivedAddressstoInternalAccount_LinkStage	This datamap writes link-stage associations to various link tables in Derived Account Address Table.
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.
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.



**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
10640	WatchListStagingTable2_CBListMembershi pUpd	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_CustListMembers hipUpd	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_EEListMembershi pUpd	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.
10670	WatchListStagingTable2_EEListMembershi pStatusUpd	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_DAListMembershi pUpd	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_DAListMembershi pStatusUpd	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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
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_WatchListStage2IntrIdUpd	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.
10730	CorrespondentBank_WatchListStage2EffectiveRisk	This datamap calculates the Client Bank Effective Risk and applies the Effective Risk and the List Risk to the CLIENT_BANK record.
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.

**Table 98. BD Datamaps (Continued)**

<b>Datamap Number</b>	<b>Datamap Name</b>	<b>Description</b>
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.
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.
	Account_EffectiveRiskFactorTxtUpd	
10880	Account_OverallEffectiveRiskUpd	This datamap updates the risk on the ACCT based on KYC, Primary customer, as well as other external risks.
	Account_AccountCustRiskUpd	
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_WatchListStage2IntrId	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
	FrontOfficeTransactionPartyRiskStage_ActivityRisk	
	FrontOfficeTransactionPartyRiskStage_ActivityRisk	
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 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
40100	InsuranceTransaction_EntityAcctivityRiskUpd	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_AcctJurisdictionReUpd	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_InvestmentObjectiveUpd	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_BackOfficeTransaction	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_FrontOfficeTransaction	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.
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.

**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
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.
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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
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.
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_TrustedFlagsUpd	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 98. BD 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.
	ExternalEntityDailyProfile	
	ExternalEntityProfile	
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.
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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
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.
	AccountDailyProfile-Trade	
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.
	InvestmentAdvisorProfile	
	RegisteredRepresentativeProfile	
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.
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.



**Table 98. BD Datamaps (Continued)**

Datamap Number	Datamap Name	Description
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 the 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).
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	CorrespondentBankPeerGroupTransaction Profile	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.

Table 98. BD Datamaps (Continued)

Datamap Number	Datamap Name	Description
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.
	AccountPosition_PercentofPortfolioUpd	
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.
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.
	CustomerIdentificationDocument	

This appendix provides instructions on how to configure the Oracle Administration Tools feature.

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.

For more information about the Administration Tools functionality, see *Oracle Financial Services Administration Tools User Guide*.



This appendix contains details of each of the pre-configured watch lists that can be used by TBAML Transaction Filtering and contains the following topics:

- [HM Treasury Reference Data](#)
- [OFAC Reference Data](#)
- [EU Reference Data](#)
- [UN Reference Data](#)
- [World-Check](#)
- [Dow Jones Watchlist](#)
- [Dow Jones Anti-Corruption List](#)
- [Accuity Reference Data](#)
- [PLI Reference Data](#)

## **HM Treasury Reference Data**

The HM Treasury publishes a sanctions list that can be used for screening in TBAML. The sanctions list provides a consolidated list of targets listed by the United Nations, European Union and United Kingdom under legislation relating to current financial sanctions regimes.

The HM Treasury website provides more details about the list at the following location:

<https://www.gov.uk/government/publications/financial-sanctions-consolidated-list-of-targets>

Transaction Filtering uses the list in a semi-colon delimited form. It can be downloaded from the following location:

<http://hmt-sanctions.s3.amazonaws.com/sanctionsconlist.csv>

## **OFAC Reference Data**

The US Treasury website states that *The US Treasury's Office of Foreign Assets Control (OFAC) administers and enforces economic and trade sanctions based on US foreign policy and national security goals against targeted foreign countries, terrorists, international narcotics traffickers, and those engaged in activities related to the proliferation of weapons of mass destruction.* More details on the OFAC list can be found on the US Treasury website available at the following location:

<http://www.treasury.gov/ofac/>

Oracle Transaction Filtering supports two lists that are produced by OFAC. The OFAC Specially Designated Nationals (SDN) list, which is available for download in three separate parts from the following links:

<https://www.treasury.gov/ofac/downloads/sdn.csv>

<https://www.treasury.gov/ofac/downloads/add.csv>

<https://www.treasury.gov/ofac/downloads/alt.csv>

The OFAC Consolidated Sanctions List, which can be downloaded in three separate parts from the following links:

[https://www.treasury.gov/ofac/downloads/consolidated/cons\\_prim.csv](https://www.treasury.gov/ofac/downloads/consolidated/cons_prim.csv)

[https://www.treasury.gov/ofac/downloads/consolidated/cons\\_add.csv](https://www.treasury.gov/ofac/downloads/consolidated/cons_add.csv)

[https://www.treasury.gov/ofac/downloads/consolidated/cons\\_alt.csv](https://www.treasury.gov/ofac/downloads/consolidated/cons_alt.csv)

## **EU Reference Data**

The European Union applies sanctions or restrictive measures in pursuit of the specific objectives of the Common Foreign and Security Policy (CFSP) as set out in Article 11 of the Treaty on European Union.

The European Commission offers a consolidated list containing the names and identification details of all persons, groups and entities targeted by these financial restrictions. See the European Commission website for more details:

[http://ec.europa.eu/cfsp/sanctions/index\\_en.htm](http://ec.europa.eu/cfsp/sanctions/index_en.htm)

The consolidated list can be downloaded from the following link:

[http://ec.europa.eu/external\\_relations/cfsp/sanctions/list/version4/global/global.xml](http://ec.europa.eu/external_relations/cfsp/sanctions/list/version4/global/global.xml)

## **UN Reference Data**

The United Nations consolidated list includes all individuals and entities subject to sanctions measures imposed by the Security Council.

Details are here:

<https://www.un.org/sc/suborg/en/sanctions/un-sc-consolidated-list>

Download link is:

<https://www.un.org/sc/suborg/sites/www.un.org.sc.suborg/files/consolidated.xml>

## **World-Check**

World-Check provides a subscription based service, offering a consolidated list of PEPs (Politically Exposed Persons) and entities and individuals appearing on the HM Treasury, OFAC, and other world lists. Three levels of subscription are provided: Standard, Premium and Premium+. Some features of the World-Check lists are only available to users with a higher subscription level.

To download the World-Check Premium+ feed, set values in the **WC Setup** section of the `watchlist-management.properties` Run Profile as follows:

```
phase.WC\ -\ Download.enabled = Y
phase.WC\ -\ Download\ native\ aliases.enabled = Y
phase.WC\ -\ Stage\ reference\ lists.enabled = Y
phase.*.snapshot.*.use_native_aliases = 1
```

To download the Standard or Premium feeds, set values in the **WC Setup** section of the `watchlist-management.properties` Run Profile as follows:

```
phase.WC\ -\ Download.enabled = Y
phase.WC\ -\ Download\ native\ aliases.enabled = N
phase.WC\ -\ Stage\ reference\ lists.enabled = Y
phase.*.snapshot.*.use_native_aliases = 0
```

See the World-Check website for more details:

<https://risk.thomsonreuters.com/en/products/third-party-risk/world-check-know-your-customer.html>

**Note:** If your instance of Transaction Filtering uses the WebLogic application server, and you are screening against the World-Check watch list, then, in order to download the World-Check reference data successfully, you must add the following to the 'Server Start' arguments of your EDQ managed server:

`-DUseSunHttpHandler=true`. This is only required if you are using the WebLogic application server and screening against the World-Check watch list.

## ***Dow Jones Watchlist***

Dow Jones provide a subscription based service offering a consolidated list of PEPs (Politically Exposed Persons) and entities and individuals appearing on the various sanctions lists. See the Dow Jones website for more details:

<http://www.dowjones.com/products/risk-compliance/>

The Dow Jones Watchlist automated download task uses one of two script files that are provided with Oracle Transaction Filtering to provide further configuration of the download process. These script files are:

- `download-djw.sh` (for use on Unix platforms)
- `download-djw.bat` (for use on Windows platforms)

The script files are invoked by the automated task and will download the data files and copy them to the appropriate sub-folder of the OEDQ landing area.

## ***Dow Jones Anti-Corruption List***

Dow Jones provides a subscription based service containing data to help you assess, investigate and monitor third-party risk with regard to anti-corruption compliance regulation. See the Dow Jones website for more details:

<http://www.dowjones.com/products/risk-compliance/>

The Dow Jones Anti-Corruption List automated download task uses one of two script files that are provided with Oracle Transaction Filtering to provide further configuration of the download process. These script files are:

- `download-djac.sh` (for use on Unix platforms)
- `download-djac.bat` (for use on Windows platforms)

The script files are invoked by the automated task and will download the data files and copy them to the appropriate sub-folder of the OEDQ landing area.

## Accuity Reference Data

The Accuity Global Watchlist is a subscription based service. The Accuity website states:

Accuity's proprietary collection of watch list screening databases is an aggregation of specially designated individuals and entities compiled from dozens of regulatory and enhanced due diligence lists from around the world. Global WatchList provides the ideal framework for your customer screening and interdiction filtering processes.

Accuity provides their aggregated data as a set of three lists, as follows:

- The Regulatory Due Diligence (RDD) Lists, covering sanctioned entities and individuals. The Accuity Group File can also be used in conjunction with this list. For more information, see *Using the Accuity Group File*.
- Enhanced Due Diligence (EDD) Lists, covering entities and individuals who are not part of the regulatory sanctions lists, but whose activities may need to be monitored.
- The Politically Exposed Persons (PEPs) Due Diligence Database, and covering PEPs.

Any or all of the lists can be downloaded and used separately or in conjunction with each other.

For more information, see <http://www.accuity.com/compliance/>.

### Using the Accuity Group File

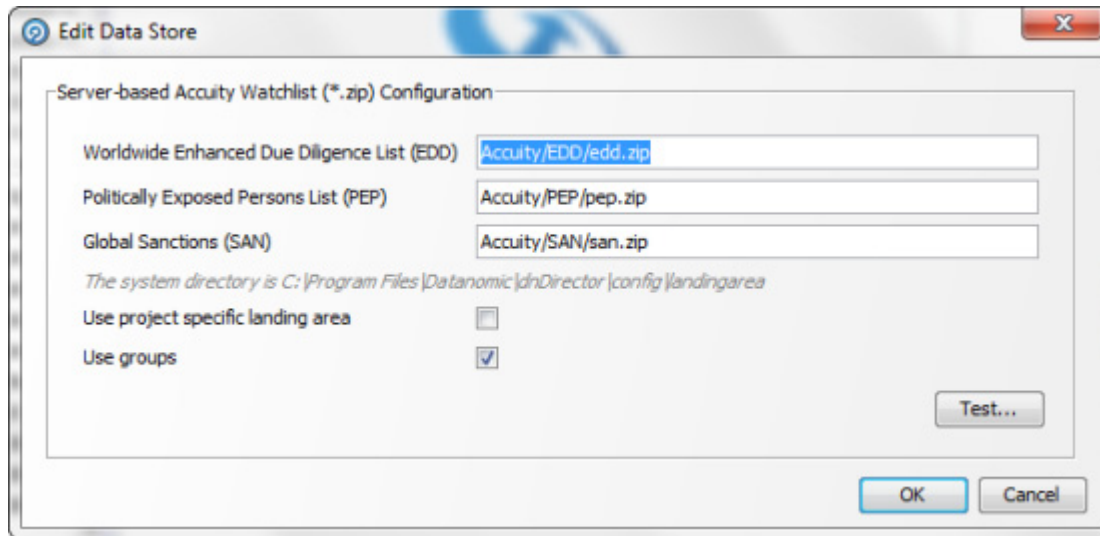
The Accuity Global Watchlist is created by aggregating many other lists. As such, any given individual or entity may be represented in the list by multiple entries.

The group file, **GROUP.XML**, provides a way to work with a data set of this type in Transaction Filtering. All records which represent the same individual or entity are collected into groups, and each group is assigned a unique group ID. The group ID is used with a prefix to indicate the fact that this is a group ID, in place of the original record identifier in Case Management. Records which are not included in a group use their original Accuity record ID, with a different prefix to indicate that they are single records.

**Note:** The group file only applies to Transaction Filtering. That is, only entities and individuals on the Regulatory Due Diligence (RDD) Lists are included in the group file.



The group file allows case generation to be centered around real-world individuals, rather than separate watch list records. Groups are used by default. To change this, open the Accuity Data Store in the Watchlist Management project, and deselect the **Use groups** option:



**Figure 64. Edit Data Store**

If you choose to use the group file but it is not present in your downloaded data, an error will be generated.

## New Alerts Resulting from Use of the Group File

Using the group file causes the original list ID for an entry to be replaced with the appropriate group ID. The list ID is used in the event key, so changes to the list ID will result in new events being raised for existing, known relationships. There are two main scenarios in which this may occur:

Individuals or entities are moved into, out of or between groups by Accuity, new events will be generated for existing relationships.

**Note:** Use of the group file may result in new events being raised for existing relationships if the group file structure is changed by Accuity. There is at present no way to circumvent this issue

The **Use Groups** setting is changed after cases and events have already been generated.

**Note:** The setting for the **Use Groups** option should be selected during the implementation phase of the project. Once screening has started, it should not be changed unless absolutely necessary. Changing this setting is likely to result in duplication of existing events with a new event ID.

## PLI Reference Data

This section describes the structure of the .csv files used in the Private List Interface (PLI).

Private watch list data are provided in two .csv(comma seperated value) files; `privateindividuals.csv` and `privateentities.csv`. These files come with a pre-defined structure and set of validation rules. On installation, these files are populated with sample private watch list data, which must be replaced with your own data, once it has been transformed into the required format. For information on the location of the .csv files, see *Installation Guide*.

**Note:**

- It is recommended that you keep a copy of the sample private watch list files, as they can be used to verify the correct functioning of your installation on a known data set.
- The files must be saved in UTF-8 format.

## PLI Attributes

Three types of attributes are used in the PLI for screening:

- **Mandatory attributes:** These attributes are tagged in the PLI tables with the [Mandatory attribute] tag, and are mandatory for screening.
- **Recommended attributes:** These attributes are used in matching, typically to either eliminate false positive matches which may occur if the mandatory fields alone were used or to reinforce the likelihood of a possible match. They are tagged in the PLI tables with the [Recommended attribute] tag.
- **Optional attributes:** These attributes are not used in matching. Information provided in these fields may be of use in processes downstream of the match process.

This section covers the following areas:

- [Individual Private Watch List Input Attributes](#)
- [Entity Private Watch List Input Attributes](#)

### Individual Private Watch List Input Attributes

This section lists the PLI fields used for individuals. In addition to the prescribed fields, fifty customizable input attributes are available for individual private watch lists, out of which forty are string attributes, five are date attributes and five are number attributes. They are available for any additional inputs required by your private watch list.

The following table lists the individual PLI fields in order, the data format expected for each field, and notes on their use in screening:

**Table 99. Private List for Individuals**

Field Name	Expected Data Format	Notes
ListSubKey	String	This field is used to identify the source list of the watch list record (for example, Private List, Accounting Private List, Financial Private List and so on). It is included in the event key.
ListRecordType	String	
ListRecordOrigin	String	This field is used to record the provenance of a record when it is part of a consolidated list.
ListRecordId	String	[Mandatory attribute] This attribute is not used as part of the matching process, but it must be populated with a unique identifier.
PassportNumber	String	This is an optional field that may be used to capture the passport numbers of customers or individuals for use in the review process. <b>Note:</b> Passport numbers are not used in the default screening rules.

**Table 99. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
NationalId	String	This is an optional field that may be used to capture customer National IDs where known for use in the review process. <b>Note:</b> The National IDs of customers and individuals must not used in the default screening rules.
Title	String	This field should contain the titles of customers or individuals (such as Mr/Mrs/Dr/Herr/Monsieur). It is used to derive gender values where the gender is not already stated, and is used during the review process. <b>Note:</b> Avoid putting titles in the name fields.
FullName	String	[Mandatory attribute] The individual matching process is based primarily on the name supplied for the individual. Either a full name, a pair of given and family names, or an original script name must be submitted to the screening process for screening to proceed.
GivenName	String	
FamilyName	String	
NameType	String	This is an optional field used in the review process only. Multiple names may exist for the same person. The Name Type therefore denotes if the name is the primary name of the listed party, or an additional name (such as an Alias, or Alternate Spelling). If two private list records were derived from a single source with multiple names (such as Mrs Louise Wilson née Hammond being split into two records, Louise Wilson and Louise Hammond) you may wish to denote one as the primary name and one as a maiden or alias name.
NameQuality	String	This field may be assigned a value of Low, Medium or High to indicate the quality of the individual name. High is used for Primary names and specified good/high quality aliases.
PrimaryName	String	For alias records, this field indicates the main name for that record.
OriginalScriptName	String	[Mandatory attribute] The individual matching process is based primarily on the name supplied for the individual. Either a full name, a pair of given and family names, or an original script name must be submitted to the screening process for screening to proceed. If you populate the OriginalScriptName, then you will also need to enable two facets of Match processor configuration that are disabled by default: the Original Script Name Cluster and some or all of the Match Rules that include Original script name in their name. To adapt Match Processor configuration, you will need to open the Transaction screening project within the Director user interface, and make the changes to every process used during the Transaction Filtering installation.
Gender	String	The value supplied should be either 'M' or 'F'. The gender is not used directly in the matching process, but optionally, the value of the Gender field can be used by the elimination rules to eliminate poor matches.
Occupation	String	This is an optional field that may be used to eliminate records with "safe" occupations, in the review process and in risk scoring. Note that customer occupations are not matched against list occupations using the default screening rules.

**Table 99. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
DateofBirth	String, representing a date, in the format 'YYYYMMDD'; day, month and year are required.	[Recommended attribute] Birth date information can be used in matching to identify particularly strong matches, or to eliminate matches that are too weak.
YearofBirth	String, in the format 'YYYY'.	
Deceased Flag	String	If populated, this optional field should contain either Y or N.
DeceasedDate	String, representing a date, in the format 'YYYYMMDD'.	If populated, this optional field should contain either the current date or a date in the past.
Address1	String	These are optional fields that may be used in the review process.
Address2	String	
Address3	String	
Address4	String	
City	String	[Recommended attribute] City data is used to strengthen potential match information.
State	String	
Postal Code	String	
AddressCountryCode	String; ISO 2-character country code.	[Recommended attribute] Address country data is used to strengthen potential match information.
ResidencyCountryCode	String; ISO 2-character country code.	[Recommended attribute] The country of residence can be used in optional country prohibition screening.
CountryOfBirthCode	String; ISO 2-character country code.	[Recommended attribute]
NationalityCountryCodes	String; commaseparated list of ISO 2-character country codes.	[Recommended attribute] The nationality can be used in optional country prohibition screening.
ProfileHyperlink	String; a hyperlink to an Internet or intranet resource for the record.	This field may contain a hyperlink to an Internet or intranet resource that can provide reviewers with additional information about the individual.
RiskScore	Number, between 0 and 100	This field is included where the risk score for a customer is calculated externally.
RiskScorePEP	Number, between 0 and 100	A number indicating the relative 'riskiness' of the individual, considered as a PEP. The risk score is expressed as an integer between 1 and 100, with higher numbers indicating a higher risk.

**Table 99. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
AddedDate	String, representing a date, in the format 'YYYYMMDD'	These are optional fields for use in the review process.
LastUpdatedDate	String, representing a date, in the format 'YYYYMMDD'	
DataConfidenceScore	Number, between 0 and 100	
DataConfidenceComment	String	
InactiveFlag	String	If populated, this optional field should contain either Y or N.
InactiveSinceDate	String, representing a date, in the format 'YYYYMMDD'	If populated, this optional field should contain either the current date or a date in the past.
PEPclassification	String	This field can be used to indicate the type of PEP (for example, whether the individual is part of an international organization or government, and at what level). It can be used to filter watch list records, and is primarily used by the World-Check watch list, but could be used by a private watch list if required.
customString1 to customString40	String	Fifty custom fields are provided in the private list data interface for individuals. Forty of these are intended to hold string data, five hold dates and five numeric data. <b>Note:</b> The interface file is a comma-separated value (.csv) file, and so all fields intrinsically contain strings. However, during the processing of Private watch lists, the custom date and number fields are checked to ensure that they include appropriate data, and warning messages are provided as output if they do not.
customDate1 to customDate5	String, representing a date, in the format 'YYYYMMDD'	
customNumber1 to customNumber5	Number	

## Entity Private Watch List Input Attributes

This section lists the PLI fields used for entities. In addition to the prescribed fields, fifty customizable input attributes are available for individual private watch lists, out of which forty are string attributes, five are date attributes and five are number attributes. They are available for any additional inputs required by your private watch list.

The following table lists the individual PLI fields in order, the data format expected for each field, and notes on their use in screening:

**Table 100. Private List for Individuals**

Field Name	Expected Data Format	Notes
ListSubKey	String	This field is used to identify the source list of the watch list record (for example, Private List, Accounting Private List, Financial Private List and so on). It is included in the event key.
ListRecordType	String	[Mandatory attribute] This field is used when filtering events, to determine whether the record is a sanctions or PEP record. It must contain a value of SAN, PEP, or a combination of these values. If you want to include a combination of values, the values should be comma-separated, and enclosed by double quotation marks. For example: "SAN, PEP".

**Table 100. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
ListRecordOrigin	String	This field is used to record the provenance of a record when it is part of a consolidated list.
ListRecordId	String	[Mandatory attribute] This attribute is not used as part of the matching process, but it must be populated with a unique identifier.
RegistrationNumber	String	This is an optional field that may be used to capture entity registration numbers where known for use in the review process. Note that entity registration numbers are not used for matching in the default screening rules.
EntityName	String	[Mandatory attribute] The entity matching process is based primarily on the name supplied for the entity. An entity name or original script name must be submitted to the screening process for screening to proceed.
NameType	String	This is an optional field used in the review process only. Multiple names may exist for the same person. The Name Type therefore denotes if the name is the primary name of the listed party, or an additional name (such as an Alias, or Alternate Spelling). If two private list records were derived from a single source with multiple names (such as Mrs Louise Wilson née Hammond being split into two records, Louise Wilson and Louise Hammond) you may wish to denote one as the primary name and one as a maiden or alias name.
NameQuality	String	This field may be assigned a value of Low, Medium or High to indicate the quality of the individual name. High is used for Primary names and specified good/high quality aliases.
PrimaryName	String	For alias records, this field indicates the main name for that record.
OriginalScriptName	String	[Mandatory attribute] The individual matching process is based primarily on the name supplied for the individual. Either a full name, a pair of given and family names, or an original script name must be submitted to the screening process for screening to proceed. If you populate the OriginalScriptName, then you will also need to enable two facets of Match processor configuration that are disabled by default: the Original Script Name Cluster and some or all of the Match Rules that include Original script name in their name. To adapt Match Processor configuration, you will need to open the Transaction screening project within the Director user interface, and make the changes to every process used during the Transaction Filtering installation.
AliasIsAcronym	String	If this field is set to Y, this flags an alias as an acronym as opposed to a full entity name. Leaving the field blank or setting it to any other value has no effect (that is, an alias is assumed to be a full entity name). <b>Note:</b> This flag is used during matching.
VesselIndicator	String	This field should be set to Y if the entity is a vessel (a ship). It should be left empty or set to N if the entity is not a vessel.
VesselInfo	String	If the entity is a vessel, you can populate this field with information about it: for example, its call sign, type, tonnage, owner, flag and so on.

**Table 100. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
Address1	String	These are optional fields that may be used in the review process.
Address2	String	
Address3	String	
Address4	String	
City	String	[Recommended attribute] City data is used to strengthen potential match information.
State	String	
Postal Code	String	
AddressCountryCode	String; ISO 2-character country code.	[Recommended attribute] Address country data is used to strengthen potential match information.
ResidencyCountryCode	String; ISO 2-character country code.	[Recommended attribute] The entity's registration country can be used in optional country prohibition screening.
OperatingCountryCodes	String; ISO 2-character country code.	[Recommended attribute] Any of the entity's operating countries can be used in optional country prohibition screening.
ProfileHyperlink	String; a hyperlink to an Internet or intranet resource for the record.	This field may contain a hyperlink to an Internet or intranet resource that can provide reviewers with additional information about the individual.
RiskScore	Number, between 0 and 100	This field is included where the risk score for a customer is calculated externally.
RiskScorePEP	Number, between 0 and 100	A number indicating the relative 'riskiness' of the individual, considered as a PEP. The risk score is expressed as an integer between 1 and 100, with higher numbers indicating a higher risk.
AddedDate	String, representing a date, in the format 'YYYYMMDD'	These are optional fields for use in the review process.
LastUpdatedDate	String, representing a date, in the format 'YYYYMMDD'	
DataConfidenceScore	Number, between 0 and 100	
DataConfidenceComment	String	
InactiveFlag	String	If populated, this optional field should contain either Y or N.
InactiveSinceDate	String, representing a date, in the format 'YYYYMMDD'	If populated, this optional field should contain either the current date or a date in the past.
PEPclassification	String	This field can be used to indicate the type of PEP (for example, whether the individual is part of an international organization or government, and at what level). It can be used to filter watch list records, and is primarily used by the World-Check watch list, but could be used by a private watch list if required.

**Table 100. Private List for Individuals (Continued)**

Field Name	Expected Data Format	Notes
customString1 to customString40	String	Fifty custom fields are provided in the private list data interface for individuals. Forty of these are intended to hold string data, five hold dates and five numeric data. <b>Note:</b> The interface file is a comma-separated value (.csv) file, and so all fields intrinsically contain strings. However, during the processing of Private watch lists, the custom date and number fields are checked to ensure that they include appropriate data, and warning messages are provided as output if they do not.
customDate1 to customDate5	String, representing a date, in the format 'YYYYMMDD'	
customNumber1 to customNumber5	Number	



## *Match Score Rules*

See *Oracle Financial Services Transaction Filtering Matching Guide* for information on Match Score Rules.





