

OFS Performance Analytics Cloud Service

Operations Guide

Release 9.0.0.0.0

Feb 2020

ORACLE
Financial Services

OFS Performance Analytics Cloud Service

Copyright © 2021 Oracle and/or its affiliates. All rights reserved.

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are “commercial computer software” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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

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

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

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

Document Control

Version Number	Revision Date	Change Log
9.0	28 January 2020	First release

Table of Contents

1	Preface	6
1.1	Intended Audience	6
1.2	Documentation Accessibility	6
2	OFS Performance Analytics Cloud Service Run Chart	7
3	Dimension Loading Process	8
3.1	Dimension Tables Population	8
3.2	Overview of SCD Process	8
3.2.1	<i>Prerequisites</i>	8
3.2.2	<i>Tables Used by the SCD Component</i>	9
3.2.3	<i>Executing the SCD Component</i>	11
3.2.4	<i>Checking the Execution Status</i>	13
4	Time Dimension Population	14
4.1	Overview of Time Dimension Population	14
4.1.1	<i>Prerequisites</i>	14
4.1.2	<i>Tables used by the Time Dimension Population Transformation</i>	15
4.1.3	<i>Executing the Time Dimension Population Transformation</i>	15
4.1.4	<i>Checking the Execution Status</i>	16
5	Account Dimension Population	17
5.1	Dimension Tables Population	17
5.2	Overview of SCD process	17
5.2.1	<i>Prerequisites</i>	18
5.3	Tables Used by the SCD Component	18
5.3.1	<i>Executing the SCD Component</i>	21
5.3.2	<i>Checking the Execution Status</i>	22
5.3.3	<i>Load DIM_ACCOUNT through SCD</i>	23
5.3.4	<i>DIM_ACCOUNT SCD</i>	23
5.3.5	<i>Load DIM Tables through SCD</i>	24
5.3.6	<i>Improve SCD Performance</i>	24

5.3.7	Handling Multiple GAAP Codes	25
6	Exchange Rate History Population.....	26
6.1	Introduction	26
6.2	Execution of Currency Exchange Rates Population T2T	26
6.3	Currency Execution Rates - Batch Execution	26
6.4	Exchange Rate History Population	27
6.4.1	Checking the Execution Status	28
6.4.2	Validating the Exchange Rate.....	29
7	Developing a New Cube.....	30
8	List of Hard-Coded Members.....	31
9	Run Rule Framework	32
9.1	Executing a Seeded Run.....	32
9.2	Runs Available for OFS Performance Analytics Cloud Service	34
10	PA Metrics Computation Logic	35
10.1	Projection Logic.....	35
10.2	Data Treatment for Months.....	35
10.3	Aggregation and Projection Logic for Forecasting of Replines under PA	37

1 Preface

1.1 Intended Audience

Welcome to Release 9.0.0.0.0 of the Oracle Financial Services Performance Analytics Cloud Service Operations Guide.

This guide is intended for the users of Oracle Financial Services Performance Analytics Cloud Service. See Related Information Sources for more Oracle product information.

1.2 Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at:

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit:

- <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info>
- <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs>, if you are hearing impaired.

2 **OFS Performance Analytics Cloud Service Run Chart**

Run chart, as the name suggests helps the consultant understand the sequence of tasks to be performed to ensure the correctness of data flow in the product. It contains the details about Data loading for all mandatory tables for the product, and the sequence of execution of seeded tasks/batches.

Oracle Financial Services Performance Analytics (OFS PA) provides the Run Chart listing the tasks required for population of data for OFS PA Reports. This covers the following tasks:

- Stage Dimension Load
- Common data Population (for example Exchange Rate)
- Set up table population
 - Dimensions
 - Facts
- Profitability tables Population
- Model Execution
- Derived Entity Refresh
- Cubes Build

Download the OFS PA Cloud Service 9.0.0.0.0 Run Chart from the [MOS page](#).

3 Dimension Loading Process

This chapter covers the following sections:

- [Dimension Tables Population](#)
- [Overview of SCD Process](#)

3.1 Dimension Tables Population

OFS Performance Analytics Cloud Service uses the SCD component to handle dimensional data changes.

3.2 Overview of SCD Process

SCDs are dimensions that have data that changes slowly, rather than changing on a time-based, regular schedule.

For more information on SCDs, see

- Oracle Data Integrator Best Practices for a Data Warehouse at <https://www.oracle.com/technetwork/middleware/data-integrator/learnmore/odi-best-practice-data-warehouse-168255.pdf>
- Oracle® Warehouse Builder Data Modeling, ETL, and Data Quality Guide at https://docs.oracle.com/cd/E11882_01/owb.112/e10935.pdf

Additional online sources include:

- http://en.wikipedia.org/wiki/Slowly_changing_dimension
- http://www.oracle.com/webfolder/technetwork/tutorials/obe/db/10g/r2/owb/owb10gr2_gs/owb/lesson3/slowlychangingdimensions.htm
- <http://www.oraclebidwh.com/2008/11/slowly-changing-dimension-scd/>
- <http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=204800027&pgno=1>
- <http://www.informationweek.com/news/software/bi/showArticle.jhtml?articleID=59301280>
- An excellent published resource that covers SCD in detail is *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling* by Ralph Kimball and Margy Ross.

The SCD component of the platform is delivered via a C++ executable. The types of SCD handled by the OFSAAI SCD component for OFSPA solution are Type 1 and Type 2.

3.2.1 Prerequisites

- The SCD executable should be present under <installation home>ficdb/bin. The file name is scd.
- The user executing the SCD component should have execute rights on the file mentioned as prerequisite in point 2.

- The setup tables accessed by SCD component are SYS_TBL_MASTER and SYS_STG_JOIN_MASTER. SYS_TBL_MASTER stores the information like which is the source stage table and the target dimension tables. The source sometimes can be the database views which could be simple or a complex view. SYS_STG_JOIN_MASTER stores the information like which source column is mapped to which column of a target dimension table. It makes use of data base sequence to populate into surrogate key columns of dimension tables.

3.2.2 Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER

The solution installer will populate one row per dimension for the seeded dimensions in this table.

SYS_TBL_MASTER Dimensions

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
TBL_NM	VARCHAR2(30) NOT NULL	Dimension Table Name
STG_TBL_NM	VARCHAR2(30) NOT NULL	Staging Table Name
SRC_PRTY	NUMBER(2) NULL	Priority of the Source when multiple sources are mapped to the same target.
SRC_PROC_SEQ	NUMBER(2) NOT NULL	The sequence in which the various sources for the DIMENSION will be taken up for processing.
SRC_TYP	VARCHAR2(30) NULL	The type of the Source for a Dimension, that is, Transaction Or Master Source.
DT_OFFSET	NUMBER(2) NULL	The offset for calculating the Start Date based on the Functional Requirements Document (FRD).
SRC_KEY	NUMBER(3) NULL	

Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

MAP_REF_NUM	6
TBL_NM	DIM_LOB
STG_TBL_NM	STG_LOB_MASTER
SRC_PRTY	
SRC_PROC_SEQ	23
SRC_TYP	MASTER

DT_OFFSET	0
SRC_KEY	

NOTE For any new dimension added, a row will have to be inserted to this table manually.

- **SYS_STG_JOIN_MASTER**

The solution installer will populate this table for the seeded dimensions.

SYS_STG_JOIN_MASTER Dimensions

Column Name	Data Type	Column Description
MAP_REF_NUM	NUMBER(3) NOT NULL	The Mapping Reference Number for this unique mapping of a Source to a Dimension Table.
COL_NM	VARCHAR2(30) NOT NULL	Name of the column in the Dimension Table.
COL_TYP	VARCHAR2(30) NOT NULL	Type of column. The possible values are given in the following section.
STG_COL_NM	VARCHAR2(60) NULL	Name of the column in the Staging Table.
SCD_TYP_ID	NUMBER(3) NULL	SCD type for the column.
PRTY_LOOKUP_REQD_FLG	CHAR(1) NULL	Column to determine whether Lookup is required for Priority of Source against the Source Key Column or not.
COL_DATATYPE	VARCHAR2(15) NULL	The list of possible values are VARCHAR, DATE, NUMBER based on the underlying column datatype.
COL_FORMAT	VARCHAR2(15) NULL	

The possible values for column type (the COL_TYPE column) in SYS_STG_JOIN_MASTER are:

- PK – Primary Dimension Value (may be multiple for a given "Mapping Reference Number")
- SK – Surrogate Key
- DA – Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
- SD – Start Date
- ED – End Date
- LRI – Latest Record Indicator (Current Flag)
- CSK – Current Surrogate Key

1. From the **Home** menu, select **Operations**, then select **Batch Maintenance**.
2. Click **New Batch** ('+' symbol in Batch Name container) and enter the Batch Name and Description.
3. Click **Save**.
4. Select the Batch you created in the earlier step by clicking the check box in the Batch Name container.
5. Click **New Task** ('+' symbol in Task Details container).
6. Enter the **Task ID** and **Description**.
7. Select **Run Executable**, from the Component ID list.
8. Click **Parameters**. Select the following from the Dynamic Parameters List and then click **Save**:
 - Datastore Type: Select the appropriate datastore from the list
 - Datastore Name: Select the appropriate name from the list
 - IP address: Select the IP address from the list
 - Executable: scd,<map ref num>
Example
scd, 61 (Refer the following sections for details)
 - Wait: When the file is being executed you have the choice to either wait till the execution is complete or proceed with the next task. Click the list box of the field provided for Wait in the Value field to select 'Yes' or 'No'. Clicking Yes that you wish to wait for the execution to be complete. Clicking No indicates that you wish to proceed.
 - Batch Parameter: Clicking Yes would mean that the batch parameters are also passed to the executable being started; else the batch parameters will not be passed to the executable.

ATTENTION Always select **Y** in Batch Parameter.

For the Parameter Executable earlier mentioned, the map ref num values are:

- -1 (if you want to process all the dimensions). The Executable parameter mentioned earlier would be scd,-1
- If you want to process for a single dimension, query the database table SYS_TBL_MASTER and give the number in the map_ref_num column for the dimension you want to process. These are the ones which come seeded with the install.
- Execute the batch from Batch Execution by choosing the batch created following the steps mentioned in the preceding sections for a date.

NOTE A seeded batch <Infodom>_SCD_Institutional_Perf_Dim is provided which has all the required dimensions as different tasks that are part of SCD.

3.2.4 Checking the Execution Status

The status of execution can be monitored using the Batch Monitor screen. You can access this from the Left Hand Side (LHS) menu as follows:

From the **Home** menu, select **Operations**, then select **Batch Monitor**.

For a more comprehensive coverage, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

The status messages in Batch Monitor are :

- N: Not Started
- O: On Going
- F: Failure
- S: Success

The ICC execution log can be accessed on the application server in the following directory:
`$FIC_DB_HOME/log/ficgen`.

The file name will have the batch execution id.

Sample

```
/dbfiles/home/oracle/OFSAAI/ficdb/log/ficgen
```

The detailed SCD component log can be accessed on the application server in the directory `$FIC_HOME`, go one folder up from there and then accessing the following path
`/ftpshare/<infodom name>/logs`

The file name will have the batch execution id.

Sample

```
/dbfiles/home/oracle/ftpshare/OFSAADEMO/logs
```

Check the `.profile` file in the installation home if you are not able to find the paths mentioned earlier.

4 Time Dimension Population

Business data commonly represents information as of a point in time (for example, a balance as of a point in time) or as of a particular span of time (for example, income for the month of March). Time dimension makes it possible to report the balances by Year, Quarter or Month using the rollup functionality of cubes. Cubes makes it possible to rollup the monthly balances to a quarter and then to a year level. For example, the monthly data for January, February and March gets rolled up to Quarter 1 and the Quarter 1, 2, 3 and 4 data get rolled up to, say Year 2011. The rollup of a particular balance depending on their nature could be a simple additive rollup wherein the child member balances are added up to arrive at the parent node balance (for example, Ending Balance) or non additive rollups wherein a node formula is used to specify how to rollup the child member balances (for example, 3 month rolling average).

Point in time reporting is supported for all the reports. The report is represented as of the data selected in the dashboard time prompts. By default, reports is always displayed for the latest available data.

This chapter discusses the following topics:

- [Overview of Time Dimension Population](#)
- [Tables used by the Time Dimension Population Transformation](#)

4.1 Overview of Time Dimension Population

Time dimension population transformation is used to populate the DIM_DATES table with values between two dates specified by the user as a batch parameter.

The database components, used by the transformations are:

1. Database function FN_DIM_DATES
2. Database procedure PROC_DIM_DATES_POPULATION, that is called by the database function FN_DIM_DATES.

4.1.1 Prerequisites

1. All the post install steps mentioned in the Oracle Financial Services Analytical Applications Infrastructure (OFSAAI) Installation and Configuration guide and the solution installation manual of Oracle Financial Services Institutional Performance Analytics have to be completed successfully.
2. Application User must be mapped to a role that has seeded batch_execution function (BATPRO).
3. Before executing a batch check if the following services are running on the application server (For more information on how to check if the services are up and on and how to start the services if you find them not running, see Oracle Financial Services Analytical Applications Infrastructure User Guide).
 - lccserver
 - Router
 - AM Server
 - Messageserver

4. Batches will have to be created for executing the function. For more details see, Executing the Time Dimension Population Transformation.

4.1.2 Tables used by the Time Dimension Population Transformation

DIM_DATES: This table stores the date details to be used for building the cubes.

For more details on viewing the structure of this table, refer to *Oracle Financial Services Analytical Applications Data Model Data Dictionary* or the *Erwin Data Model*.

4.1.3 Executing the Time Dimension Population Transformation

To execute the function from OFSAAI Information Command Center (ICC) frame work, create a batch by performing the following steps:

1. From the **Home** menu, select **Operations**, then select **Batch Maintenance**.
2. Click **New Batch** ('+' symbol in Batch Name container) and enter the Batch Name and description.
3. Click **Save**.
4. Select the Batch you have created in the earlier step by clicking on the check-box in the Batch Name container.
5. Click **New Task** ('+' symbol in Task Details container).
6. Enter the **Task ID** and **Description**.
7. Select **Transform Data**, from the components list.
8. Select the following from the Dynamic Parameters List and then click Save:
 - Datastore Type: Select appropriate datastore from the list
 - Datastore Name: Select appropriate name from the list
 - IP address: Select the IP address from the list
 - Rule Name: Select Dim_Dates_Population from the list of all available transformations. (This is a seeded Data Transformation which is installed as part of the OFSIPA solution installer. If you don't see this in the list, contact Oracle support)
 - Parameter List: Start Date, End Date (Refer the following for details on Parameter list)

Explanation for the parameter list is:

- Start Date: This is the date starting from which the Transformation will populate Dim_Dates table. Date should be specified in the format 'YYYYMMDD'.
 - End Date: This is the date up to which the Transformation will populate Dim_Dates table. Date should be specified in the format 'YYYYMMDD'. Sample parameter for this task is '20081131', '20091231'.
9. You can execute the batch in two ways:
 - Execute the batch from Batch Execution by choosing the batch created following the steps mentioned in the preceding sections for a date.

NOTE

A seeded batch <INFODOM>_aCRM_CommonTasks - Task2 is provided so that the user can just modify the parameters and execute the batch.

- The function can also be executed directly on the database through SQLPLUS.

Details are:

- Function Name: FN_DIM_DATES
 - Parameters: P_BATCH_RUN_ID, P_AS_OF_DATE, P_ST_DT, and P_ED_DT
- Sample parameter values: 'Batch1', '20091231', '20081131', and '20091231'

4.1.4 Checking the Execution Status

The status of execution can be monitored using the batch monitor screen.

The status messages in batch monitor are :

- N: Not Started
- O: On Going
- F: Failure
- S: Success

The Event Log window in Batch Monitor provides logs for execution with the top row being the most recent. If there is any error during execution, it will get listed here. Even if you see Successful as the status in Batch Monitor it is advisable to go through the Log and re-check if there are any errors. The execution log can be accessed on the

application server by going to the following directory \$FIC_DB_HOME/log/date. The file name will have the batch execution id. The database level operations log can be accessed by querying the FSI_MESSAGE_LOG table. The batch run id column can be filtered for identifying the relevant log.

Check the `.profile` file in the installation home if you are not able to find the paths mentioned earlier.

5 Account Dimension Population

This chapter discusses the following topics:

- [Dimension Tables Population](#)
- [Overview of SCD process](#)
- [Tables Used by the SCD Component](#)

5.1 Dimension Tables Population

Data Foundation solutions use the SCD component to handle dimensional data changes.

5.2 Overview of SCD process

A Slowly Changing Dimension (SCD) is a dimension that stores and manages both current and historical data over time in a data warehouse. SCDs are dimensions that have data that changes slowly, rather than changing on a time-based, regular schedule. It is considered and implemented as one of the most critical ETL tasks in tracking the history of dimension records. There are three types of SCDs and you can use Warehouse Builder to define, deploy, and load all three types of SCDs.

- Type 1 SCDs: Overwriting

The Type 1 methodology overwrites old data with new data, and therefore does not track historical data. This is useful for making changes to dimension data.

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	PL	5/31/2010	12/31/9999	N

In this example, N_PRODUCT_SKEY is the surrogate key column which is a unique key for each record in the dimension table. V_PRODUCT_NAME is the product name. D_START_DATE indicates the date from which this product record is valid. D_END_DATE indicates the date till which this product record is valid.

F_LATEST_RECORD_INDICATOR with value 'Y', which indicates this is the latest record in the dimension table for this product and 'N' indicates it is not. If the V_PRODUCT_NAME column is set as a Type 1 SCD column and if there is a change in the product name to 'Personal Loan' from 'PL' in the above example, in the next processing period, then when SCD is executed for the new processing period the record in the above example changes to:

Type 1 SCDs - Overwriting1

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	Personal Loan	6/30/2010	12/31/9999	Y

- Type 2 SCDs: Creating another dimension record

The Type 2 method tracks historical data by creating multiple records for a given natural key in the dimensional tables with separate surrogate keys. With Type 2, the historical changes in dimensional data are preserved. In the above example for the change in product name from 'PL' to 'Personal Loan' if history has to be preserved, then the V_PRODUCT_NAME column has to be set as Type 2 when SCD is processed for the processing period and the change inserts a new record as shown in the following example:

N_PRODUCT_SKEY	V_PRODUCT_NAME	D_START_DATE	D_END_DATE	F_LATEST_RECORD_INDICATOR
1	PL	6/30/2010	12/31/9999	N
2	Personal Loan	6/30/2010	12/31/9999	Y

A new record is inserted to the product dimension table with the new product name. The latest record indicator for this is set as 'Y', indicating that this is the latest record for the personal loan product. The same flag for the earlier record was set to 'N'.

- Type 3 SCDs - Creating a current value field

A Type 3 SCD stores two versions of values for certain selected level attributes. Each record stores the previous value and the current value of the selected attribute.

When the value of any of the selected attributes changes, the current value is stored as the old value and the new value becomes the current value.

5.2.1 Prerequisites

Following are the prerequisites:

1. The SCD executable should be present under <installation home>ficdb/bin. The file name is scd.
2. The user executing the SCD component should have execute rights on the file mentioned as prerequisite in point 2.
3. The setup tables accessed by SCD component are SYS_TBL_MASTER and SYS_STG_JOIN_MASTER. SYS_TBL_MASTER stores the information like which is the source stage table and the target dimension tables. The source sometimes can be the database views which could be simple or a complex view. SYS_STG_JOIN_MASTER stores the information like which source column is mapped to which column of a target dimension table. It makes use of data base sequence to populate into surrogate key columns of dimension tables.

5.3 Tables Used by the SCD Component

The database tables used by the SCD component are:

- SYS_TBL_MASTER

The solution installer will populate one row per dimension for the seeded dimensions in this table.

SYS_TBL_MASTER dimensions

Table Name	Column Name	Expected Values
DIM_CUSTOMER_TYPE	V_CUST_CATEGORY	C
FCT_CRM_ACCOUNT_SUMMARY	V_SCENARIO_CODE	PLAN, BUDGET
FCT_OPPORTUNITY_ACTIVITY	V_ACTIVITY_STATUS	O, C
DIM_BANDS	V_BAND_TYPE	AGEONBOOK
		TURNOVER
FCT_ACCOUNT_PROFITABILITY	N_REP_LINE_CD	98000 - Net Income Before Taxes
		98500 - Tax Expense
		99000 - Net Income After Taxes
		107100 - Number of Customers
		107130 - Number of Open Customers
		107200 - Number of Accounts
		107230 - Number of Open Accounts
		107300 - Attrition Rate

Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

NOTE For any new dimension added, a row will have to be inserted to this table manually.

- SYS_STG_JOIN_MASTER

The solution installer will populate this table for the seeded dimensions.

SYS_STG_JOIN_MASTER dimensions

Table Name	Column Name	Expected Values
DIM_CUSTOMER_TYPE	V_CUST_CATEGORY	C
FCT_CRM_ACCOUNT_SUMMARY	V_SCENARIO_CODE	PLAN, BUDGE
FCT_OPPORTUNITY_ACTIVITY	V_ACTIVITY_STATUS	O, C
DIM_BANDS	V_BAND_TYPE	AGEONBOOK
		TURNOVER
FCT_ACCOUNT_PROFITABILITY	N_REP_LINE_CD	98000 - Net Income Before Taxes
		98500 - Tax Expense
		99000 - Net Income After Taxes

Table Name	Column Name	Expected Values
		107100 - Number of Customers
		107130 - Number of Open Customers
		107200 - Number of Accounts
		107230 - Number of Open Accounts
		107300 - Attrition Rate

The possible values for column type (the COL_TYPE column) in SYS_STG_JOIN_MASTER are:

- PK – Primary Dimension Value (may be multiple for a given "Mapping Reference Number")
- SK – Surrogate Key
- DA – Dimensional Attribute (may be multiple for a given "Mapping Reference Number")
- SD – Start Date
- ED – End Date
- LRI – Latest Record Indicator (Current Flag)
- CSK – Current Surrogate Key
- CSK – Current Surrogate Key
- SS – Source Key
- LUD – Last Updated Date/Time
- LUB – Last Updated By

Sample Data: This is the row put in by the solution installer for the Line of Business dimension.

Table Name	Column Name	Expected Values
DIM_CUSTOMER_TYPE	V_CUST_CATEGORY	C
FCT_CRM_ACCOUNT_SUMMARY	V_SCENARIO_CODE	PLAN, BUDGET
FCT_OPPORTUNITY_ACTIVITY	V_ACTIVITY_STATUS	O, C
DIM_BANDS	V_BAND_TYPE	AGEONBOOK
		TURNOVER
FCT_ACCOUNT_PROFITABILITY	N_REP_LINE_CD	98000 - Net Income Before Taxes
		98500 - Tax Expense
		99000 - Net Income After Taxes
		107100 - Number of Customers
		107130 - Number of Open Customers
		107200 - Number of Accounts
		107230 - Number of Open Accounts
		107300 - Attrition Rate

Example

scd, 61 (Refer the following sections for details)

- **Wait:** When the file is being executed you have the choice to either wait till the execution is complete or proceed with the next task. Click the list box of the field provided for Wait in the Value field to select Yes or No. Clicking Yes confirms that you wish to wait for the execution to be complete. Clicking No indicates that you wish to proceed.
- **Batch Parameter:** Clicking Yes would mean that the batch parameters are also passed to the executable being started else the batch parameters will not be passed to the executable.

ATTENTION Always select **Y** in Batch Parameter.

For the Parameter Executable earlier mentioned, the map ref num values are

- -1 (if you want to process all the dimensions). The Executable parameter mentioned earlier would be scd,-1
 - If you want to process for a single dimension, query the database table SYS_TBL_MASTER and give the number in the map_ref_num column for the dimension you want to process. These are the ones which come seeded with the install.
9. Execute the batch from Batch Execution by choosing the batch created following the steps mentioned in the preceding sections for a date.

A seeded batch <Infodom>_SCD is provided which has all the required dimensions as different tasks that are part of SCD.

5.3.2 Checking the Execution Status

The status of execution can be monitored using the Batch Monitor screen. You can access this from the Left Hand Side (LHS) menu as follows:

From the **Home** menu, select **Operations**, then select **Batch Monitor**.

For a more comprehensive coverage, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.

The status messages in Batch Monitor are:

- N: Not Started
- O: On Going
- F: Failure
- S: Success

The ICC execution log can be accessed on the application server in the following directory:
\$FIC_DB_HOME/log/ficgen.

The file name will have the batch execution id.

Sample

/dbfiles/home/oracle/OFSAAI/ficdb/log/ficgen

The detailed SCD component log can be accessed on the application server in the directory \$FIC_HOME, go one folder up from there and then accessing the following path:
/ftpshare/<infodom name>/logs

The file name will have the batch execution id.

Sample

/dbfiles/home/oracle/ftpshare/OFSAADEMO/logs

Check the .profile file in the installation home if you are not able to find the paths mentioned earlier.

5.3.3 Load DIM_ACCOUNT through SCD

The SCD population in DIM_ACCOUNT table generates individual numeric SKEYs for every account number with an additional leg skey. Following are the columns that will be loaded during SCD population:

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

This approach replaces the function load in which the table DIM_ACCOUNT is getting loaded through the function, FN_POPDIMACCOUNT. This loads the following columns into DIM_ACCOUNT table:

- V_ACCOUNT_NUMBER
- N_ACCT_SKEY
- N_RCV_LEG_ACCT_SKEY
- FIC_MIS_DATE

Where, the sources are the different product processor tables present in the solution, which are configured in FSI_DIM_ACCOUNT_SETUP_DETAILS table.

5.3.4 DIM_ACCOUNT SCD

Batch <INFODOM>DIM_ACCOUNT_SCD has been introduced with 33 tasks under it. These 33 tasks represent the 33 SCD processes where different product processors would be the source and DIM_ACCOUNT would be the target. MAP_REF_NUMs 188 to 217 have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

DIM_ACCOUNT_SCD has been introduced with 33 tasks under it. These 33 tasks represent the 33 SCD processes where different product processors would be the source and DIM_ACCOUNT would be the target. MAP_REF_NUMs 188 to 217 have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

Depending on the requirement by an application, a task can be excluded or included from the batch execution.

5.3.5 Load DIM Tables through SCD

Batch <INFODOM>_SCD has been introduced with 129 tasks under it. These 129 tasks represent the 129 SCD processes where different staging tables would be the source and Dimension Tables would be the targets. The required SCDs have been introduced into SYS_TBL_MASTER table, and subsequently into SYS_STG_JOIN_MASTER.

Depending on the requirement by an application, a task can be excluded or included from the batch execution.

5.3.6 Improve SCD Performance

SCD performance can be improved by providing hints and session alter statements. This requires the presence of the following four columns in SYS_TBL_MASTER:

- merge_hint
- select_hint
- session_enable_statement
- session_disable_statement

MERGE_HINT and SESSION_ENABLE_STATEMENT in SYS_TBL_MASTER

Table Name	Stage Table Name	Stage Table Name	Session Enable Statement
DIM_ACCOUNT	STG_LOAN_CONTRACTS_V	/*+ parallel (DIM_ACCOUNT,10) */	"alter session enable parallel dml query", "alter table DIM_ACCOUNT nologging parallel 10"

- All the tasks can be executed in parallel. This might cause the N_RCV_LEG_ACCT_SKEY to have an incremental value as compared to N_ACCT_SKEY.
- Execute the SQL file with all the SESSION_DISABLE_STATEMENTS, after the successful completion of the SCD batch.
 - Once the DIM_ACCOUNT table is populated using this approach, you will not be able to use the initial approach (FN_POPDIMACCOUNT) as this will lead to skey conflict.
 - Ensure that you have set the value of the sequence seq_dim_account_scd as max (value of key in DIM_ACCOUNT) + 1, before moving from old to new approach.
 - The F_LATEST_RECORD_INDICATOR for an existing DIM_ACCOUNT data already loaded by the function should be updated to 'Y' before running the SCD, failing which a new skey might get generated for the same account number.
 - SCD execution occurs based on the GAAP code which is configured in SETUP_MASTER table. This has been introduced to tackle the scenario of multiple GAAP codes. Whether or not there exist multiple GAAP codes, SETUP_MASTER should be manually configured as follows:

SETUP_MASTER configuration

V_COMPONENT_CODE	V_COMPONENT_DESC	V_COMPONENT_VALUE
DEFAULT_GAAP	DEFAULT_GAAP	USGAAP

Where V_COMPONENT_VALUE should be manually populated with the required GAAP code.

5.3.7 Handling Multiple GAAP Codes

When multiple GAAP codes exist for the same account number for the same MIS date, configure the SETUP_MASTER table manually as mentioned in the preceding section:

V_COMPONENT_VALUE will hold the GAAP code for which the SCD is to be executed.

If there are different GAAP codes for two distinct account numbers for the same MIS date, then the SCD has to be executed for each GAAP code by changing the V_COMPONENT_VALUE manually in setup_master table. The SETUP_MASTER table should have only one record WHERE V_COMPONENT_DESC = 'DEFAULT_GAAP'.

6 Exchange Rate History Population

This chapter discusses the following topics:

- [Introduction](#)
- [Execution of Currency Exchange Rates Population T2T](#)
- [Currency Execution Rates - Batch Execution](#)
- [Exchange Rate History Population](#)

6.1 Introduction

Exchange Rate History entity stores the exchange rates between the currencies for an effective date from one or multiple sources.

Exchange Rate History population should be executed before any fact table is populated to ensure exchange rates between currencies are available prior. Exchange Rate History entity is loaded by means of Table to Table Transformation process.

Following is the seeded Table-to-Table definition that loads data into Exchange Rate History:

T2T Definition Exchange Rate History

T2T Definition Name	Source Table(s)	Destination Table
T2T_EXCHANGE_RATE_HIST	STG_EXCHANGE_RATE_HIST	FSI_EXCHANGE_RATE_HIST

6.2 Execution of Currency Exchange Rates Population T2T

The batch <INFODOM>_POP_EXCHANGE_RATES needs to be executed to populate fsi_exchange_rates as the entries in setup_master are seeded during installation.

FSI_EXCHANGE_RATES table has to be loaded prior loading any of the other Account Summary tables.

- Metadata Browser
- Common Account Summary

6.3 Currency Execution Rates - Batch Execution

A seeded batch, <Infodom>_POP_EXCHANGE_RATES has to be executed for the required MIS Date.

Alternatively, following steps will help you create a new batch:

1. From the **Home** menu, click **Operations** and select **Batch Maintenance**.
2. Click **New Batch** ('+' symbol in Batch Name container). Enter the Batch Name and Description.
3. Click **Save**.
4. Click the check box in the Batch Name container to select the Batch, you created in the earlier step.
5. Enter the Task ID and Description.

6. Select Load Data from the Components list.
7. Select the following from the Dynamic Parameters List:
 - Datastore Type: Select the appropriate datastore from the list.
 - Datastore Name: Select the appropriate name from the list.
 - IP address: Select the IP address from the list.
 - Load Mode: Select Table to Table from the list.
 - Source Name: Select the <T2T Source Name> from the list.
 - File Name: Select the T2T name for the source stage channel table you want to process.
8. Click **Save**.
 Data file name will be blank for any Table to Table Load mode. Default value refers to currency calculation. If there is any need for currency conversion in T2T transactions, Default value has to be provided.
9. Execute the batch created in the preceding steps.

6.4 Exchange Rate History Population

To execute the T2T component from OFSAA Infrastructure ICC framework (accessed through the application Batch Operations screen).

A seeded batch, **<INFODOM>_aCRM_CommonTasks - Task4** has to be executed for the required date.

The screenshot displays the 'Batch Execution' window. The 'Batch Details' section lists several tasks, with 'CRM60NFO_aCRM_CommonTasks' selected. The 'Task Details' section shows a table of tasks, with 'Task4' highlighted. The 'Information Date' is set to 12/31/2019. An 'Execute Batch' button is visible at the bottom.

Task ID	Task Description	Metacata Value	Component ID	Precedence	Task Status
Task1	Update SetupMaster with Current FIC Mta Date	Fa_con_DT_Setup	TRANSFORM DATA		H
Task2	Populate Time Dimension	Dim_Rates_Population	TRANSFORM DATA		H
Task3	Populate Account Dimension	tl_popDimAccount	TRANSFORM DATA		H
Task4	Populate Currency Exchange Rates	T2T_EXCHANGE_RATE_HIST	LOAD DATA		N

Alternatively, following steps will help to create a new batch task for Loading Historical Exchange Rates:

1. From the **Home** menu, select **Operations**, then select **Batch Maintenance**.
2. Click **New Batch** ('+' symbol in Batch Name container). Enter the Batch Name and Description.
3. Click **Save**.
4. Click the check box in the **Batch Name** container to select the Batch, you created in the earlier step.
5. Enter the **Task ID** and **Description**.
6. Select **Load Data** from the components list.
7. Select the following from the Dynamic Parameters List and then click Save.
 - Datastore Type: Select appropriate datastore from the list.
 - Datastore Name: Select appropriate name from the list.
 - IP address: Select the IP address from the list.
 - Load Mode: Select Table to Table from the list.
 - Source Name: Select **<T2T Source Name>** from the list.
 - File Name: Select the table to table transformation T2T_EXCHANGE_RATE_HIST.Data file name will be blank for any Table to Table Load mode.
8. Repeat steps 4 to 8 for adding the remaining T2Ts within the same batch definition.
9. Execute the batch created in the preceding steps.

For more information, see *Oracle Financial Services Analytical Applications Infrastructure User Guide*.
10. Check T2T component logs and batch messages to check the status of load.

T2T component can fail because of following cases:

 - Unique constraint error: Target table may already contain the primary keys that are part of the staging tables.
 - NOT NULL constraint error: Do not have values for NOT NULL columns in the target table.

6.4.1 Checking the Execution Status

The status of execution can be monitored using the batch monitor screen.

The status messages in batch monitor are:

- N: Not Started
- O: On Going
- F: Failure
- S: Success

The execution log can be accessed on the application server in the `$FIC_DB_HOME/log/t2t` directory. The file name will have the batch execution id.

6.4.2 Validating the Exchange Rate

The Function Fn_ratevalidation is executed using the task. Edit the "Task1" of the batch "<INFODOM>_FN_RATEVALIDATION" and pass the below parameters to the task:

- Starting date
- End date

All the exchange rates present in FSI_EXCHANGE_RATE_HIST table whose 'effective date' lies in the range of these values will be validated on execution of this batch. The validated rates will be available in the table FSI_EXCHNG_RATE_DIRECT_ACCESS.

Rate Triangulation is also achieved during this process.

7 Developing a New Cube

This section details the steps to be performed by the user for developing a new cube. Make sure that the existing cubes do not provide the required analytics / reporting coverage before deciding to define a new cube. In case user would like to see measures against a new dimension that is not part of the existing seeded metadata, then suggest including the new dimension as part of the existing cubes instead of creating a new cube. As a prerequisite, user should have defined datasets, measures, hierarchies and dimensions before defining a cube.

Procedures to develop a new cube involves the following steps:

Step 1 - Add Cube: From Unified Metadata Manager, select Business Metadata Management, then select Cube. Specify the MDB details that will be created in ESSBASE.

Step 2 - Include Dimensions: Include dimensions that are part of the cube definition. Users mandatorily need to include TIME and MEASURE dimensions.

Step 3 - Specify Variations: Specify variations between each of the measures to the respective dimensions. All the measures that are part of the cube need not vary against all of the dimensions. Depending on business needs, variations can be specified to control the rollup of measures against a set of dimensions.

Step 4 - Specify Dataset: Specify dataset corresponding to the selected dimensions and measures. Data set will supply required data to the cube.

Step 5 - Specify Node Level Formula: If node level formula's are required to be specified for the nodes within the hierarchy, then they can be specified in this UI.

Step 6 - Save and Build: Save the cube. Define and execute batch in ICC to build the cubes.

For more information on Cubes, see *Cubes* under *Unified Metadata Manager* chapter in Oracle Financial Services Analytical Applications Infrastructure User Guide.

8 List of Hard-Coded Members

Following are the dimension members that are hard-coded within the application:

Table Name	Column Name	Expected Values
DIM_CUSTOMER_TYPE	V_CUST_CATEGORY	C
	F_LATEST_RECORD_INDICATOR	Y
FCT_CRM_ACCOUNT_SUMMARY	V_SCENARIO_CODE	PLAN, BUDGET
FCT_OPPORTUNITY_ACTIVITY	V_ACTIVITY_STATUS	O, C
DIM_BANDS	V_BAND_TYPE	AGEONBOOK TURNOVER AGE INCOME ACCT_ATTRITION_SCORE CUST_CR_RISK_SCORE NO_OF_ACCOUNTS RESPRATE DELQBAND
FCT_ACCOUNT_PROFITABILITY	N_REP_LINE_CD	98000 - Net Income Before Taxes 98500 - Tax Expense 99000 - Net Income After Taxes 107100 - Number of Customers 107130 - Number of Open Customers 107200 - Number of Accounts 107230 - Number of Open Accounts 107300 - Attrition Rate
FCT_TXN_CHANNEL	V_F_CHNL_TYPE	MONETARY NONMONETARY
	F_F_TXN_DR_CR_IND	C D
DIM_PRODUCT	V_PROD_TYPE	CARDS RB DEPOSITS CASA AUTOLOAN TD MORTGAGE

9 Run Rule Framework

In cases where data is required to be loaded for fact tables in multiple runs, the OFSAAI Rule Run Framework comes in handy. For example, for population of FCT_CRM_ACCOUNT_SUMMARY, the parameters of the batch execution include a parameter \$RUNSK = -1.

(Refer to the parameters of the batch ##INFODOM##_aCRM_CRM_Acc_Summ,Task1).

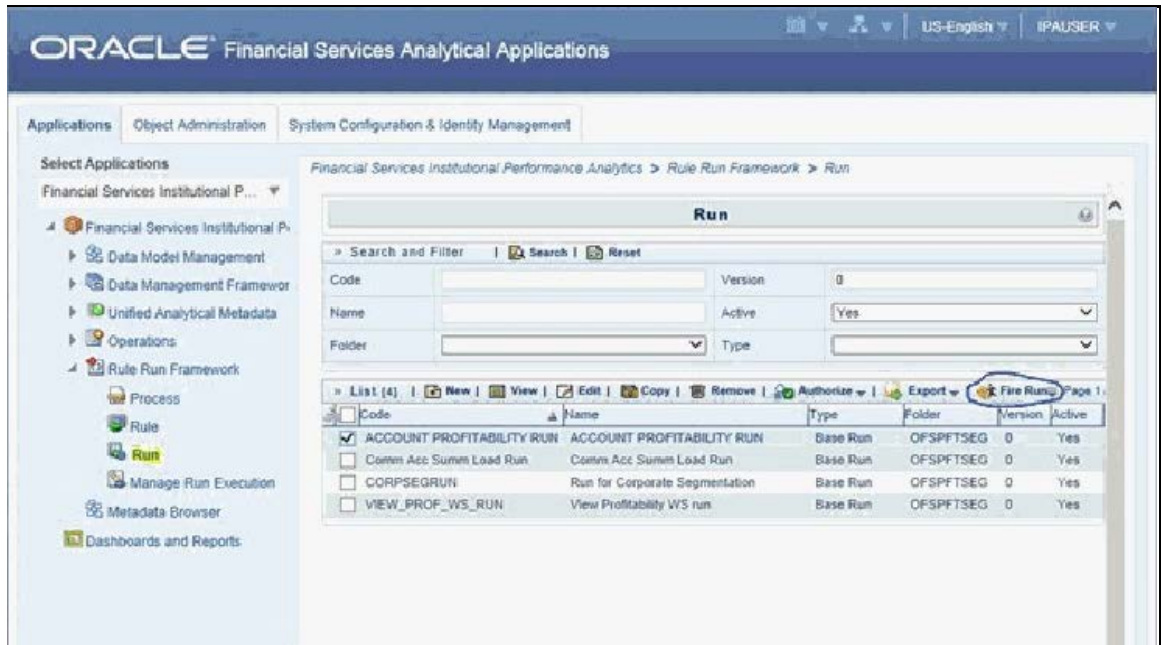
This batch execution loads the column N_RUN_SKEY in FCT_CRM_ACCOUNT_SUMMARY as -1. This will be a default run from the seeded batch. In order to be able to enter data for multiple runs, the batch tasks can be defined in Rule Run Framework. This will then create a batch internally which will load data for a different run into the fact table.

9.1 Executing a Seeded Run

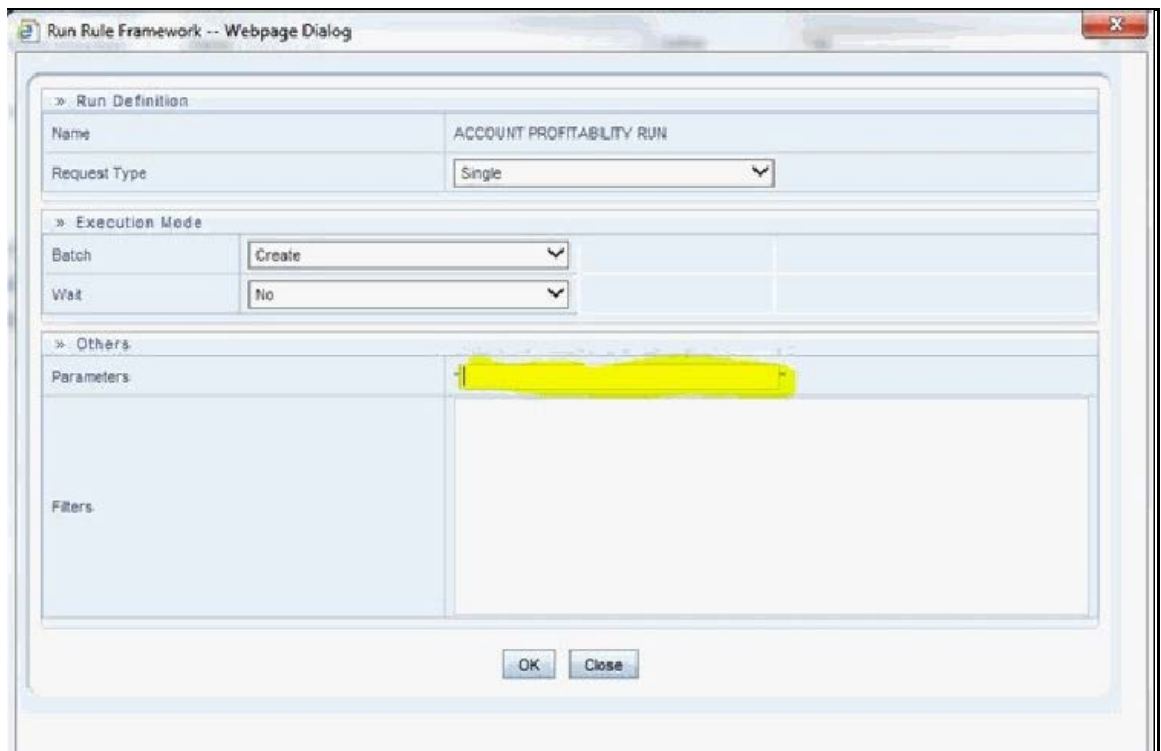
1. Navigate to **Rule Run Framework > Run**.



2. Choose a Run by checking the box before it and click **Fire Run**.



3. Enter the parameters required to execute the run (refer to details of individual runs) and click **OK**.



The following message will be displayed: Fire run successful

4. Execute the batch by navigating to **Operations > Batch Execution** and select the latest batch created in the Run Rules Framework module.

9.2 Runs Available for OFS Performance Analytics Cloud Service

The following are the runs available for OFS Performance Analytics Cloud Service:

- **ACCOUNT PROFITABILITY RUN:** Run for loading FCT_CRM_ACCOUNT_SUMMARY and FCT_ACCOUNT_PROFITABILITY

Parameters to be entered as follows:

```
##RCY##", "<INFODOM>", "FCT_ACCOUNT_PROFITABILITY
```

NOTE

There are no quotes at the beginning and the end of the string. Consider as if the quotes are enclosing the delimiter (comma).

- **Comm Acc Summ Load Run:** Run for loading FCT_COMMON_ACCOUNT_SUMMARY
- **VIEW_PROF_WS_RUN:** Run for executing Web Service
- **RETSEGRUN:** Run for executing Retail Segmentation Rule

10 PA Metrics Computation Logic

10.1 Projection Logic

We have data up to the last MIS Date based on which projections are done on Replines in FCT_ACCOUNT_VALUE_FORECAST. For each repline, for last MIS Date, there is a projection on every projected date (generally last date of every month)

PA Metrics calculation

The following table list the available web services (APIs) for OFSPA Cloud Service:

Web Service	Input	Output	Version
Account Level Metrics (without UDM)	Account Number, Date	ROTA, RAROC, ROE, Total Expenses, Total Income, Net Income	
Customer Level Metrics (without UDM)	Customer Number, Date	ROTA, RAROC, ROE, Total Expenses, Total Income, Net Income	
Account Segmentation	Account Number, Date	Account Segment Code	
Customer Segmentation	Customer Number, Date	Customer Segment Code	
Metrics on existing Customer/Account with UDM	Account Number, Date of Report, {Date of UDM, UDM} repeated for all UDMs	ROTA, RAROC, ROE, Total Expenses, Total Income, Net Income, Customer ROTA, Customer RAROC, Customer ROE, Customer Total Expenses, Customer Total Income, Customer Net Income	
Metrics for New Account of existing Customer	Customer Number, Date of Report, Account Schedule	ROTA, RAROC, ROE, Total Expenses, Total Income, Net Income, Customer ROTA, Customer RAROC, Customer ROE, Customer Total Expenses, Customer Total Income, Customer Net Income	
Metrics for New Account of New Customer	Date of Report, Account Schedule	Account Level ROTA, RAROC, ROE, Total Expenses, Total Income, Net Income	

NOTE

Segment Metrics are done through batch process. The segment metrics considering the life time calculations of all the accounts in that segment.

At Present Customer level segment metrics are based on the segment of first account identified (by the system) of the customer. The segment averages as computed against the identified account segment would be reported.

10.2 Data Treatment for Months

1. Data to be considered for end of every month

- **Historical Data:** Take data for each month (as of month end date) from FCT_ACCOUNT_PROFITABILITY (aggregated under NIBT hierarchy)
- **Projected Data:** For Last MIS Date, take data from FCT_ACCOUNT_VALUE_FORECAST for each subsequent Projected Date (aggregated under NIBT hierarchy: for aggregations we will consider all forecasted replines whether leaf or node)
- Union the above two queries so that we get a data set from first month end date to last month end date. This will be inclusive of both actual + forecasted.
- Discount each row in this dataset to the account start date; N_Origination_date
- Sum up all the discounted values
- Arrive at the value of NIBT

NOTE

Discounting: For projection purpose, to arrive at the Net Present Value (NPV) of the future stream of NIBT; a discount factor be applied to arrive at expected value of NIBT.

2. Treatment of Data of Month End Values and for Missing Months

- **Actual Data with Missing Value:** For profitability calculations or reporting the missing value should be considered as 0.
- **For Forecasting of data:** If there is a missing value in the actual data then, the missing value treatment should be applied only to fill in a value which can be used for forecasting.

3. Calculation of Profitability Metrics

- **Return on Total Assets (ROTA) =**
 - Loan Products- NIBT/ Sanctioned Limit (For Loans);
 - Line of Credit Products - NIBT/ Max of EOP Balance for Line of Credits
- **Risk-adjusted return on capital (RAROC) = NIBT/ Sum of Unexpected Losses**
 - Unexpected Losses comprise of- Operating Risk Capital - Economic, Credit Risk Capital - Economic, Market Risk Capital - Economic.
- **Return on Equities (ROE)**
 - Loan Products- NIBT/ Sanctioned Limit (For Loans); [N_SANCTIONED_LIMIT for Loans (first MIS Date), FCAS]
 - Credit Products -
 - NIBT/ Max of EOP Balance for Line of Credits [max(N_EOP_BAL), FCAS for Line of Credit (across all MIS Dates)]
 - Average of EOP Balance for all deposits (including term deposits) [average(N_EOP_BAL), FCAS for all deposits (including term deposits)]
- **TOTAL REVENUE = NET INTEREST INCOME + NON-INTEREST INCOME** [Non-Interest Revenue' + 'Indirect Non-Interest Income' + 'Other Revenue']

- **Total Expenses:** Total Operating and Non-Operating Expenses [Net Credit Losses = Credit Losses- Recoveries of amounts previously written-off)
- **Net Income:** Net Income Before Tax
- **Currency:** Metrics would be reported in reporting Currency.

NOTE

As per the current functionality, by default, if there is no requested currency code, the customer would be showing in reporting currency and the Accounts are shown in its natural currency. In case of any requested currency code, both customer and currency would be shown in requested currency code.

- Customer Metrics
 - ROTA and RAROC would be aggregated only against asset products.
 - All other metrics would be aggregated against all products.
 - Customer Metrics would be reported in Reporting Currency.
- Customer Level Metrics Calculation
 - ROTA: $(\text{Sum of ROTA Numerator over all asset accounts}) / (\text{sum of ROTA denominator over all asset accounts})$
 - RAROC: $(\text{Sum of RAROC Numerator over all asset accounts}) / (\text{sum of RAROC denominator over all asset accounts})$
 - ROE: $(\text{Sum of ROE numerator over all accounts}) / (\text{Sum of ROE denominator over all accounts})$
 - TOTAL INCOME, TOTAL EXPENSES, NET INCOME: Sum over all accounts
- Segment Metrics
 - Segment Level Metrics will be reported against Segmentation done at Account Level
 - Segment Metrics to be reported in Reporting Currency.

10.3 Aggregation and Projection Logic for Forecasting of Replines under PA

Current Behavior/Functionality:

At present the following Replines are considered for forecasting (@ (fsi model parameters table):

N_REP_LINE_CD	Repline Name	Whether Forecasted
82100	Interest Income	Yes
82200	Transfer Pricing Credit	Yes
83300	Interest Expenses	Yes

N_REP_LINE_CD	Repline Name	Whether Forecasted
90000	Transfer Pricing Charge	Yes
91900	Customer Non-Interest Income	Yes
94000	Non-Customer Non-Interest Revenue	Yes
107420	Indirect Non-interest Income	Yes
107501	Other Revenue	Yes
107502	Net Credit Losses	Yes
107503	Operating Expenses	Yes
107504	Non-Interest Income	Yes
107505	Outstanding Balance	Yes
87000	Utilization Rate	Yes
87500	Revolve Rate	Yes
90000	Purchases	Yes
86000	Payments	Yes

Observation:

1. From the above, Interest Income (82100) and Transfer Pricing Credit (82200) are two leaf level Replines which are the members of the Node Total Interest Income (82000) are being forecasted while the other leaf lever members coming under Node Total Interest Income are not considered for forecasting

All Leaf Level members of the Node Total Interest Income is given below:

Leaf members of Node Total Interest Income		
82000	Total Interest Income	Node
84000	Amortization of Discount for Liability	Leaf
84100	Amortization of Premium for Asset	Leaf
82900	Central Bank Int. Income	Leaf
83000	Credit for Float	Leaf
82800	Credit for Liquidity	Leaf
82700	Credit for Other Allocated Liabilities	Leaf
82300	Customer Break Funding Fees	Leaf
82100	Interest Income	Leaf, Forecasted
82200	Transfer Pricing Credit	Leaf, Forecasted

2. From the above, Interest Expense (83300) and Transfer Pricing Charges (90000) are two leaf level Replines which are the members of the Node Total Interest Expenses (82001) are being forecasted while the other leaf lever members coming under Node Total Interest Expenses are not considered for forecasting.

Leaf members of Node Total Interest Expenses		
82001	Total Interest Expense	This is a node
82500	Amortization of Discount for Asset	Leaf
82600	Amortization of Premium for Liability	Leaf
83500	Charge for Basis Risk	Leaf
83900	Charge for Central Bank Reserves	Leaf
83400	Charge for Liquidity	Leaf
83600	Charge for Optionality	Leaf
83700	Charge for Other Allocated Assets	Leaf
84500	Credit for Equity	Leaf
84200	Economic Provision	Leaf
82400	Pricing Incentive	Leaf
83300	Transfer Pricing Charge	Leaf, Forecasted
83200	Interest Expenses	Leaf, Forecasted

Reason for Considering Specific leaf level Replines for forecasting in-place of Node level forecasting

Reason for forecasting interest income, Transfer Pricing Credit, interest expense and Transfer Pricing Charges at leaf members and not their respective parent nodes, to apply variations on interest income/expense and transfer pricing charge/credit separately in what-if analysis and monitor sensitivity at each of these leaf levels.

Doing projections, at the parent node, level would not allow monitoring and reflecting variations at such micro levels; as both interest income & Transfer Pricing Credit belong to Parent Node Total Interest Income, as well as both Interest Expense and Transfer Pricing Charges belong to Parent Node Total Interest Expense.

Issues arising out of such approach

If only the above leaf level replines are projected, and other leaf level replines under their respective node replines are not projected, leading to an incorrect aggregation of NIBT in forecasted values.

Suggested Approach

Repline Forecasting: IPA

- The leaf level members under Total Interest Income & Total Interest Expense, which are missed out in forecasts, should be aggregated to one head each respectively and then forecasted.
- The head, can be an additional node in the income statement hierarchy

Details: Other than the 4 leaf level replines viz. Interest Income, Interest Expense, TP Charge and TP Credit, aggregate the remaining replines under the nodes 'Total Interest Income' and 'Total Interest Expense' to form two notional replines under their heads respectively. Then forecast for these two notional replines and use for aggregation. These notional replines would be named as "Other Interest Income" and "Other Interest Expenses"

The node replines of 'Total Interest Income' and 'Total Interest Expense' are not projected; instead only 4 child leaf replines from the hierarchy viz interest income/expense, TP charge/credit are projected.

Introduce two new replines, which would be aggregations of several replines as:

- Additional repline 1 (Other Interest Income) = sum(All leaf replines which are child of Total Interest Income)-Interest Income - TP Credit

When calculating total interest income; it will be, total interest income = other interest income+ interest income+ TP Credit

- Additional repline 2 (Other Interest Expenses) = sum(All leaf replines which are child of Total Interest Expense)-Interest Expense - TP Charge

When calculating total interest expense; it will be, total interest expense = other interest expense+ interest expense+ TP charges

These 2 new replines would be a part of the income statement hierarchy.

When aggregating for NIBT in the historical data, these two new replines would not get considered.

When aggregating for NIBT in projected data, the new replines 1&2 should roll up into Total Interest Income and Total Interest Expense respectively.

New Structure of Repline Hierarchy:

Total Interest Income:

Leaf members of Node Total Interest Income		
82000	Total Interest Income	Node
	+ Other Interest Income - New Repline	Node
	84000 Amortization of Discount for Liability	
	84100 Amortization of Premium for Asset	
	82900 Central Bank Int. Income	
	83000 Credit for Float	
	82800 Credit for Liquidity	
	82700 Credit for Other Allocated Liabilities	
	82300 Customer Break Funding Fees	
82100	Interest Income	Leaf, Forecasted
82200	Transfer Pricing Credit	Leaf, Forecasted

Total Interest Expense:

Leaf members of Node Total Interest Expenses		
82001	Total Interest Expense	This is a node
	+ Other Interest Expenses - New Repline	Node
	82500 Amortization of Discount for Asset	

Leaf members of Node Total Interest Expenses		
	82600 Amortization of Premium for Liability	
	83500 Charge for Basis Risk	
	83900 Charge for Central Bank Reserves	
	83400 Charge for Liquidity	
	83600 Charge for Optionality	
	83700 Charge for Other Allocated Assets	
	84500 Credit for Equity	
	84200 Economic Provision	
	82400 Pricing Incentive	
83300	Transfer Pricing Charge	Leaf, Forecasted
83200	Interest Expenses	Leaf, Forecasted

Key Point:

- The New nodes other interest income and other interest expenses would be a part of the income statement hierarchy
- The forecasts required for PCD, Profitability, LTV and What-if. The inputs would be required at this granularity.

Addendum and Update

- Repline Codes: 81000 (Other Interest Income) and 81001 (Other Interest Expenses) are finalized and accepted.
- Operational Signage: Operational Signage for Both Replines would be +. These replines are having other leaf replines under each, which are in turn aggregated to Total Interest Income (82000) and Total Interest Expenses (82001) respectively.
- The New nodes other interest income and other interest expenses would be a part of the income statement hierarchy
- The forecasts required for PCD, Profitability, LTV and What-if. The inputs would be required at this granularity.
- These new Notional Replines needs to be introduced under What-if Logic and for calculation under Advanced What-If Logic.
- What-if logic would be in accordance with the changes to be made in the Repline Hierarchy (on Aggregation and Projection Logic for forecasting of Replines under PA). In case of addition to any new (or notional repline), the same would be required to be introduced under what-if logic.
- For what if Advanced Logic; the newly introduced replines to be added to the calculation while establishing the impacts (happening through Linear Regression Approach)

