Oracle® Financial Services Cash Flow Engine

Reference Guide

Release 6

Part No. E38626-02

October 2014

Oracle Financial Services Cash Flow Engine Reference Guide, Release 6

Part No. E38626-02

Primary Author: Arpana Danayak

Contributor: Christopher Spofford, Brad Bruckschen

Contents

5 Cash Flow Calculations

7 Distributed Originations

8 Transfer Pricing Option Costs

9 Rate Conversion

12 Autobalancing

13 Forward Starting Instruments

14 Historical Simulation

15 Derivatives in Stochastic Processes

16 Bonds with Embedded Options

17 Financial Elements

A Cash Flow Edit Error Messages

Glossary

Send Us Your Comments

Oracle Financial Services Cash Flow Engine Reference Guide, Release 6

Part No. E38626-02

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document. Your feedback is important, and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the procedures?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

If you find any errors or have any other suggestions for improvement, then please tell us your name, the name of the company who has licensed our products, the title and part number of the documentation and the chapter, section, and page number (if available).

Send your comments to us using the electronic mail address: financialservices_ww@oracle.com

Please give your name, address, electronic mail address, and telephone number (optional).

If you need assistance with Oracle software, then please contact your support representative or Oracle Support Services.

If you require training or instruction in using Oracle software, then please contact your Oracle local office and inquire about our Oracle University offerings. A list of Oracle offices is available on our Web site at www.oracle.com.

Preface

Intended Audience

Welcome to Release 6 of the *Oracle Financial Services Cash Flow Engine Reference Guide.*

The Oracle Financial Services Cash Flow Engine Reference Guide provides useful guidance and assistance to:

- Technical end users
- Functional end users
- Data Administrators
- Consultants
- Systems Analysts
- System Administrators
- Other MIS professionals

See Related Information Sources on page xix for more Oracle product information.

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 or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Structure

1 Overview

2 Boundaries and Limitations

This chapter lists the boundaries and limitations that are applicable to Oracle Asset Liability Management, Oracle Funds Transfer Pricing, and the configuration of Oracle Financial Services Analytical Applications (OFSAA) database.

3 Consolidation of Results

Users can combine the result sets from multiple processing runs if needed for reporting. This chapter contains the following sections, which describe how to merge results from multiple Oracle Asset Liability Management processes:

4 Detail Cash Flow Audit Options

This chapter describes the Detail Cash Flow Audit Options, which enable you to view daily cash flow results.

5 Cash Flow Calculations

Oracle Asset Liability Management uses a cash flow engine to ensure modeling consistency across the Oracle Financial Services Analytical Applications (OFSAA) suite of products.

This chapter describes the calculations performed by the OFSAA Cash flow engine and defines concepts that are vital in forming a complete understanding of the capabilities of the cash flow engine.

6 Cash Flow Dictionary

This chapter includes a list of columns required for this processing as well as a list of columns required to run the Oracle Financial Services Analytical Applications (OFSAA) cash flow edits.

7 Distributed Originations

This chapter describes how new business instruments are generated as part of an Oracle Asset Liability Management (ALM) forecast to replace balances that have run off for a particular product.

8 Transfer Pricing Option Costs

This chapter describes the option cost calculations implemented in Oracle Funds Transfer Pricing, including theoretical overview, the mathematical details of the calculations, and examples.

9 Rate Conversion

This chapter defines interest rate formats used within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing, describes which format is used in a particular process and what type or types of conversion are used; and outlines the algorithms used for such conversions.

Interest rates come in a variety of formats. Within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing (FTP), interest rates are used for multiple purposes, with each rate requiring a specific format. The system must apply conversion formulas to translate the interest rates from their starting format into the format proper for its use in any given process.

10 Forecast Rate Calculations

This chapter describes how forecast rate assumptions are created and managed within the Forecast Rate Scenario's user interface.

Forecast assumptions for currency exchange rates and interest rates are defined within the Oracle Asset Liability Management (ALM) Forecast Rates assumption rule. The resulting rates can be calculated and viewed through the user interface. These calculations are also used during Oracle ALM deterministic processing, at which time the resulting rates can be output for auditing or reporting purposes.

11 Monte Carlo Analytics

This chapter describes the details of stochastic processing used within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing (FTP). This includes the detailed architecture of the Monte Carlo rate generator that is used for stochastic forecasts of interest rates and calculation of market value, Value-at- Risk, and Earnings-at-Risk.

12 Autobalancing

This chapter describes how to setup the Asset Liability Management (ALM) – Auto-balancing process and demonstrates how the process works.

13 Forward Starting Instruments

14 Historical Simulation

15 Derivatives in Stochastic Processes

16 Bonds with Embedded Options

17 Financial Elements

This chapter contains the descriptions and calculation details of Oracle Asset Liability Management (ALM) financial elements.

A Cash Flow Edit Error Messages

Glossary

Related Information Sources

Related Documents

For more information about using Oracle Financial Services Analytical Applications (OFSAA), see the following related documents:

- Oracle Financial Services Analytical Applications Data Model Utilities User Guide.
- Oracle Financial Services Analytical Applications Data Model Data Dictionary.
- Oracle Financial Services Asset Liability Management (OFSALM) User Guide.
- Oracle Financial Services Funds Transfer Pricing User Guide.

Conventions

The *Oracle Financial Services Cash Flow Engine Reference Guide* uses the following text and font characteristics:

Upper case text connected by underscores	Used to denote table names. For example, FSI_O_RESULT_MASTER table or FSI_O_PROCESS_ERRORS table.
Lower case text connected by underscores	This convention is used to designate column name within a table. Examples of this convention include market_value and transfer_rate.

Symbols

■ Bullets indicate a list of items or topics.

1. Numbered lists are used for sequential steps in completing a procedure.

How to Use This Guide

The *Oracle Financial Services Cash Flow Engine Reference Guide* is a single, centralized source of information about the underlying structure and processing of OFSAA. The information provided in this guide can assist you in the understanding and implementation of OFSAA features and functions, as well as the integration of OFSAA with other tools.

For example, the *Oracle Financial Services Cash Flow Engine Reference Guide* can help you:

- Incorporate additional account instrument data
- Create custom reports and queries
- Integrate OFSAA with other product systems
- Integrate additional tools for use with OFSAA
- Validate results

Here are several examples of how this guide can be used:

Example#1

Suppose your institution has merged with a separate financial institution and you need to incorporate new instrument account data into your OFSAA database. The cash flow engine data dictionary and table/column detail provided in this guide can be used to help define the data requirements for integrating additional data sources into your installation. If the instrument requires cash flow processing, appropriate defaults for the OFSA cash flow columns can be researched using the Cash Flow Column Definitions.

Example #2

Suppose you are trying to validate a Market Value result written to the Result Master table. You can review the Chapter - Cash Flow Calculations, page 5-1 to better understand the related calculation logic.

How Not to Use This Guide

Do not use this guide to plan modifications to OFSAA tables.

Except for instrument tables and tables that you create for your database (user-defined tables), the structure of the tables in the OFSAA database should not be altered in any way. Modifying OFSAA Reserved Objects limits your ability to upgrade to future releases of OFSAA. In addition, altering such tables could interfere with problem resolution and assistance from Oracle Support Services.

The OFSAA suite of products has been constructed to provide for user customization without altering any of the OFSAA Reserved Objects. To satisfy your business needs, Oracle Financial Services Analytical Applications Infrastructure provides the functionality needed to register customized instrument tables and other user-defined objects for use the OFSAA suite.

Do not rely on upward compatibility of the data model.

Oracle reserves the right to change the structure of the OFSAA tables and to change the meaning of, add, or delete codes and other data in future releases. Upward compatibility of the data model is not guaranteed. For example, if you create a report to display metadata information about the OFSAA database by querying the OFSAA Metadata tables, you are not guaranteed that the report will work properly after an upgrade.

1

Overview

Introduction to Oracle Financial Services Cash Flow Engine Reference Guide

The Oracle Financial Services Cash Flow Engine Reference Guide emphasizes business analysis and provides definitions, analytical concepts, processes, and calculations used in the processing of the Oracle Asset Liability Management and Oracle Funds Transfer Pricing applications. The information provided includes:

- Cash flow engine data requirements, cash flow and repricing event logic, formula calculations and required rate formats, and conversion methodologies used to produce cash flows for both Asset Liability Management and Funds Transfer Pricing.

- Detailed concepts, processes, and formulas used in the generation of Option Costs in Transfer Pricing and Monte Carlo based projections in Asset Liability Management, including Rate Generator computations, analytical calculations, and Term Structure parameters.

- Instructions for generating detailed cash flow output for review and audit purposes and for consolidating the results of multiple ALM processes.

The reference guide assumes that the user has a basic understanding of:

- The Oracle Financial Services Analytical Applications suite of products, particularly Oracle Asset Liability Management and Oracle Funds Transfer Pricing.

- Associated analytic theory and concepts.

This document contains the following chapters:

- Detail Cash Flow Audit Options, page 4-1
- Cash Flow Calculations, page 5-1
- Cash Flow Dictionary, page 6-1
- Distributed Originations, page 7-1
- Transfer Pricing Option Cost, page 8-1
- Rate Conversion, page 9-1
- Forecast Rate Calculations, page 10-1
- Monte Carlo Analytics, page 11-1
- Auto-balancing, page 12-1
- Oracle Financial Services Analytical Applications (OFSAA) Financial Elements, page 17-1

2

Boundaries and Limitations

This chapter lists the boundaries and limitations that are applicable to Oracle Asset Liability Management, Oracle Funds Transfer Pricing, and the configuration of Oracle Financial Services Analytical Applications (OFSAA) database.

This chapter covers the following topics:

- Oracle Asset Liability Management
- Oracle Funds Transfer Pricing
- Database Configuration

Oracle Asset Liability Management

The following limitations are applicable to this product.

Modeling Buckets

There is a maximum of 240 modeling buckets available for processing. There is no limit on the modeling horizon. However, the user interface display of dates within the Time Bucket Rule is limited to dates less than or equal to 12/31/2499.

Because the Oracle ALM simulation engine uses the frequency of the modeling bucket, as expressed in units and multipliers, this user interface restriction does not impact the calculations or the output into the result tables. To model periods extending beyond the year 2499, enter bucket frequencies as needed. Although dates beyond the year 2499 do not appear in the user interface, the cash flow engine can properly handle them.

Dynamic Buckets

There is a maximum of 240 Interest Rate GAP and Liquidity GAP buckets and no limit to the number of dynamic start dates. GAP bucket display of dates is restricted, as with the modeling buckets, to dates less than or equal to 12/31/2499. As described in the modeling bucket section, this restriction does not impact the calculation or the output of dynamic results.

Forward Rate Agreements (FRAs)

FRAs can be modeled only through the Instrument Table, they cannot be modeled through Forecast Balance or Transaction Strategies.

Product Characteristics

The valid range for Option Adjusted Spread is between 5.000 and -5.000.

Transaction Strategies

The maximum balance or deferred balance within a single entry in the Transaction Strategies is 9,999,999,999. To create transactions with larger values, create multiple entries in this interface.

The maximum origination date that can be entered is 12/31/2499.

Terms and frequencies are restricted to a maximum value of 999.

Stochastic Rate Indexing

There is a maximum of eight components to the Rate Index formula per Interest Rate Code (IRC) and term.

Rate Management Term Structure Parameters

Valid values for mean reversion are between .00001 and 1.00000.

Valid values for volatility are between 0.000001 and 9.999999.

When using the Extended Vasicek method, combinations of high volatility with low speed of mean reversion may cause unreasonable results.

ALM Processing

When a process generates a balance value larger than 999,999,999,999.99, the database cannot support the value and a numeric overflow occurs.

This may be caused by large buckets in combination with large balance amounts per product leaf and short terms. For example, a large balance that matures monthly in a multi-year modeling bucket may overflow the results of the total runoff timing financial element.

Formula Results

Dimension members larger than 10 digits are restricted from input into a function that has leaf member as a parameter. To input leaves with more integers into these functions, use the decimal. For example, 1234567890123 should be entered as

1234567890.123.

Formulas within Formula Results can have a maximum of 1,024 characters.

There is a maximum of 999 coefficients within a Formula Results assumption rule.

Forecast Balance

The maximum balance that can be entered into a Forecast Balance rule is 999,999,999,999.98.

Forecast Rates

There is a maximum of ninety nine scenarios within a single Forecast Rates assumption rule.

Prepayment Model

There is a maximum of three dimensions per Prepayment Model.

Prepayment Rule

There is a maximum of 100 origination date ranges per product within a Prepayment rule. The maximum origination date that can be entered through the user interface is 12/31/2499.

User-defined Behavior Patterns

The valid range for user-defined pattern codes is 70000-99999.

User-defined Payment Patterns

The valid range for user-defined pattern codes is 1000-69999.

User-defined Repricing Patterns

The valid range for user-defined pattern codes is 500-4999.

Cash Flow Calculations

The maximum number of events that can be modeled for a single instrument within the Cash Flow Engine is 16,000.

Stochastic (Monte Carlo) Processing

There is a maximum of 2,000 rate paths.

The Value at Risk (VaR) Term has a maximum of 10 years.

Bonds with Embedded Options

Floating Rate or Adjustable Rate Instruments are not supported in Bonds with Embedded Options. In case the user gives embedded options for a floating rate instrument, embedded options would be ignored and the instrument would be treated like a normal bond.

Amortization = 700 (Non amortizing or bullet bond) is the only amortization type supported.

New business through Product Characteristics is not supported for bonds with embedded options.

Oracle Funds Transfer Pricing

Prepayment Rule

There is a maximum of 100 origination date ranges per product within a Prepayment rule.

The maximum origination date that can be entered through the user interface is 12/31/2499.

Prepayment Model

There is a maximum of three dimensions per Prepayment Model.

User-defined Behavior Patterns

The valid range for user-defined pattern codes is 70000-99999.

User-defined Payment Patterns

The valid range for user-defined pattern codes is 1000-69999.

User-defined Repricing Patterns

The valid range for user-defined pattern codes is 500-4999.

Cash Flow Generation

The maximum number of events that can be modeled for a single instrument within the Cash Flow Engine is 16,000.

Transfer Pricing Calculations

There is a maximum size of transfer rate and matched spread of 9999.9999. If a transfer rate or matched spread larger than this value is inserted into an instrument table, an error, "value too large for column" or "numeric overflow," occurs.

Database Configuration

The following limitations are applicable to the configuration of the OFSAA database.

Users and Groups

User names have a maximum of 16 characters.

OFSAA Assumption Rule Names

The maximum number of characters in a short description name is 15, and in a long description name is 60.

Number of assumption rules

The maximum System ID number in the system is 9,999,999,999. The maximum number of assumption rules for each rule type is 16,000.

SQL Statements in All Processes

SQL statements within processes are limited to a maximum length of 65,535 characters.

Instrument Table - ID Numbers

ID numbers can have a maximum length of 25 digits.

User Interface Display of Dates

All dates displayed in user interfaces have an input and display range of 01/01/1901 to 12/31/2499.

Dimension Leaf Member Set Up

Dimension Leaf values can have a maximum of 14 digits.

Only 26 key (processing) dimensions are allowed in the database. Examples of seeded key leaf types are Common COA ID, Organizational Unit ID, GL Account ID, Product ID.

The maximum number of columns that the Oracle database allows in a unique index is

32. This is the overriding constraint. After subtracting IDENTITY_CODE, YEAR_S, ACCUM_TYPE_CD, CONSOLIDATION_CD, and ISO_CURRENCY_CD, this leaves 27 columns available for Key Processing Dimensions (leaf dimensions). BALANCE_TYPE_CD is now part of the unique index so this brings the maximum number of leaf columns down to 26.

Balances

Balances stored in Instrument tables are limited to 999,999,999,999.99. Balances stored in the LEDGER_STAT table are limited to 99,999,999,999.9999.

The maximum precision for a balance used in a calculation process is 15 digits, with the range of 1.7e-308 to 1.7e+308. Calculation precision on larger numbers is compromised.

Rates

By default, rates stored in instrument tables are limited to 999.9999 and -999.9999. More precision can be achieved by increasing the number of decimals in the column. Internally, rates are stored with the same precision as balances.

Terms and Frequencies

The maximum size of a term used within any process, including Oracle ALM and Oracle FTP is 32,767, although interface limits are imposed to restrict this number to 999 when entered through a user interface. Terms affected by this restriction include payment frequency, reprice frequency, original term, amortization term, negative amortization frequency and payment change frequency.

3

Consolidation of Results

Users can combine the result sets from multiple processing runs if needed for reporting. This chapter contains the following sections, which describe how to merge results from multiple Oracle Asset Liability Management processes:

This chapter covers the following topics:

- Check the Process Unique Identifier
- Create a new ALM Process rule for your Consolidated Results
- Two Approaches to Consolidating Results
- Columns in the Result Detail Table
- Result_Master Table (Market Value Results)

Check the Process Unique Identifier

Confirm that the results were generated with *identical* assumptions for the following:

- Modeling Date Buckets
- Dynamic Gap Date Buckets and Start Dates
- Gap Profiles (for example, were gap profiles generated on all processing runs?)
- Market Value Start Dates
- Market Valuation (for example, were market values performed on all processing runs?)
- Multiple Modeling Rates (for example, Model with Transfer rates option enabled)
- Auto-Balancing (for example, was auto-balancing performed in each processing run?)

- All assumption rules are the same (Optional: see Pre-Populate the Supporting Result Files, page 3-2)

If any of these assumptions are not consistent (with the possible exception of Assumption rules), your consolidated results are not valid.

Create a new ALM Process rule for your Consolidated Results

To report off your consolidated results, you must create an Asset Liability Management (ALM) Process rule for your consolidated results. Remember that you cannot consolidate multiple ALM process results through the ALM Process interface. This step merely enables you to report off your consolidated results using the Oracle Financial Services Asset Liability Management Analytics (ALM BI) (or a 3rd party) reporting interface.

Pre-Populate the Supporting Result Files

There are four tables that get updated with the results of an ALM processing run. Two of them do not hold actual results, but rather the supporting information necessary to report off a processing run:

- **FSI_Result_Bucket**: The FSI_Result_Bucket table is updated with all of the date bucket header information (that is start and end dates for all date buckets). This table is accessed in all date related reports.

- **FSI_Result_Header**: The FSI_Result_Header table is updated with the names of all assumption rules that make up the processing run. The rule names can also be accessed when generating reports.

To populate these tables for your Consolidated Results - ALM Processing rule, you can set up an ALM process with a single record and run it. The result is that both the FSI_Result_Bucket and FSI_Result_Header tables are populated, as well as the results in FSI_O_Result_Master and Res_ Dtl_<Sys_ID_Num>. The results in FSI_O_Result_Master and Res_Dtl_<Sys_ID_Num> are not accurate and must be replaced in Step Consolidate Results Data.

Consolidate Results Data

Deterministic Results in Oracle ALM are stored in four tables:

- **Detail Result Tables**: RES_DTL_<Sys_ID_Num> and Cons_Dtl_<Sys_ID_ Num>. This contains cash flow and gap results for an individual ALM deterministic processing rule.

- **FSI_O_RESULT_MASTER** and **FSI_O_CONSOLIDATED_MASTER**: This contains market value results for all ALM deterministic processing rules.

Detail Results Tables (Cash Flow and Gap Results)

All ALM processing runs have an associated table for the storage of detailed results. A few key characteristics of these tables which are important to understand when consolidating results:

- There are 240 columns used to store forecast information BUCKET_001... BUCKET_240.
- The meaning in any of the BUCKET columns is determined by the dimension columns (listed as follows). In other words, the same column is used to store any type of financial information - you must interpret the key columns to determine what the value represents.
- All information can be consolidated by simply summing up the Bucket columns. The numbers are stored so that for weighted average purposes you need only to sum up the values and group by the relevant key columns.

Two Approaches to Consolidating Results

There are two approaches you can take to consolidating results:

- Physically consolidate the data into the existing Result Detail table. If you want to physically update the Res_Dtl_<Sys_ID_Num> table, remember to delete all of the data that exists in the table (if any) from Step Pre-Populate the Supporting Result Files, page 3-2.
- Drop the existing Result Detail table and replace this with a database view.

An alternative to physically inserting the consolidated data into the Res_Dtl_<Sys_ID_Num> table is to drop the physical Res_Dtl_<Sys_ID_Num> table and create a Database View with the identical name. Advantages to this approach are: a view saves disk space, and always reflects the latest underlying results. If you want to store the consolidated results, but not the underlying detail, however, you must store the consolidated results in a physical table.

Sample SQL: View Approach

```
CREATE VIEW RES_DTL_101 AS SELECT * FROM RES_DTL_1 UNION SELECT * FROM
RES_DTL_2
```

This approach merely appends the results from both underlying results set. This is acceptable because all reports sum records with the same key values together.

> **Note:** If the Database Administrators create a view, you must grant access to that view to the appropriate users.

An easy way to consolidate the results is to append the records from both underlying results set into the consolidated results set. This does not violate the unique key because the Result_Sys_Id from both underlying results are different. The reporting engine sums all records with the same key values together.

Columns in the Result Detail Table

There are three categories of columns in the Result Detail table:

- **Dimension Columns**: Always include these columns in your group-by statement.
- **Result Columns**: These columns should always be summed.
- **Other Columns**: These columns are irrelevant to the results, but must be included in the consolidation table or view for reporting to function properly.

If you want to use a more efficient method for physically consolidating data, create a SQL statement that groups by the Dimension Columns, sums the Result Columns, and Defaults the Other Columns. This approach saves space over the simple approach of appending the results from all detailed ALM Processing rules.

Result Detail: Dimension Columns (Group-By)

Column	Description
ALM Product Dimension	Your ALM product (or Chart of Accounts) dimension as specified in Application Preferences
Common Chart of Account dimension	Your Common Chart of Account dimension
Organization Unit dimension	If results are consolidated to the entity level, this may be irrelevant
ISO Currency Code	Currency in which results are held
Financial Element	The dimension that gives meaning to the row of data in the table - for example, are the numbers in bucket_001-bucket _240 referring to ending balance, ending rate, average balance, and so on.
Scenario Number	Results are stored by scenario number
Start Date Index	Start Date 0 = Cash Flow Information and > = 1 are used for Gap results, Market Value results, and Dynamic Start Dates

Column	Description
Result Type Code	Code describing the source of the record, current position, new business, or formula results. This column may be omitted in the group by statement if you do not want to maintain this distinction in your consolidated results.

Result Detail: Result Columns (SUM)

Bucket_001 ….. Bucket_240

Result Detail: Other Columns (DEFAULT - But do not omit)

Column	Description
All other Dimension Columns	These columns are not relevant to your simulation

Result_Master Table (Market Value Results)

The market value results in the result_master table must be manually consolidated through a SQL statement. As with the Detail Results table, the columns in the Result_Master Table can be grouped into the same three categories:

Result Master: Dimension Columns (Group By)

Column	Description
ALM Product Dimension	Your ALM product (or Chart of Accounts) dimension as specified in Application Preferences
Common Chart of Account dimension	Your Common Chart of Account dimension
Organization Unit dimension	If results are consolidated to the entity level, this may be irrelevant

Column	Description
Result Sys ID	The unique identifier for process results. This identifier maps to a specific ALM Process rule
ISO Currency Code	Currency in which results are held
Scenario Number	Results are stored by scenario number
Start Date Index	Start Date 0 = Cash Flow Information and > = 1 are used for Gap results, Market Value results, and Dynamic Start Dates
Financial Rollup	A code indicating the financial account type of the line item
Leg Type	Indicates the leg type of the instrument
Result Type Code	Code describing the source of the record, current position, new business, or formula results. This column may be omitted in the group by statement if you do not want to maintain this distinction in your consolidated results.

Result Master: Result Columns (SUM):

AVERAGE_LIFE
CONVEXITY
CUR_DEFER_BAL_C
CUR_INTR_REC_ACCR
CUR_NET_PAR_BALANCE
CUR_NET_RATE_W
CUR_PAR_BAL
CUR_TRANSFER_RATE

CUR_WARM
DURATION
LEG_TYPE
MARKET_VALUE
MARKET_VALUE_CLEAN
MODIFIED_DURATION
NEW_GROSS_BALANCE
NEW_NET_BALANCE
YTM

Result Master: Other Columns (DEFAULT - But do not omit):

Column	Description
Result_Sys_ID	This is a key column for associating the results with your ALM Process. Plug this with the value from the System ID of your consolidated processing rule.
All other Dimension Columns	These columns are not relevant to your simulation.

Sample SQL:

The following statement was used to consolidate the results of sys_id_num = 1 and 2 into a sys_id_num = 101:

```
INSERT INTO FSI_O_RESULT_MASTER
SELECT 101 AS RESULT_SYS_ID,
    SCENARIO_NUM,
  ORG_UNIT_ID,
  0 AS GL_ACCOUNT_ID,
      COMMON_COA_ID,
      PRODUCT_ID,
     FINANCIAL_ROLLUP,
      START_DATE_INDEX,
  LEG_TYPE,
  SUM(CUR_PAR_BAL),
  SUM(CUR_NET_PAR_BAL),
  SUM(CUR_DEFER_BAL_C),
  SUM(CUR_NET_RATE_W),
  SUM(CUR_WARM),
  SUM(CUR_INTR_REC_ACCR),
  SUM(NEW_GROSS_BALANCE),
  SUM(NEW_NET_BALANCE),
  SUM(MARKET_VALUE),
  SUM(DURATION),
  SUM(CONVEXITY),
  SUM(CUR_TRANSFER_RATE),
  SUM(MARKET_VALUE_CLEAN),
  SUM(MODIFIED_DURATION),
  SUM(YTM),
  SUM(AVERAGE_LIFE)
FROM FSI_O_RESULT_MASTER
WHERE RESULT_SYS_ID IN (1,2)
GROUP BY
  SCENARIO_NUM,
  ORG_UNIT_ID,
  COMMON_COA_ID,
  PRODUCT_ID,
  FINANCIAL_ROLLUP,
  START_DATE_INDEX,
  RESULT_TYPE_CD,
  LEG_TYPE
```

4

Detail Cash Flow Audit Options

This chapter describes the Detail Cash Flow Audit Options, which enable you to view daily cash flow results.

This chapter covers the following topics:

- Overview of Detail Cash Flow Audit Options
- FSI_O_PROCESS_CASH_FLOWS Table
- The Detail Cash Flow Process
- Financial Elements
- Cash Flow Codes
- Data Verification

Overview of Detail Cash Flow Audit Options

By selecting the Detail Cash Flow option in the Asset Liability Management (ALM) Process rule (the Audit block in Oracle ALM and Oracle Funds Transfer Pricing (FTP), you can view the daily cash flow results. Selecting this option writes out all cash flow and repricing events that occur for processed records. The number of records written is determined within the Audit block of both ALM and FTP processes.

> **Note:** If you are running with multiprocessing enabled, you may get fewer records, for example, FSI_PROCESS_ID_STEP_RUN_OPT.NUM_OF_PROCESSES >1. In this case, only the first process will write detail cash flows.

FSI_O_PROCESS_CASH_FLOWS Table

The relevant financial elements for each instrument record and the cash flow results are stored in the FSI_O_PROCESS_CASH_FLOWS table.

Column	Description
Result Sys ID	Key column that corresponds to the Sys ID Num of the Process rule. All cash flow results for the processing run are stored with the same Result Sys ID value. See Determining the value of the Result Sys ID, page 4-3.
Calc Source Cd	The type of cash flow processing (selected in the Process Rule):
	0 - ALM - Regular (scenario based processing)
	1 - ALM - Stochastic processing
	2 - TP - Regular
	3 - TP - Forward Rates
	4 - TP - Stochastic processing
Record Sequence	The processing order of the records.
Cash Flow Sequence	The order of the events (cash flows, repricing events).
Scenario Num	The scenario number assigned in the Forecast Rates rule for scenario-based processing or the rate path number determined in stochastic processing.
Financial elem ID	The numeric code describing a piece of financial information described in the row of result data.
ID Number	The unique record identifier for the instrument records processed.
Cash Flow Date	The date of the event.
Cash Flow Cd	The code identifying the type of event.
Float Value	The value assigned to each financial element.
Product Leaf Node	The product leaf number.
Org Leaf Node	The organizational unit leaf member, an optional output dimension in Oracle ALM and always populated in Oracle FTP output.

Column	Description
Iso Currency Cd	Currency associated with the cash flow values.

The Detail Cash Flow Process

After processing cash flows from either Oracle ALM or Oracle FTP, do the following to view the audit results:

- Determine the value of Result Sys ID
- View data by:
  - Navigating to Oracle ALM Business Intelligence and viewing results in the Detail Cash Flow Audit Report

Or you can use SQL Developer or a similar SQL query tool. For example, you can write the following SQL statement to view cash flow results:

```
SELECT * FROM FSI_O_PROCESS_CASH_FLOWS WHERE RESULT_SYS_ID = <your SysID
number> ORDER BY SCENARIO_NUM, RECORD_SEQUENCE, CASH_FLOW_DATE,
FINANCIAL_ELEM_ID;
```

Determining the Value of Result Sys ID

1. Navigate to the Process type that you want to query.
2. Select the folder and Process name.
3. Edit or View the rule.
4. At the bottom of the page, there is an **Audit Section** which contains the Sys ID Number.
5. Alternatively, on the summary page for the selected process type, you can mouseover the Process Name and the Sys ID and Description will appear.

Financial Elements

The Financial Element ID column lists the financial elements written for each payment and repricing event processed by the cash flow engine. An initial set of data is also written, recording the balance and rate as of the last payment date. The base set of financial elements written during a cash flow audit process may include some or all of the following:

Financial Element	Description
Payment Events	
100	Ending Par Balance. Final balance on payment date, after the payment has occurred.
430	Interest cash flow.
210	Total Principal Runoff, including scheduled payments, prepayments and balloon payments.
60	Beginning Par Balance. Starting balance and payment date, prior to payment.
120	Runoff Net Rate. Rate at time of payment, weighted by ending balance. To view actual rate, divide financial element 120 by financial element 100.
490	Discount rate used to determine the present value of cash flow on the payment date.
Repricing Events	
250	Par Balance at the time of repricing.
280	Before Reprice Net Rate. Rate Prior to repricing, weighted by reprice balance. To determine true rate, divide this financial element value by financial element 250.
290	After Reprice Net Rate. Newly assigned net rate after repricing occurs, weighted by reprice balance. To determine true rate, divide this financial element value by financial element 250.
Initial Event	
100	Initial par balance at start of processing.
120	Initial net rate at start of processing.

In addition to these financial elements, other data may be output, depending on the type of processing and the optional financial elements selected. For more information on financial elements, see OFSAA Financial Elements, page 17-1.

Cash Flow Codes

The Cash Flow Code column lists a code for each row that describes the event modeled by the cash flow engine:

Cash Flow Code	Description
1	Initial recording of balances and rates
2	Payment event only
20	Reprice event only (not during tease period)
8	Reprice during tease period
22	Reprice and payment event together (not during tease period)
14	Reprice and payment during tease period

Data Verification

You can copy the results from the Process Cash Flows table (FSI_O_PROCESS_CASH_FLOWS) and paste them into a spreadsheet to facilitate analysis against validated data. If the cash flows do not behave as expected, examine instrument table data or your assumptions. For more information, see Cash Flow Calculations, page 5-1.

5

Cash Flow Calculations

Oracle Asset Liability Management uses a cash flow engine to ensure modeling consistency across the Oracle Financial Services Analytical Applications (OFSAA) suite of products.

This chapter describes the calculations performed by the OFSAA Cash flow engine and defines concepts that are vital in forming a complete understanding of the capabilities of the cash flow engine.

This chapter covers the following topics:

- Instrument Level Modeling
- Modeling Flexibility Defined by Instrument Data
- Daily Cash Flows
- Event Driven Logic
- Financial Elements
- Multicurrency Accounting and Consolidation
- Terminology Used in This Chapter
- Cash Flow Calculation Process
- Additional Processing Events
- Accounting for Exchange Rate Fluctuations
- Market Value Calculation
- Consolidation of Results
- Currency-Based Gap Modeling
- Detail Cash Flow Data
- Rule of 78's Example

Instrument Level Modeling

Several processes within the OFSAA applications require cash flows in order to produce results, including:

- Transfer Pricing
- Market Valuation
- Income Simulation

Oracle generates cash flows at the individual instrument level. Each individual instrument record is processed and generates a unique set of cash flows as defined by that instrument record's product characteristics. This provides an optimum level of accuracy. Cash flows from individual instrument records are used. The cash flows produced are then manipulated to produce the required results.

Modeling Flexibility Defined by Instrument Data

Specific cash flow characteristics defined in an instrument record determine the cash flow results for each instrument. Each instrument record has:

- Payment information (dates, frequencies, amounts)
- Balance information (current balance, original balance)
- Rate information (gross rate, net rate, transfer rate)
- Currency (the currency in which the record is denominated)

There are over 50 cash flow columns that drive the results. Depending on the information in these columns, you can model an unlimited number of unique instruments.

Daily Cash Flows

Cash flows can occur on any date and with any frequency thereafter. Not only does this provide accurate results, but you also have the ability to change the modeling buckets without having to worry about an impact on the underlying cash flows.

Event Driven Logic

The OFSAA Cash Flow Engine generates cash flows as a series of events. On any day and with any frequency thereafter, depending on the instrument characteristics, any of the following events can occur:

- Payment
- Re-pricing
- Payment recalculation
- Prepayment

This reference guide explains the calculations that occur for each event.

Financial Elements

On an event date, the Cash Flow Engine computes the results of that event, the financial elements. For example, on a payment event, it can compute the following:

- Interest
- Principal runoff
- Total cash flow
- Ending balance

The Oracle Financial Services Cash Flow Engine generates over 50 financial elements that can be used in analysis.

Multicurrency Accounting and Consolidation

Currency fluctuation is a critical risk affecting multicurrency balance sheets. Oracle ALM enables users to model instruments denominated in different currencies, computes the effect of currency fluctuations on earnings, and performs cross-currency consolidations.

Terminology Used in This Chapter

Non-Currency-Based versus Currency-Based Processing

Users may select one of four functional dimensions in their Asset Liability Management (ALM) Deterministic Process. These dimensions describe the way the output is aggregated and the selection impacts the approach to translation of multicurrency data. In this chapter, we refer to processing as being either **non-currency-based** or **currency-based**, depending on which functional dimension is selected:

Currency Processing Type	Functional Dimension in ALM Process
Non-Currency Based	Product
	Product / Organization Unit
Currency Based	Product / Currency
	Product / Organization Unit / Currency

Local Currency versus Reporting Currency

When processing multicurrency data, we generally refer to the currency as one of the following:

- Local Currency

  This can be any active currency. It is the currency in which the record is denominated. For example, USD (US Dollars) or JPY (Japanese Yen).

- Reporting Currency

  This is the target for currency translation. That is, the cash flow engine translates cash flows or values on the instrument record by using the exchange rate from the local currency to the reporting currency.

Default Currency

Oracle ALM enables users to define modeling assumptions (for example, market value Discount Rates, new business Pricing Margins, Maturity Strategies, and so on) for each Product/Currency combination. Generally, the cash flow engine retrieves currency-based assumptions based on Product + Currency. If no such assumptions exist, it uses the assumptions the user has defined for the Product + Default Currency. If it still cannot find a match, it applies a default or may write out an error message.

[m] Forecast (in memory) versus [r] Actual (on record)

As a cash flow instrument is modeled through time, the data associated with this instrument changes due to payments, repricings, or other circumstances. These changes apply only in memory and do not affect the information stored in the instrument tables.

A subscript notation of [m] for memory and [r] for detail record differentiates between the forecasted data and the actual data. For example, current payment refers to the current payment stored in the instrument table. Current payment_m refers to the current payment in memory that is updated each time a payment recalculation occurs.

Cash Flow Calculation Process

The following steps summarize the cash flow calculation process:

1. Initialize modeling data and parameters for instrument to be modeled.

2. Process modeling event(s) until current date equals the maturity date or the modeling end date, or the current balance equals zero:

    - Calculate changes to underlying instrument
    - Calculate financial elements associated with event
    - Increment forward event dates

3. Generate secondary results:

    - Deferred amortization recognition
    - Market values (Clean Price, Market Value)
    - Duration (Macaulay Duration, Modified Duration, Convexity)
    - Gap (Repricing and Liquidity)
    - Average Life
    - Yield to Maturity

4. Accumulate daily information into appropriate time buckets.

5. Optionally translate and consolidate values in each time bucket for Oracle ALM currency-based processes.

Initialization of Data

The first step in the process is to gather the information necessary to model the current instrument. This information is available from several sources, including:

- The instrument table
- Payment Schedule table
- The As of Date and active Time Bucket rule
- Payment Pattern interface

- Repricing Pattern interface
- The assumption rules specified in the ALM Process
- The historical exchange rates table

Determine Account Type of Instrument

Account types classify instruments by their use in financial statements and determine how the cash flow engine processes an individual instrument. The Common COA ID value determines the account type of an individual record. The Dimension Member interface classifies each Common COA ID member as an account type. All other Product type dimensions contain a Common COA ID member mapping as an attribute and through the Common COA ID association, the account type is known. Based on these account types, there are five categories of cash flow processing, as shown in the following table.

Cash Flow Category	Process Description - Detail	Process Description - New Business	Associated Account Type
Detail cash flow	Process daily cash flow events and generate necessary financial elements	Process daily cash flow events and generate necessary financial elements	Interest-Earning Asset Interest-Bearing Liability Off Balance Sheet Receivable Off Balance Sheet Payable
Balance Only	Process the record originating on the origination date and running off on the maturity date. No payments are processed.	Show ending balances equal to the forecasted amounts in each bucket.	Other Asset Other Liability

Cash Flow Category	Process Description - Detail	Process Description - New Business	Associated Account Type
Interest Only	Process the instrument as a single interest payment covering the time from the origination date to the maturity date. Recognize the current balance as an interest cash flow on the maturity date, but accrue interest from the origination date to the maturity date.	Show interest cash flows / accruals equal to forecasted amounts in each bucket.	Interest Income Interest Expense
Non-Interest Income / Expense	Process the instrument as a single non-interest payment covering the time from the origination date to the maturity date. Recognize the current balance as a non-interest cash flow on the maturity date.	Show non-interest financial elements equal to forecasted amounts in each bucket.	Non-Interest Income Non-Interest Expense
Special Autobalancing	Detail information is used only to update the current position.	Generate results as needed during the autobalancing process if account is specified as autobalancing account.	Taxes Dividends Equity

Cash Flow Category	Process Description - Detail	Process Description - New Business	Associated Account Type
Off Balance Sheet Account Type	Instruments with the Off Balance Sheet account type should be located in one of the derivative instrument tables, e.g. swaps, futures. Instruments with the Off Balance Sheet account type should not be located in non-derivative instrument tables. Instruments with the " Off Balance Sheet" account type should have the correct leg type populated, for example, it should be "1 " (payable leg) or "2" (receivable leg). It should not be "0".	.	Off Balance Sheet Receivable Off Balance Sheet Payable

Note: For **Other Assets**, engine does not refer the maturity mix or new business timing. Origination for such accounts always occur at the beginning of the bucket and mature occurs at bucket end + 1 . **Other Assets**, **Other Liabilities**, and **Equity** account types will have the same behavior.

For **Non interest Income/Expense** and **Interest Income/Expense** accounts, origination happens on bucket start date -1 and matures at bucket end date.

For more information about financial element output by account type, see the Financial Element Calculations table, page 17-65.

Initialize Interface Data

Data retrieved from the user interface impacts how an instrument is processed. The following pieces of user interface data affect processing:

- Product Characteristics > Model with Gross Rates

- Product Characteristics > Interest Credited
- Product Characteristics > Percent Taxable
- Product Characteristics > Currency Gain/Loss Basis
- Product Characteristics > Pay-Equivalent Compounding Convention
- Behavior Patterns
- Payment Patterns
- Repricing Patterns

Model with Gross Rates

The Model with Gross Rates option is available in the Oracle ALM > Product Characteristics rule, and in the Oracle Funds Transfer Pricing > Transfer Pricing Rule.

Interest Credited

The interest credited option resides in the ALM > Product Characteristics Rule. The switch can be enabled for any leaf dimension. However, it affects only the cash flows of Simple/Non-Amortizing instruments.

When the switch is enabled for a non-amortizing instrument, interest cash flows are added to the principal balance at each payment prior to maturity. On the maturity date, the initial principal balance plus the accumulated interest cash flows are reflected as principal runoff at maturity. When the switch is enabled for amortizing instruments, it is ignored by the cash flow engine.

Percent Taxable

The Oracle ALM auto-balancing option requires the Percent Taxable input to determine the percentage of total income/expense that is subject to tax. This can be set up in the ALM > Product Characteristics rule, and should be defined for each Product/Reporting currency (or Product/Default Currency 000) combination.

Currency Gain/Loss Basis

Oracle ALM enables users to select the currency accounting method for each Product/Currency combination modeled in multicurrency simulations. This is referred to as the Currency Gain/Loss Basis; it is an option in the ALM > Product Characteristics rule.

Pay-Equivalent Compounding Convention

This switch is in the Oracle ALM Product Characteristics rule. When the cash flow engine calculates interest, this switch indicates when it needs to convert rates from the quoted compounding basis to the payment basis for the record.

Behavior Patterns

Behavior pattern data is retrieved when the cash flow engine must process an instrument with an amortization code in the behavior pattern range from 70000 to 99999. The amortization code from the detail instrument record is matched to the set of behavior pattern data with the same code. If no match is found to the amortization code from the detail record, the record is processed as a non-amortizing instrument.

Payment Patterns

Payment pattern data is retrieved when the cash flow engine must process an instrument with an amortization code in the payment pattern range from 1000 to 69999. The amortization code from the detail instrument record is matched to the set of pattern data with the same code. If no match is found to the amortization code from the detail record, the record is processed as a non-amortizing instrument.

Repricing Patterns

Repricing pattern data is retrieved when the cash flow engine must process an instrument with an adjustable type code in the repricing pattern range from 500 to 4999. If no match is made, the engine defaults to the record characteristics in the repricing frequency column and processes the instrument as a standard adjustable instrument.

When a match is made, the instrument is modeled based on the repricing pattern. The cash flow engine first evaluates what the status of the instrument is as of the starting date of the cash flow process. The current repricing date is determined by rolling forward from the origination date to the next repricing date that follows the process start date. If that date does not correspond to the next repricing date, the repricing date from the record is used.

A repricing event is triggered when the period of time between events has elapsed. When this occurs, the defined rates are assigned to the detail record of the instrument. If the repricing type is Flat Rate, the rate from the event detail of the repricing pattern is applied to the detail record of the instrument. If the repricing type is Indexed Rate, a rate lookup is triggered for the customer rate and the transfer pricing rate. If the interest rate code (IRC) is a yield curve, the point on the yield curve used is the repricing term associated with the current repricing information, unless the IRC term has been specified in the repricing pattern event. This rate, plus the specified margin, is the new fully indexed rate. Rate caps and floors are applied after this calculation occurs.

Initialize Cash Flow Data

The cash flow engine gathers cash flow data for the instrument to be processed, representing a subset of the information stored in the instrument tables for this record. The Detail Cash Flow Data Table lists the columns referenced in this process.

Cash flow data provides current information about the characteristics of a cash flow instrument. This information must be consistent to ensure consistent output. Prior to processing cash flows, the Cash Flow Edit Process should be run to avoid producing unreasonable results.

Cash flow data can be classified into categories defining its use during the processing of an event:

- Static characteristics
- Dynamic characteristics
- Triggers

Static Characteristics

Static characteristics provide information to the cash flow engine about how the instrument should be modeled. For non-pattern and non-schedule instruments, all of the following characteristics remain constant during the modeling process.

- **Event frequencies**
  - Repricing
  - Payment
  - Payment change
- **Financial Code Values**
  - Accrual basis code
  - Amortization type code
  - Rate change round code
- **Leaf values**
  - Product leaf
  - Common COA ID
- **Currency Code**
- **Repricing parameters**
  - Margin
  - Rate caps

- Rate increase period
- Rate change minimum

For pattern and schedule instruments, the payment frequency and/or repricing frequency can vary throughout the life of the instrument.

Dynamic Characteristics

Dynamic characteristics are updated each time an event occurs, as a result of what has occurred during the event. They include:

- **Balances**
  - Current
  - Current deferred
- **Rates**
  - Current net
  - Current gross
  - Current transfer
- **Event counters** (remaining number of payments)

Triggers

Triggers signal the cash flow engine, indicating it is time to model a particular event and can therefore change their value during the modeling horizon.

- Event dates
  - Next payment
  - Next reprice
  - Payment change
- Negative Amortization balance in conjunction with neg-am limit

Translation (Non-Currency-Based Processes)

If the process is non-currency-based (for example, the ALM Deterministic Process specifies a functional dimension of **Product** or **Product/Organizational Unit**), the cash flow engine compares the local currency (on the instrument) with the reporting currency (specified in the Process rule) to determine whether balance and payment amounts in the instrument need to be translated. If the instrument is already denominated in the reporting currency, no translation is necessary. If it is in a different currency, the process must translate all amounts prior to processing cash flows. It translates those values from the local currency to the reporting currency by dividing by the exchange rate in effect at the As-of-Date.

> **Note:** In currency-based processes (that is, where the ALM Deterministic Process rule specifies a functional dimension of **Product/Currency** or **Product/Organizational Unit/Currency**), no translation is required because the instrument is processed in its local currency and all results are accumulated in the local currency.

The following is a list of all instrument values requiring translation for non-currency-based processing:

- Original Par Balance
- Current Par Balance (from instrument table or Transaction Strategy)
- Original Payment Amount
- Current Payment Amount
- Negative Amortization Amount
- Current Deferred Balance (from instrument table or Transaction Strategy)

Initializing Schedule Records

Processing of scheduled amortization instruments requires gathering additional data from the FSI_D_PAYMENT_SCHEDULE table. An amortization code of 800, 801, or 802 signals to the cash flow engine that this instrument record is a schedule record. To properly model schedule instruments, the cash flow engine must retrieve payment dates and payment amounts from the Schedule table. A match is made between the Identity Code, ID number, and Instrument type code from the instrument record to the same data in the Schedule table. If no match is found, the instrument is processed as a non-amortizing record.

Amortization Code 800

The different schedule amortization codes relay to the cash flow engine what type of payment is stored in the schedule table. An amortization code of **800** signifies that payment amounts are principal and interest amounts. On payment dates, these payments are processed as conventional amortization payments.

Amortization Code 801

An amortization code of **801** is used for level principal payment schedules. On the schedules for these instruments, the payment amount represents the principal portion of the payment only. For 800 and 801 schedules, the payment amounts should be expressed gross of any participations.

Amortization Code 802

Instruments with amortization code of **802** do not reference the payment amount column in the schedule tables. These instruments are processed as interest only records, with all principal, with the exception of prepayments, running off on the maturity date.

The data in the maturity date, next payment date, last payment date, remaining number of payments, and current payment columns from the instrument record should coincide with the same information in the schedule table. When this information is inconsistent, the information in the detail record supersedes the data in the schedule table.

In this case, the payment on the next payment date occurs on the date defined in the next payment date column of the instrument record, for the amount defined in the current payment column of the instrument record.

All payments after this date and prior to the maturity date are made according to the payment date in the schedule table. On the maturity date, the date from the maturity date column of the instrument record is used to pay off the remaining balance of the instrument record. If payment dates exist in the schedule beyond this date, they are ignored.

Translation (Non-Currency-Based Processes)

Similar to the considerations discussed earlier under **Initializing Instrument Records**, if the process is non-currency-based, the scheduled payment amount must be translated from the local currency to the reporting currency. The cash flow engine uses the exchange rate in effect at the As of Date, based on the local currency stated in the instrument record associated with this schedule record.

Initializing Pattern Records

The following logic applies to both User-Defined Payment Patterns and User-Defined Repricing Patterns. Applicability to repricing is indicated in parenthesis.

Single Timeline Patterns

To initialize an instrument record whose payment (or repricing) characteristics are defined by a single timeline pattern, the cash flow engine must synchronize the detail instrument record with the payment (or repricing) pattern. Synchronization determines the current payment of the instrument within the payment (or repricing) pattern.

The synchronization process depends on whether the pattern is relative or absolute. To synchronize a relative pattern, the cash flow engine calculates the payment (or repricing) dates for the instrument record by rolling the origination date forward by the pattern frequencies. Once it calculates a payment (or repricing) date greater than the as-of-date, it stops. The number of times it was necessary to roll the date forward determines the current payment (or repricing event) number for the record.

Example

An instrument record processed on an as-of-date of 03/31/2011 with an origination date of 01/01/2011, and a next payment (or repricing) date of 05/15/2011 is matched to the following pattern:

ROW #	Frequency	Repetitions
1	1M	3
2	3M	3

The origination date is rolled forward in the following manner:

- Starting point -- 01/01/2011
  - Add first monthly payment (or repricing) frequency --02/01/2011
  - Add second monthly payment (or repricing) frequency -- 03/01/2011
  - Add third monthly payment (or repricing) frequency -- 04/01/2011

  After the third roll forward, the payment (or repricing) date is greater than the as-of-date. The cash flow engine interprets that the record is on its third payment (or repricing), which is the final monthly payment (or repricing). It models this payment (or repricing) on the next payment (or repricing) date from the detail record, in this case, 05/15/2011. The next payment (or repricing) is scheduled for 8/15/2011, using the three month frequency from the fourth payment (or repricing) in the schedule.

  Absolute patterns do not require the same rolling mechanism for synchronization. The next payment (or repricing) date from an absolute pattern is determined by the first month and day after the as-of-date. If this date does not correspond to the next payment (or repricing) date from the detail record, the next payment (or repricing) date of the detail record supersedes the date of the pattern. From that point on in the process, the payment (or repricing) dates from the pattern are used.

  The cash flow engine has been designed in this manner to provide greater flexibility in modeling payment and repricing patterns. However, this flexibility increases the importance of detail data accuracy to ensure that when discrepancies exist between detail data and patterns, the differences are intended.

Multiple Timeline Patterns (Payment Patterns Only)

To initialize a detail instrument record tied to a split pattern, the cash flow engine must generate a separate record for each split. The current balance for each split record is calculated using the percentage apportioned to that split, as defined through the payment pattern interface. The original balance, original payment, and current payment columns are also apportioned according to the percent defined through the interface.

For each timeline resulting from the split of a detail instrument record, the current payment date must be determined. The method for determining the payment date is the same as described for single timeline patterns with one exception. For these instruments, the next payment date from the original instrument record does not override the calculated next payment date. The date derived from rolling the origination date forward for relative timelines or locating the next date for absolute timeliness is assumed to be the correct payment date.

Translate Values (Non-Currency-Based Processes)

If the Payment Pattern is defined using an absolute payment amount, the engine may need to translate the amount to the reporting currency specified in the Process rule. This is similar to the considerations discussed earlier under **Initializing Instrument Records** that is, if the process is non-currency-based, the Pattern's Payment Amount needs to be translated. The cash flow engine uses the exchange rate in effect at the As of Date, based on the local currency stated in the instrument or new business record associated with this pattern record.

Modeling Start and End Dates

Modeling start and end dates are determined by the type of processing (ALM or FTP) and the instrument being processed, as shown in the following table:

	Oracle Transfer Pricing - Adjustable	Oracle Transfer Pricing - Fixed	Oracle Transfer Pricing - Adjustable Rem Term	Oracle Transfer Pricing - Fixed Remaining Term	Oracle Asset Liability Management
start date	last reprice date	origination date	As of Date + 1 day (App Pref's)	As of Date + 1 day (App Pref's)	As of Date + 1 day (App Pref's)
end date	next reprice date	maturity date	next reprice date	maturity date	last day of last modeling bucket (Time Bucket rule)

Only records that have a value in the As of Date column of the database equal to the as of date in your Application Preferences are processed.

Additionally Derived Data

Initialization of Adjustable Rate Instruments for Transfer Pricing

For transfer pricing of adjustable rate instruments, data is reset to values consistent with the last reprice date. The next payment date is rolled back by the payment frequency to the first payment date after the last reprice date. The remaining number of payments is increased by the number of payments added in the rollback process.

The field Last Reprice Date Balance is used in place of the current balance in a transfer pricing process. If the balance as of the last reprice date is not available, update this column with the current balance. The transfer pricing program has a special feature that re-amortizes the current payment if the following conditions are met:

- The last reprice date balance equals the current balance.
- Payments occur between the last reprice date and the as-of-date.
- The instrument is not tied to an amortization pattern or an amortization schedule.

These three conditions signal to the transfer pricing engine that the balance as of the last reprice date was not available and the current balance should be used as a proxy.

Percent Sold Adjustment

Balances must be adjusted for participations:

- Current net balance = current par balance * (100 - percent sold)
- Current payment net = current payment * (100 - percent sold)
- Original net balance = original par balance * (100 - percent sold)
- Last reprice balance net = last reprice balance * (100 - percent sold)
- Original payment net = original payment * (100 - percent sold)

Forecast Balance assumptions

- Non-currency-based processing

  If the ALM Process rule includes Forecast Balance new business, only the assumptions for the reporting currency are applied. For associated new business assumptions in the Pricing Margin rule, Maturity Mix rule, and Product Characteristics rule and if no assumptions exist for the reporting currency, the cash flow engine uses assumptions defined for the Default Currency (Code **000**). If it still cannot find a match, it employs a default or may write out an error message.

- Currency-based processing

  In currency-based processes, results are produced in their local currency. Therefore, Oracle ALM reads all Forecast Balance assumptions regardless of their associated currency.

Process Modeling Events

There are four events modeled in the cash flow engine:

- Payment
- Payment change
- Reprice
- Prepayment

When multiple events occur on the same day the order of processing is as follows:

Interest in Arrears

1. Payment calculation
2. Payment
3. Prepayment
4. Reprice

Interest in Advance

1. Reprice
2. Payment
3. Prepayment

For interest in advance instruments, payment calculation is not applicable. Payment calculation occurs only on conventionally amortizing instruments. Processing of an event includes these steps:

- Dynamic information is updated
- Financial elements summarizing the event is generated
- Event dates are incremented to the next event date

Payment Calculation Event

Cash flow data characteristics are:

Static Information - Conventional Adjustable and Payment Patterns

- Current gross rate
- Current par balance
- Amortization Term and Multiplier
- Amortization type
- Adjustable Type Code
- Compounding Basis Code

Additional Information - Adjustable Neg-Am

- Payment increase cycle
- Payment increase life
- Payment decrease cycle
- Payment decrease life
- Payment change frequency and multiplier
- NGAM Equalization frequency and multiplier
- Original payment amount
- Dynamic Information
- Current payment

Event Trigger - Transfer Pricing

- Cash flow transfer pricing of an Adjustable instrument

Event Triggers -Conventional Adjustable and Conventional Payment Patterns

- Reprice Event

Event Triggers -Adjustable Neg Am

- Next payment change date
- NGAM balance > NGAM limit
- NGAM Equalization date

Payment Calculation Steps

1. Calculate new current payment.

   Conventionally Amortizing Payment =

$$\frac{\text{Current Par Balance}_m}{\frac{1}{\text{Current Rate}_C} * \left(1 - (1 + \text{Current Rate}_C)^{(-1 * \text{rem pmts}_a)}\right)}$$

where Current Rate_C = Current compounded customer rate per payment.

rem pmts_a = remaining number of payments based on amortization.

Current Par Balance_m = current balance at time of payment recalculation.

For conventional schedules that reprice, payment recalculation does not occur. For patterns that reprice, payment recalculation does not occur during the repricing event. For these instruments, the payment is calculated at the time of payment. For more details, see the Payment Event, page 5-22 section.

Current Compounded Customer Rate per Payment

The customer rate must be adjusted to a rate per payment. If no compounding occurs, the rate can be divided by the payments per year.

Example: current customer rate: 7.5%

Payment Frequency	Calculation	Rate per Payment
Monthly	7.5 ÷12	0.625
Quarterly	7.5 ÷ 4	1.875
Yearly	7.5 ÷ 1	7.5

If the instrument compounds, the rate must be adjusted for compounding. For monthly rates that compound daily, an average number of days assumption of 30.412 is used.

Remaining Number of Payments Based on Amortization

If the amortization term is equal to the original term, then the remaining number of payments is used.

If the amortization term <> original term, the remaining number of amortized payments are calculated by adding the amortization term to the origination date to determine the amortization end date. The remaining number of payments are calculated by determining how many payments can be made from and including the next payment date and this date.

The remaining number of payments is calculated for patterns based on the payment frequency at the time of repricing. As with conventional instruments, the amortization end date is used for payment recalculation. The remaining term is calculated using the difference between this date and the next payment date. This term is divided by the active payment frequency and one additional payment is added to it for the payment on the next payment date.

2. Apply periodic payment change limits.

   Periodic payment change limits restrict the amount the payment can increase over its previous value. These limits are applied only when the payment recalculation is triggered by a payment adjustment date or a negative amortization limit. Because of these limits, principal may continue to negatively amortize when the negative amortization limit has been reached.

	Increasing Payment	**Decreasing Payment**
Condition	Newly Calculated Payment > (1 + (Payment Increase Life_r / 100)) * Original Payment_r	Newly Calculated Payment < (1 + (Payment Decrease Life_r / 100)) * Original Payment_r
Adjustment if True	Current Payment_m = (1 + (Payment Increase Life_r / 100)) * Original Payment_r	Current Payment_m = (1 + (Payment Decrease Life_r / 100)) * Original Payment_r

3. **Apply lifetime payment change limits**

   Lifetime payment caps and floor set a maximum and a minimum amount for the payment. These limits are applied only when the payment recalculation is triggered by a payment adjustment date or a negative amortization limit. Because of these limits, principal may continue to negatively amortize when the negative amortization limit has been reached.

	Increasing Payment	Decreasing Payment
Condition	Newly Calculated Payment > (1 + (Payment Increase $Life_r$ / 100)) * Original $Payment_r$	Newly Calculated Payment< (1+ (Payment Decrease $Life_r$ / 100)) * Original $Payment_r$
Adjustment if True	Current $Payment_m$ = (1 + Payment Increase $Life_r$ / 100)) * Original $Payment_r$	Current $Payment_m$ = (1 + Payment Decrease $Life_r$ / 100)) * Original $Payment_r$

4. NGAM equalization event.

   If the payment recalculation is triggered by a NGAM equalization date, payment change limits do not apply. If the newly calculated payment is greater than the lifetime payment cap or less than the lifetime payment floor, the appropriate lifetime payment limit (cap/floor) is set equal to the newly calculated payment.

5. Update current payment field.

   Once all payment limits have been applied, the new current payment is updated in memory for processing of future events.

Payment Event

Cash flow data characteristics are:

- Static Information
- Dynamic Information
- Event Triggers
- Additional Assumption Information

Static Information

- Amortization type
- Current Payment
- Accrual Basis code
- Current gross rate
- Current net rate

- Current transfer rate
- Origination Date
- Payment Frequency and multiplier
- Interest Type
- Compounding Basis Code
- Last Payment Date

Dynamic Information

- Current Par Balance
- Remaining Number of Payments
- NGAM Balance

Event Triggers

- Next Payment Date
- Maturity Date

Additional Assumption Information

- Interest credited switch
- Compounding Method

Payment Event Steps

The following are descriptions of the Payment Event Steps:

1. Calculate interest cash flow(s)

   The amount of interest to be paid on a payment date is calculated as follows:

Interest Cash Flow	Calculation
Interest cash flow gross	Current net par balance x gross rate per payment

Interest cash flow net	Current net par balance x net rate per payment
Interest cash flow transfer rate	Currency net par balance x transfer rate per payment

> **Note:** Rule of 78's Exception: The Rule of 78's loans have a pre-computed interest schedule.

Rate per Payment -- Accrual Adjustment

The annual coupon rate must be adjusted to a rate per payment. The accrual basis code defines how this adjustment should be made.

Accrual Factor Codes

Example

rate per payment for a June 30 payment (annual rate = 6.0%, payment frequency = 3 months)

Accrual Basis Code	Payment Adjustment	Rate per Payment
30/360	(3*30)/360 * 6.0	1.500%
30/365	(3*30)/365 * 6.0	1.4795%
30/Actual	90/365 * 6.0	1.4795%
Actual/Actual	(30+31+30)/365 * 6.0	1.4959%
Actual/365	(30+31+30)/365 * 6.0	1.4959%
Actual/360	(30+31+30)/360 * 6.0	1.5167%

This formula assumes a single rate per payment period. If an instrument reprices multiple times within a payment period, only the last repricing event affects the interest cash flow.

Rate per Payment -- Compounding Adjustment

Compounding is applied to the rate used in interest calculation if the compounding frequency is less than the payment frequency. The compounding formula that is applied to the current rate is as follows:

Compounding Code	Calculation
Simple	No compounding is calculated
At Maturity	No compounding is calculated
Continuous	e^(rate per payment) - 1
Daily	(1 + rate/365 or 366)^ (days in payment)-1
Monthly, Quarterly, Semi-Annually, Annually	(1+ rate per pmt/cmp pds per pmt)^(cmp pds per pmt) - 1

Rate per Payment - Stub and Extended Payment Adjustment

An adjustment may be made if the expected days in the payment are different than the actual days in the payment.

The number of days between the next payment date and the last payment date is compared to the payment frequency, specified in days. The payment frequency specified in days depends on the month the payment occurs. If these numbers are not equal, the interest cash flow is adjusted by the ratio (next payment date - last payment date) /payment frequency in days.

On the last payment processed in the modeling horizon, the number of days between the maturity date and the last payment prior to the maturity date is compared to the payment frequency specified in days. The payment frequency specified in days depends on the month in which the maturity date occurs. If these numbers are not equal, the interest cash flow is adjusted by the ratio (maturity date - last payment date)/ payment frequency in days.

2. Calculate current payment for patterns and schedules

   For amortization patterns, the payment amount can be defined independently for each payment date. Therefore, on each payment, the payment amount must be calculated according to the characteristics defined for that date. Payment amounts can be driven by several different factors. These following factors are each explained:

   Percent of Original Payment

   Oracle Asset Liability Management

   On the first modeled payment on a detail instrument record, the amount in the current payment column is assumed to accurately represent the payment amount as of the next payment date. If this instrument record is partially sold, the current payment is multiplied by (100-percent sold) to get the net payment amount.

If this is the first payment made by a new business record, the payment amount is calculated using the original balance, original rate, and original number of payments. The original number of payments is calculated by using the amortization term, as specified through the Maturity Mix Rule or Transaction Strategies Rule, and the original payment frequency.

On subsequent payment dates, Oracle calculates the amount paid by multiplying the pattern percent by the amount in the original payment amount column, adjusted for percent sold, if applicable. The pattern percent is the percent of original payment specified in the user interface for that payment.

<u>Oracle Funds Transfer Pricing</u>

For standard transfer pricing, the model calculates the payment amount for each payment that falls between the last reprice date and the next reprice date (adjustable rate instruments) or between the origination date and maturity date (fixed rate instruments) by using the original payment from the instrument record, and applying the pattern percent from the user interface.

For remaining term transfer pricing, the model calculates the payment amount in the same manner as described earlier for Oracle ALM.

Percent of Current Payment

<u>Oracle Asset Liability Management and Transfer Pricing</u>

On the payment date, Oracle determines the amount to be paid by first calculating a new payment according to the **active** characteristics, including the current balance, current rate, current payment frequency, and calculated remaining number of payments. The remaining number of payments is calculated by determining the amount of time remaining in the amortization term and dividing this term by the current payment frequency.

After the payment has been calculated, the pattern percent is applied.

Percent of Current Balance

<u>Oracle Asset Liability Management</u>

Percent of Current Balance is applicable only for Level Principal payment patterns. On the first modeled payment, the amount in the current payment column is assumed to accurately represent the payment amount as of the next payment date. If the instrument is partially sold, the amount should be multiplied by (100- percent sold) to get the net payment amount.

For all subsequent payments, the payment amount should be calculated at the time of payment by multiplying the outstanding balance by the pattern percent.

<u>Oracle Funds Transfer Pricing</u>

Calculations for Transfer Pricing work similarly to ALM. However, for fixed rate instruments, modeling begins at the origination date, using the original balance. For adjustable rate instruments, modeling begins at the last reprice date, using the last reprice date balance.

Percent of Original Balance

Oracle Asset Liability Management and Transfer Pricing

Percent of Original Balance is applicable only for Level Principal payment patterns. On the first modeled payment, the amount in the current payment column is assumed to accurately represent the payment amount as of the next payment date. If the instrument is partially sold, the amount should be multiplied by (100- percent sold) to get the net payment amount.

For all subsequent payments, the payment amount should be calculated at the time of payment by multiplying the original balance, net of participations, by the pattern percent.

Absolute Value

Oracle Asset Liability Management

Absolute value is available for detail instruments; it cannot be used for new business instruments. On the first modeled payment, the amount in the current payment column is assumed to accurately represent the payment amount as of the next payment date. If the instrument is partially sold, the amount should be multiplied by (100-percent sold) to get the net payment amount.

For all subsequent payments, the absolute value amount from the pattern is used. If the instrument has a percent sold, the percent sold is applied to the absolute payment amount.

Oracle Funds Transfer Pricing

For standard transfer pricing, the absolute payment amount is used, adjusted for the participation percent.

Interest Only

Oracle Asset Liability Management and Transfer Pricing

On all interest only payments, the payment amount is calculated as the interest due on that date. No reference is made to the current payment column from the detail instrument record. Any payments in the current payment column are ignored.

3. Calculate principal runoff

   Principal Runoff is a function of the amortization type of the instrument and the current payment. The current payment on a conventionally amortizing record represents the total P&I payment, while the current payment on a level principal record represents the principal portion of the total payment.

   Simple Amortization (code = 700, 802, and any of the Non-Amortizing Pattern Codes)

   General case: Principal Runoff = 0

   Interest Credited: -1 * interest cash flow gross

Conventional Amortization (code = 100, 500, 600, 800, and any conventionally amortizing pattern codes)

Principal Runoff = current payment - interest cash flow gross

Lease Amortization (code = 840)

Principal Runoff = current payment - interest cash flow gross

Level Principal Amortization (code = 820, 801, and any other level principal amortizing pattern codes)

Principal Runoff = current payment

Annuity Amortization (code = 850)

Principal Runoff = current payment * -1

Rule of 78's (code = 710)

Principal Runoff = current payment - interest cash flow gross

4. Special negative amortization check

   If principal runoff is negative and the instrument record is adjustable neg-am; then additional checks must be made to ensure that the record is not exceeding neg-am limits. The check that is made is the following:

   -1 * principal runoff + neg am balance$_m$ > neg am limit$_t$ /100 * original balance$_r$

   If this condition is true, the payment is not made. The payment is recalculated (see payment calculation event, page 5-19). After the new payment has been calculated, the scheduled principal runoff is recalculated, based on the new payment information.

5. Maturity date case

   If the payment date is also the maturity date, then the remaining balance must be paid off. Principal At Maturity = Current Balance$_m$ - Scheduled Principal Runoff

6. Update current balance

   The current balance must be updated to reflect the principal portion of the payments and any interest credited.

   Current Balance$_m$ = Current Balance$_m$ - Principal Runoff - Principal At Maturity + Interest Credited

7. Update remaining number of payments

   After a payment has been made, the underlying information must be updated in preparation for the next event.

   The remaining number of payments is reduced by 1. If remaining number of payments is zero, the modeling for this instrument is complete.

$\text{Remaining Payments}_m = \text{Remaining Payments}_m - 1$

8. Update next payment date

   For standard amortization instruments, the next payment date is set equal to the current payment date plus the payment frequency.

   $\text{Next payment date}_m = \text{Current payment date} + \text{Payment frequency}$

   If instrument is an amortization schedule, the next payment date is determined from the dates in the schedule table.

   If the instrument is an amortization pattern, the next payment date is determined by incrementing the current payment date by the current payment frequency for relative patterns. For absolute patterns, the next payment date is determined by the next consecutive date in the pattern.

   If the remaining number of payments is equal to 1, or the next payment date is greater than the maturity date, the next payment date is set equal to the maturity date.

Interest in Advance Calculations

The following steps are applicable to interest in advance records only. Interest in advance instruments make their first payment on the origination date. The last payment, on the maturity date, is a principal only payment.

1. Determine new current payment on schedules and patterns. Current payment is calculated as described in Step 2, earlier.

2. Calculate principal runoff.

   For interest in advance records, the principal runoff occur before the interest cash flow is calculated. Because conventionally amortizing instruments cannot have interest in advance characteristics, amortizing interest in advance instruments are always level principal. Therefore, the principal runoff equals the current payment amount.

   For the payment on the maturity date, all remaining principal is also paid off.

3. Update current balance. Prior to calculating the interest cash flow, the current balance must be updated for the amount of principal runoff. If the payment date is the maturity date, the balance is set to zero, and no further calculations are necessary.

4. Calculate interest cash flow. If the payment date is not the maturity date, an interest cash flow is made. The interest cash flow calculation for interest in advance instruments is similar to the interest in arrears calculation. The calculation differs in the count for number of days. Rather than counting from the last payment date to the current payment date, the number of days is counted from the current payment

date to the next payment date.

5. Update remaining number of payments. After a payment has been made, the underlying data must be updated in preparation for the next event. The remaining number of payments is reduced by 1.

6. Update next payment date. For standard amortization instruments, the next payment date is set equal to the current payment date plus the payment frequency.

   Next payment date$_m$ = Current payment date + payment frequency

   If the instrument is an amortization schedule, the next payment date is determined from the dates in the schedule table.

   If the instrument is an amortization pattern, the next payment date is determined by incrementing the current payment date by the current payment frequency for relative patterns. For absolute patterns, the next payment date is determined by the next consecutive date in the pattern.

   If the remaining number of payments is equal to 1, or the next payment date is greater than the maturity date, the next payment date is set equal to the maturity date.

Prepayment Event

Cash flow data characteristics are:

Static Information

- Current gross rate
- Current net rate
- Payment frequency and multiplier
- Origination date
- Original Term
- Next Reprice Date
- Reprice Frequency
- Maturity Date

Dynamic Information

- Current Par Balance

- Current Payment

Additional Assumption Information

- Prepayment Assumption Rule
- Prepayment Model
- Forecast Rates Rule

Event Trigger

- Next Payment Date

Prepayment Event Steps

Perform the following in order to execute prepayment event steps:

1. Update value of prepayment dimensions.

   Depending on the prepayment assumptions for the product leaf, values for the prepayment dimensions may need to be updated. The prepayment assumptions for the product leaf are defined in a Prepayment Rule, which is then selected for the current processing run.

   If the prepayment method is Constant Rate, these updates are not necessary. If the prepayment method is Arctangent, only the rate ratio is necessary to calculate. For Prepayment Model method, the required updates depend on the dimension within the table for the proper origination date range.

   Following are the all possible prepayment dimensions and their calculations:

   Market Rate

   The market rate is selected per product within the Prepayment Rule. You must select an IRC from the list of IRCs defined in Rate Management > Interest Rates. The chosen IRC provides the base value for the market rate.

   Additionally, you must specify the reference term you want to use for IRCs that are yield curves. There are three possible methods for you to select:

   - **Original Term**

     The calculation retrieves the forecasted rate from the term point equaling the original term on the instrument.

   - **Reprice Frequency**

     The calculation retrieves the forecasted rate from the term point equaling the reprice frequency of the instrument. If the instrument is fixed rate and, therefore, does not have a reprice frequency, the calculation retrieves the forecasted rate associated with the term point equaling the original term on the instrument.

- **Remaining Term**

  The calculation retrieves the forecasted rate from the term point equaling the remaining term of the instrument. See the description of the remaining term calculation listed in the following for more details.

  The market rate is determined by retrieving the proper forecasted rate and adding the user-input spread.

  Market Rate = f(Current Date, IRC, yield curve term) + spread

Coupon Rate

The coupon rate is the current rate of the instrument record (as of the current date in the forecast).

Rate Difference

The rate difference is the spread between the coupon rate and the market rate. Prior to calculating this dimension, the market rate must be retrieved.

Rate Difference = Coupon Rate - Market Rate

Rate Ratio

The rate ratio is the proportional difference between the coupon rate and the market rate. Prior to calculating this dimension, the market rate must be retrieved.

Rate Ratio = Coupon Rate / Market Rate

Original Term

The original term is retrieved from the original term of the instrument. If the original term is expressed in months, no translation is necessary. Otherwise, the following calculations are applied:

$\text{Original Term}_{\text{months}} = \text{ROUND}((\text{Original Term}_{\text{days}})/30.412)$

$\text{Original Term}_{\text{months}} = \text{Original Term}_{\text{years}}*12$

Reprice Frequency

The value for reprice frequency depends on the adjustable type code and the tease characteristics of the instrument data.

- **Fixed Rate**

  If the instrument is fixed rate, as designated by an adjustable type code = fixed (code value = **0**), the original term, as defined earlier, is used as the reprice frequency.

  Reprice Frequency = Original Term (months)

- **Non-Tease Floating**

  If the adjustable type of the instrument is floating (code value of **30** or **50** and not in a tease period), the reprice frequency is assumed to be one day, which when rounded to a month value, becomes 0 months.

  Reprice Frequency = 0 months

- **Non-Tease Adjustable**

  If the adjustable type of the instrument is adjustable (code value of **250**) and not in a tease period, the reprice frequency columns is used. All cases where terms are not expressed in months should be translated into months, calculated as follows:

  $\text{Reprice Frequency}_{\text{months}} = \text{Reprice Frequency}_{\text{years}} * 12$

  $\text{Reprice Frequency}_{\text{months}} = \text{Round(Reprice Frequency}_{\text{days}} / 30.412)$

- **Teased Loans**

  The tease period is identified by a tease end date > current date. The reprice frequency during the tease period is calculated as follows, rounded to the nearest whole number of months.

  Reprice Frequency = ROUND((Tease End Date - Origination Date) /30.412)

Remaining Term

The remaining term value represents the remaining number of months until maturity. The value is rounded to the nearest whole number of months.

Remaining Term = ROUND((Maturity Date - Current Date)/30.412)

Expired Term (Age)

The expired term represents the age of the instrument. It represents the time elapsed since the origination of the instrument. The value is rounded to the nearest whole number of months.

Expired Term = ROUND((Current Date - Origination Date)/30.412)

Term to Reprice

As with reprice frequency, the calculation of term to reprice depends on the adjustable type code and tease characteristics of the instrument characteristics.

- **Fixed Rate**

  If the instrument is fixed rate, as designated by an adjustable type code = fixed (code value = **0**), the term to reprice is calculated in the same manner as remaining term. The value is rounded to the nearest whole number of months.

  Term to Reprice = Round (Maturity Date - Current Date/30.412)

- **Non-Tease Floating**

  If the adjustable type of the instrument is floating (code value of **30** or **50**), and is not in its tease period, the reprice frequency is taken as 1 day. The term to reprice is assumed to be one day, which when rounded to a month value, becomes 0 months.

  Term to Reprice = 0 months

- Non-Tease Adjustable

  If the adjustable type of the instrument is adjustable (code value of **250**) and not in its tease period, the term to reprice is calculated as the difference between the current date and the next reprice date. The value is rounded to the nearest whole number of months.

  Term to Reprice = ROUND((Maturity Date - Current Date)/30.412)

- **Teased Loans**

  The tease period is identified by a tease end date > current date. The term to reprice, while in this period, is calculated as the difference between the current date and the tease end date. The value is rounded to the nearest whole number of months.

  Term to Reprice = ROUND((Tease End Date - Current Date)/30.412)

2. Determine Base Annual Prepayment Rate.

   The method for determining the annual prepayment rate depends on the prepayment method.

   Constant Rate

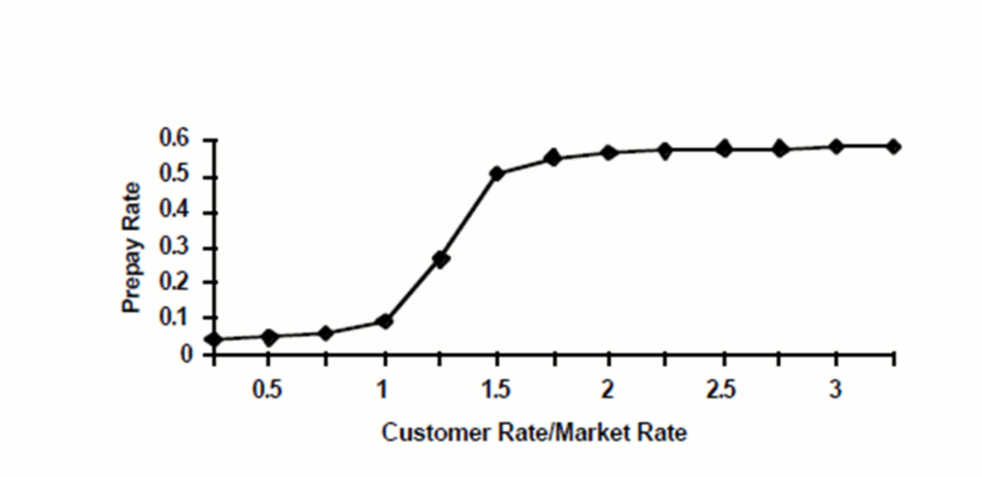
   Constant prepayment rates can vary for different origination date ranges. The rate is determined by finding the proper range of origination dates and using the constant rate from this range.

   Base Annual PP Rate = Constant Rate

   Arctangent

   The arctangent formula describes the relationship between prepayments and the ratio of coupon rate to market rate. Four coefficients you enter define the shape of the curve. These coefficients can vary by origination date range.

   Base Annual PP Rate = Coeff1-Coeff2 * ARCTANGENT(Coeff3* (Coeff4-Rate Ratio))



Prepayment Model

Under the Prepayment Model method, a Prepayment Model rule is referenced within the Prepayment Rule for a particular product and origination date range. This prepayment model may be factored by a coefficient to scale the prepayment rates that reside in the table up or down. The prepayment model factor is also defined per product and origination date.

The Prepayment Model rule contains a table of prepayment rates dimensioned by other characteristics, as listed in Step 1, earlier. The Prepayment Model rule can hold a maximum of three dimensions. For each dimension, you can define the lookup method along that dimension, either range or interpolate.

Range Lookup

Range Lookups treats the nodes within the dimension as a starting value for a range that extends to the next node dimension. For example, take an original term dimension with node values of 0,12, and 24. The range lookup treats these values as three sets of ranges: 0 to 11, 12 to 23, and >= 24.

Interpolation Lookup

If the interpolation method is selected, the lookup applies straight line interpolation to determine the proper prepayment rate for values that fall between nodes.

Lookups Outside the Given Range

For both lookup methods, lookup for values less than the lowest node value receives the prepayment rate associated with the lowest node. Values greater than the highest node receive the prepayment rate associated with the highest node.

Along each dimension of the table, range lookup or interpolation is performed to pinpoint the proper prepayment rate from the table. Once the prepayment rate is retrieved from the prepayment table, the prepayment table factor is applied to this rate.

Base Annual PP Rate = PPModelFactor * PPTModelLOOKUP(dimension$_x$, dimension$_y$, dimension$_z$)

3. Adjust for Seasonality.

   For each prepayment method, seasonality factors can be applied to adjust the prepayment rate. The seasonality factors are defined per month. The month of the current date is used to determine the proper seasonality factor to use.

   Annual PP Rate = Seasonality Factor(Current Month) * Base Annual PP Rate

4. Check Prepayment in Full Option.

   If the adjusted final prepayment rate is equal to 100%, the instrument is paid off in full.

5. De-annualize the Prepayment Option.

   The annual prepayment rate is adjusted to a rate per payment. The formula is as follows:

   Prepayment Factor = (1-(1- Annual PPRate)^(1/payments per year)

6. Adjust Prepayment Rate for Stub or Extended Payments.

   The prepayment rate per payment is adjusted if the payment is a stub or extended payment. This adjustment is made in the same manner that interest cash flows are adjusted, as follows:

   Adjusted prepayment factor = Prepayment Factor * (next payment date - last payment date) / (payment frequency in days)

7. Determine prepayment amount.

   The amount of runoff due to prepayments is calculated. The prepayment factor is applied to the current balance.

   Prepayment Runoff = Current Balance * prepayment factor

8. Update current balance.

   The current balance must be reduced by the amount of prepayment runoff.

   Current Balance = Current Balance - Prepayment runoff

9. Apply prepayment factor to current payment.

   An option exists in the Prepayment Rule to reduce the payment proportionally to reflect the amount of principal that has been prepaid. If the prepayment treatment is **Refinance**, the current payment is reduced as follows:

   Current Payment = Current Payment * (1 - prepayment factor)

   If the payment treatment is **Curtailment**, the current payment remains constant, effectively reducing the term of the instrument.

Reprice Event

Cash flow data characteristics are:

Static Information:

- Adjustable type code
- Interest rate code
- Net Margin
- Net Margin Code
- Gross Margin
- Reprice frequency and multiplier
- Rate cap life
- Rate floor life
- Rate increase period
- Rate decrease period
- Rate set lag and multiplier
- Rate change minimum
- Rate change rounding code
- Rate change rounding factor

Dynamic Information

- Current gross rate
- Current net rate
- Current transfer rate
- Current TP Adjustment rates

Event Triggers

- Bucket start date
- Tease end date
- Next reprice date

Additional Assumption information

- Forecast Rates

Notes About Reprice Event

Transfer Pricing

The modeling of adjustable rate instruments in TP begins at the last reprice date and ends at the next reprice date, with the next repricing date treated like a maturity date for funding purposes. Therefore, no repricing events occur during a transfer pricing process.

Customer Rate Definition

Repricing characteristics (rate caps, floors, periodic change limits) are based on the customer rate. In a standard processing run, the current net rate is defined as the customer rate. However, when the Modeling with Gross Rates option is used, the customer rate is defined as the current gross rate.

Reprice Steps

1. Determine new IRC value(s).

   The raw customer rate (Raw Rate_C) is determined from the set of forecasted IRC values contained in the Forecast Rates assumption rule chosen within the ALM Deterministic Process. Additionally, a new transfer rate (Transfer Rate_T) is derived if the Modeling with Transfer Rates option is used. The variables used to determine the transfer rates are:

   Raw $\text{Rate}_C = f$(Rate set date, IRC, yield curve term)

   Rate Set Date

   The rate set date is the date from which the IRC value is taken. The date is determined as follows:

   Rate Set Date = Next Reprice Date_m - Rate Lag_r

   If the rate set date is less than the As of Date, the historical rate is retrieved from up to one year prior to the As of Date.

Yield Curve Term

If the IRC is a single point IRC (Prime, LIBOR), then the forecasted rate is used. If the IRC is a yield curve (for example, Treasury Yield Curve), the point on the yield curve equivalent to the repricing frequency is used. If no such point exists, straight line, cubic spline or quartic spline interpolation is used to find the rate between the two nearest terms.

Example: (straight line interpolation):

An instrument has a repricing frequency of 18 months, which does not exist on the yield curve. The two nearest points are the 12 month point and the 24 month point.

The 18 month point is determined as follows:

Term	Rate
12 Months	6.00%
24 Months	9.00%

Interpolated Rate = Rate(Begin) + (Rate(End) - Rate(Begin))*(Term(interpolated rate)-Term(Begin))/(Term(End)-Term(Begin))

Rate(18 M) = 6.00% + (9.00%-6.00%) * (18 M - 12 M) / (24 M - 12 M) = 6.00% + 3.00% * (6 M) / (12 M) = 6.00% + 1.50% = 7.50%

2. Add applicable margin to raw customer rate.

   The margin of the customer is added to the raw customer rate. If using the Modeling with Gross Rates option, the gross margin is used. Otherwise, repricing depends on the Net Margin Flag. If the Net Margin Flag is set to floating net rate, the net margin is used. If the Net Margin Flag is set to fixed net rate, no repricing occurs.

   Raw $Rate_C$ = Raw $Rate_C$ + $Margin_C$

3. Update current transfer rate if modeling transfer rates.

   At this point the current transfer rate (Current $Rate_T$) can be updated. Unlike the customer rates, no further adjustments are necessary.

   Current $Rate_T$ = Raw $Rate_T$ + $margin_T$

4. Apply rounding codes.

   The raw customer rate is adjusted using the method defined by the rounding codes and to the precision specified by the rounding factor. If the rounding factor is set equal to zero, no rounding occurs.

Method	Description	Example (Raw rate = 5.123; rounding factor = 0.01)
No Rounding	Rate is not rounded	5.123%
Round Up	Rate is rounded to the nearest value greater than the raw rate with the specified precision	5.1300%
Round Down	Rate is rounded to the nearest value less than the raw rate with the specified precision	5.1200%
Truncate	Rate is truncated to whole value	5.0000%
Round Nearest	Rate is rounded to nearest value to the raw rate with the specified precision	5.1200%

5. Apply rate change minimum.

   The raw customer rate including margin is compared with the current customer rate (Current $Rate_C$). If the amount the current customer rate would change by is less than the rate change minimum then the rate does not change. Therefore, raw customer rate is set equal to the current customer rate.

Condition	Absolute Value (Raw Rate - Current $Rate_C$) < Rate Change Minimum
Adjustment if True	Raw Rate = Current $Rate_C$

6. Set value of fully indexed rate(s).

   The fully indexed rates are updated after rate change minimums and rounding codes are applied, and before caps and floors are applied to the raw rate.

Customer Rate Definition	Floating Net Rate	Fixed Net Rate
Gross Rate	Fully Indexed Rate_G = Raw Rate_C	Fully Indexed Rate_G = Raw Rate_C
	Fully Indexed Rate_N = Raw Rate_G – Margin_G + Margin_N	Fully Indexed Rate_N = Current Rate_N
Net Rate	Fully Indexed Rate_N = Current Rate_N	Fully Indexed Rate_N = Raw Rate_C

7. Calculate tease effect financial elements for instruments in a tease period.

   For instruments in a tease period, determined by tease end date > current date in modeling horizon, no adjustments are made to the current rate. However, the effect of the tease is recorded in two financial elements that are used at the next payment to calculate the income effect of the tease. On a tease record, the processing of a reprice event is complete at this point.

Financial Element	Calculation
tease rate	Fully Indexed Rate_C - Raw Rate_C
tease balance	Current Balance_m

8. Check periodic caps and floors.

   The customer rate cannot change by more than the amount specified by the periodic change limits (periodic floor and periodic cap). If the raw customer rate would effect a change to the current customer rate that exceeds the periodic change limitations, the current customer rate is adjusted only by the amount specified by the periodic change limit. If periodic limits are applied to the raw customer rate, then this adjustment occurred should be recorded in the periodic cap/floor financial elements.

	Decreasing Rate Environment	Increasing Rate Environment
Condition	Raw $Rate_C$ < Current $Rate_C$ and Current $Rate_C$ - Raw $Rate_C$ > Periodic floor	Raw $Rate_C$ > Current $Rate_C$ and Raw $Rate_C$ - Current $Rate_C$ > Periodic cap
Adjustment if True	Raw $Rate_C$ = Current $Rate_C$ - Periodic floor	Raw $Rate_C$ = Current $Rate_C$ + Periodic cap

9. Check lifetime caps and floors.

   The customer rate cannot be greater than the lifetime cap or less than the lifetime floor. If the raw customer rate fails either of these conditions, the raw customer rate is set equal to the appropriate value.

	Decreasing Rate Environment	Increasing Rate Environment
Condition	Raw $Rate_C$ < Rate Floor Life	Raw $Rate_C$ > Rate Cap Life
Adjustment if True	Raw $Rate_C$ = Rate Floor Life	Raw $Rate_C$ = Rate Cap Life

10. Update Current Rates

Customer Rate = Gross Rate	Current $Rate_G$ = Raw $Rate_C$ Current $Rate_N$ = Raw $Rate_C$ - $Margin_G$ + $Margin_N$
Customer Rate = Net Rate	Current $Rate_N$ = Raw $Rate_C$

11. Update Next Repricing Date

   The next reprice date is rolled forward in preparation for the next repricing event. If the adjustable type code is **Adjustable**, the next reprice date is calculated as:

   Next reprice date = Current reprice date + reprice frequency

   If the adjustable type code is **Floating** or **Variable**, the next reprice date is set equal to the first date in the next modeling bucket.

12. Trigger Payment Recalculation

    If the amortization type is a standard conventional amortization, the current payment on the instrument data is updated based on the new rate. (See Payment Calculation Event, page 5-19).

Additional Processing Events

Deferred Amortization Calculation Steps

1. Determine the flat rate scenario.

    1. The process may already have a flat rate scenario. If the change from the base rates is zero for all buckets and all interest rate codes, then there is a flat rate scenario.

    2. If there is not a flat rate scenario, then a flat rate scenario is created. This scenario is created by reading in the base rates and applying a zero change for all buckets.

2. Determine if a record needs to have deferred amortization records calculation applied to it.

    Deferred amortization records are instruments or new business records where the column Deferred_cur_bal is not equal to zero.

3. Calculate cash flows for instrument from as-of-date until maturity using the flat rate scenario.

    1. If one of the scenarios within the process is a flat rate scenario, further cash flows do not need to be generated. The cash flows from the flat rate scenario can be used.

    2. If the instrument is not rate-sensitive, further cash flows do not need to be generated. The cash flows from any scenario can be used.

    3. In all other cases, cash flows in a flat rate scenario must be generated.

4. Calculate the internal rate of return.

    1. Calculate the Clean Price (Market Value – Accrued Interest) of the instrument using the coupon rate, the cur_net_rate, as the discount rate from as-of-date to maturity date. The principal and interest cash flows from the as-of-date to the maturity date should be used in all cases.

    2. Use the following simple market value calculation:

$$\text{Clean Price} = \sum_{n=1}^{N} \frac{CF_n}{(1+c)^n}$$

c = coupon rate (cur_net_rate) from instrument record

n = payment number

N = remaining number of payments

CF_n= cash flow at payment n = total runoff plus interest cash flow net

3. Calculate the duration of the instrument using the same discount rate.

$$\text{Duration} = \frac{\sum_{n=1}^{N} \frac{CF_n * n}{(1+c)^{n+1}}}{\sum_{n=1}^{N} \frac{CF_n}{(1+c)^n}}$$

4. Find the amount necessary to add to the coupon rate to more closely approximate the internal rate of return, using the following formula:

$$\text{Rate Change} = \frac{Clean\ Price - (ParBal + Deferred)}{duration * Clean\ Price}$$

5. Using a new discount rate equal to the coupon plus the rate change calculated earlier, recalculate the clean price and duration.

6. Repeat steps 4 and 5 until the rate change is very small (less than 0.001% in absolute value). The discount rate used when this state is reached is the internal rate of return. It is negative if:

$$\sum_{n=1}^{N} CF_n < ParBal + Deferred$$

5. Calculate the spread to use in each scenario.

   1. The calculation is:

Spread = *(IRR -c)*

2. The value for the coupon **c** in the formula mentioned earlier, is the cur_net_rate after the first true repricing event. A true repricing event is a non-tease repricing event.

3. If the instrument is fixed rate, the cur_net_rate from the record is sufficient.

6. Calculate the deferred runoff financial elements in each bucket and for each scenario.

   1. The deferred runoff must be calculated first.

      For each modeling bucket, calculate the amount of total income to be recognized as:

$$\text{Total Income} = (ParBal + Deferred) * \frac{c_b + spread}{a}$$

$$\text{Total Income} = DeferredRunoff + InterestAccrued$$

$$\text{Deferred Runoff} = (ParBal + Deferred) * \frac{c_b + spread}{a} - InterestAccrued$$

a = accrual factor, see the following description.

c_b = Average Rate

ParBal = Average Balance

Deferred = Deferred End Balance in previous bucket, Deferred_cur_bal in bucket 1

InterestAccrued = Interest Accrual Net for Current Bucket

- The value **a** in the formula mentioned earlier, is the accrual factor associated with that bucket. The accrual factor is the portion of the year that the modeling bucket represents. This calculation varies according to the accrual basis code associated with the instrument.

- The financial element 140, Average Balance should be used for ParBal in the formula mentioned earlier.

- The financial element 160, Average Rate should be used for c_b in the formula mentioned earlier.

- In the bucket in which the instrument matures, if this bucket falls within the modeling horizon, the deferred runoff should be set equal to the remaining deferred balance. This is to ensure that the entire deferred balance is run off by the maturity date.

2. Calculate the change in the deferred balance as:

   $\text{Deferred Balance}_{end} = (\text{Deferred Balance}_{beginning} - \text{Deferred Runoff})$

3. Assume that the average deferred balance is equal to the beginning deferred balance.

Accounting for Exchange Rate Fluctuations

The following discussion applies to Oracle ALM currency-based processing. As mentioned earlier in this chapter, currency-based processing occurs when an Oracle ALM Deterministic Process is set up with functional dimensions of either Product/Currency or Product/Organizational Unit/Currency.

Definition of Currency Methods

The effect of currency fluctuations on balance sheet accounts can be modeled in Oracle ALM in one of three ways. In the Product Characteristics rule, the user selects one of the following currency accounting methods for each Product/Currency combination. If no method is specified, the Temporal method is used:

Temporal Method	Currency gains/losses on the outstanding balance are reflected in the income statement each reporting period. Because currency effects have already been reflected in the income statement, no additional translation adjustment accounts are necessary.
Current Rate Method	The reported balance at the end of the period reflects the value at the current exchange rate. An Accumulated Translation adjustment account is maintained, which stores the accumulated unrealized gains/losses on principal as a contra-equity account on the balance sheet. When principal payments are made, the realized gain/loss is recognized in the income statement and the contra-equity account is adjusted appropriately.
Historical Basis Method	Balances are carried at the historical cost. The effect of exchange rate fluctuation on cash flows is recorded as a realized gain/loss in the income statement when received.

For auditing purposes, the currency financial elements are retained separately for each local currency. Realized and unrealized gain/loss entries are recorded in a series of financial elements, as described in the following chart.

Term used in this chart:

- Current Exchange Rate: Rate in effect for that particular modeling bucket.
- Previous Exchange Rate: Rate in effect for the previous modeling bucket.
- Original Exchange Rate: Rate in effect at the origination of the instrument.

Code	Description	Method(s)	Purpose	Calculation
950	Accumulated Translation Amount	Current Rate	Accumulated effect of unrealized currency gain/loss on balances at the end of each period.	(FE100 Ending Balance) * [(1/Current Exchange Rate) – (1/Original Exchange Rate)]
465	Total Currency Gain/Loss (Principal)	Temporal	Effect of exchange rate fluctuation on the existing balance over the current period.	(FE60 Beginning Balance) * [(1/Current Exchange Rate) - (1/Previous Exchange Rate)]
475	Realized Currency Gain/Loss (Principal)	Current Rate and Historical Basis	Actual change in value of the principal cash flow due to exchange rate fluctuation.	(FE210,212 Total Runoff) * [(1/Current Exchange Rate) -(1/Original Exchange Rate)]
485, 486 or 487	Realized Currency Gain/Loss (Interest-- Net, Gross, or Transfer Rate)	All Methods	Actual change in value of the interest cash flow due to exchange rate fluctuation.	Interest Cash Flow * [(1/Current Exchange Rate) -(1/Original Exchange Rate)]

For a complete listing of financial elements used for cash flow results, see the Financial Element Calculations chart, page 17-81.

Examples of Exchange Rate Fluctuations

The following example illustrates the differences among the currency accounting methods.

This is a United States Bank (USD) with holdings in Japan (JPY) and Germany (DEM).

	Historical Rate (at origination)		Current Rate (at As of Date)		Forecast (Month 1)	
Local Balance	**Exchange Rate**	**USD Balance**	**Exchange Rate**	**USD Balance**	**Exchange Rate**	**USD Balance**
JPY 100,000	120 JPY/USD	$833.33	133 JPY/USD	$751.88	130 JPY/USD	$769.23
DEM 2,000	1.8 DEM/USD	$1,111.11	1.69 DEM/USD	$1,183.43	1.6 DEM/USD	$1,250.00

In this example, the Japanese Yen holdings have decreased their US Dollar value since origination, and the German Mark holdings have increased their US Dollar value since origination. One month into the forecast, the Yen-to-Dollar exchange rate is forecast at 130 and the Deutschmark-to-Dollar exchange rate is forecast at 1.60.

- Temporal Method

  If the Temporal method is used, the change in value should be reflected in net income and passed through to retained earnings. The change in value from the current USD balances to the USD balance in Month 1 is reflected as a realized gain. In this case, the currency gain account would reflect a net amount of $83.92 for Month 1.

  JPY currency change $ 17.35

  DEM currency change <u>$66.57</u>

  Currency gain for Month 1 $83.92

  That is, (see Financial Element 465, page 5-47) the formula for total gain/loss on principal is:

  Beginning Balance * [(1/current exchange rate) - (1/previous exchange rate)]

- Current Rate Method

  If we apply the Current Rate method, the Accumulated Translation Amount, a separate contra-equity account, reflects a total of $74.79. This is derived from the accumulated difference between the USD balance from origination to Month 1 of the forecast. In this case, no principal payments have been made; USD value of Yen holdings have decreased in value since origination, but the USD value of Deutschmark holdings have appreciated more, resulting in a net increase in value.

JPY Accumulated Translation Amount - $64.10

DEM Accumulated Translation Amount <u>$138.89</u>

Accumulated Translation $74.79

That is, (see Financial Element 950, page 5-47) the formula for unrealized gain/loss on principal is:

Ending Balance * [(1/current exchange rate) - (1/original exchange rate)]

- Historical Basis Method

  All balances are carried at the historical rate, so there is no need for an accumulated translation account. Currency gains/losses are realized when cash flows are received.

Effect of Exchange Rate Forecasts on Cash Flows

Let's modify our example slightly, to show a payment made in Month 1.

Amortization of principal and payments of interest must reflect the effect of changes in the exchange rate.

All methods treat interest payments in the same manner: They are valued when received and corresponding gains or losses are then realized (see Financial Elements 485-487, page 5-47). Therefore, the following discussion focuses on currency accounting for principal amortization. In this example, both the Yen and Deutschmark accounts have runoff equal to 5% of the outstanding balances.

Local Currency Runoff Amount	USD Runoff Amount (historical rate)		USD Runof Amount (current rate)		Realized Gain/Loss on Principal
JPY 5,000	120 JPY/USD	$41.67	130 JPY/USD	$38.46	-($3.21)
DEM 100	1.8 DEM/USD	$55.56	1.6 DEM/USD	$62.50	$6.94

- Temporal Method

  In the Temporal method, the effect of exchange rate fluctuations on the balance sheet is reflected at the period end based on the balance at the beginning of the period. This method requires no additional recognition of gain/loss on principal because it has already been recognized on the beginning balance (see Financial Element 465, page 5-47, discussed earlier).

- Current Rate Method

  In the Current Rate method, the Accumulated Translation Balance (Financial Element 950) should reflect only the change in value of the remaining balance, based on the current and historical exchange rates.

  [JPY 95,000 / (130 JPY/$)] – [JPY 95,000 / (120 JPY/$)] =

  $730.77 - $791.67 = -$60.90... or compute as:

  Ending Balance * [(1/current exchange rate) - (1/original exchange rate)]

  In addition to the Accumulated Translation Balance, a realized currency gain/loss should be reflected in the income statement. Realized currency gain/loss transactions reflect the change in currency from the time of origination of an instrument to the time of cash flow repayment, when the initial balance loaned/invested/borrowed must be translated into the reporting currency. At each payment, the realized gain/loss on the principal cash flows (scheduled payments, maturing balances, and prepayments) are calculated based on the amount of the cash flow and the change in exchange rate. That is, (see Financial Element 475, page 5-47) the formula for realized gain/loss on principal is:

  Principal Cash Flow * [(1/current exchange rate) - (1/original exchange rate)]

- Historical Basis Method

  When payments are made, Historical Basis accounts should also reflect the realized gain/loss on currency in the income statement. The method used should be the same as the method described for the Current Rate method; that is (see Financial Element 475, page 5-47), the formula for realized gain/loss on principal is:

  Principal Cash Flow * [(1/current exchange rate) - (1/original exchange rate)]

Market Value Calculation

Cash Flow Inputs

Interest Cash Flow (Net, Gross, or Transfer Rate)

Total Runoff

Repricing Balance

Deferred Runoff

Interface Inputs

Discount Methods

Forecast Rates

Market Value Calculation Steps

1. Define components of cash flow.

   Within the interface, you must define what components make up the cash flow that is discounted to derive the market value. The standard components of a cash flow are the following:

   - Interest Cash Flow Net
   - Scheduled Principal Runoff
   - Principal At Maturity
   - Prepayments

   Choosing special options in the Discount Rates UI adjusts the cash flow definition in the following manner:

Cash Flow Switches	Effect on Cash Flow
Interest Cash Flow	Value interest component of cash flow only
Mature At Reprice	Value instrument as if it matured on the first repricing date after the start date

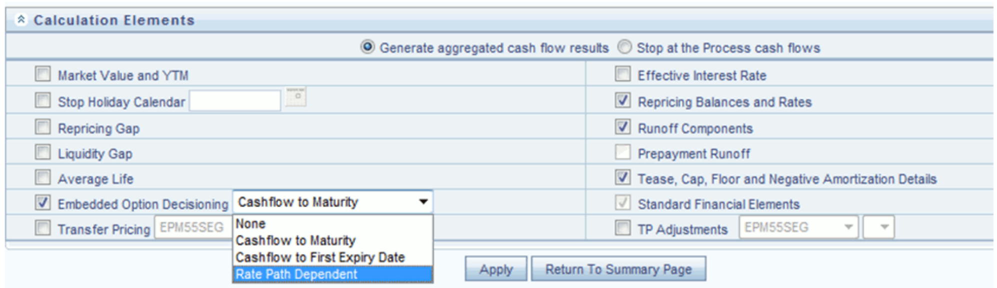
Future Originations Adjustment

Instruments that originate after a designated start date can be included in the market value for that start date if the issue date is less than or equal to the start date. In this case, the negative flow of funds on the origination date is considered to be a cash flow for discounting purposes.

In the case of Bonds with Embedded Options, forward Starting Instruments would have a change while calculating the market value (MV) for determining the exercisability of the option. The initial cashflow outlay would not be part of the market value calculation. Because, including the initial cash flow outflow while calculating the market value significantly reduces the MV. And the 'in the moneyness' comparison would not make sense.

Market Value Calculation for Bonds with Embedded Options:

The trigger for the Bonds with Embedded options calculation is the EMBEDDED_OPTION_FLG in the instrument table and selection in the Calculation Element block in the process definition (Needless to say corresponding entries in the FSI_D_EMBEDDED_OPTIONS and FSI_D_EMBEDDED_OPTIONS_SCH tables). Screenshot of the calculation element block follows:



The calculation logic for each option and expiry style explained in detail in the Bonds with Embedded Options Chapter.

2. Determine discount rate for cash flow.

   Within the Discount Rates rule, you specify an IRC and a discount method. The methodology determines whether current or forecasted rates are referenced and which yield curve point from the chosen IRC is used.

Discount Method	Date of IRC	Yield Curve Point
Spot Input	not applicable	not applicable
Spot IRC	start date	payment date - start date
Forecast Original Term	payment date	fixed rate: original term adjustable: reprice frequency
Forecast Remaining Term	payment date	payment date - start date

   The Discount Rate is calculated by following formula:

   1/(1+(Forecasted Rate/100))^Days Expressed as Year fraction

   Here, Days Expressed as Year Fraction = Days in Payment/Days in the Year

   FE490 = (Discount Rate* PV of Cashflow)/ Total MV

3. Calculate market value of cash flow.

   For the market value of an instrument as of a particular start date, the present value of each cash flow is calculated for all cash flows that occur after the start date, using the appropriate discount factor.

4. Treat reprice date as maturity where necessary.

   For repricing instruments, the cash flows are evaluated from the start date up to the reprice date, effecting a maturity on the reprice date for duration calculation. For market values, this method is used if the **Mature At Reprice** option is enabled in the Discount Rates interface.

   If this methodology is used and the reprice date falls mid-payment, an extra interest cash flow must be calculated. This interest cash flow represents the portion of the next interest cash flow that applies from the last payment date prior to the reprice date and the reprice date.

5. Sum market values of cash flows.

   The market values per payment are summed to arrive at a total market value number.

$$\text{Total Market Value} = \sum_{n=1}^{n} (\text{MV}_{\text{cf}})$$

6. Calculate additional financial measures of the instrument.

   The **Macaulay Duration** of the instrument is calculated by weighting the market value of each payment by time.

$$\text{Macaulay Duration} = \frac{\sum_{n=1}^{n} (\text{MV}_{\text{cf}} * \text{t})}{\sum_{n=1}^{N} \text{MV}_{\text{cf}} * \frac{\text{pmts}}{\text{year}}}$$

   The **Modified Duration** of the instrument is calculated by dividing the Macaulay Duration by YTM.

$$\text{Modified Duration} = \frac{\text{Duration}}{\left(1 + \frac{\text{Yield to Maturity}}{\text{Payment per year}}\right)}$$

   The **Convexity** of the Instrument is calculated as follows:

$$\text{Convexity} = \frac{\text{MV} \times \text{Time Expired} \times (1 + \text{Time Expired})}{(1 + \text{Discount Rate})^2}$$

The **Average Life** of the Instrument is calculated as follows:

$$\text{Average Life} = \frac{\sum_{t=1}^{n} \text{Principal Runoff} * (\text{Principal Runoff date} - \text{As of Date})}{\frac{\sum_{t=1}^{n} \text{Principal Runoff}}{365.25}}$$

The **Yield to Maturity** in OFSAA is the interest rate that equates the present values of the cash flows (coupons & maturity value) to the present Market price.

The YTM takes into account all the sources of income that is:

- Coupon interest
- Capital Gain or Loss
- Reinvestment income - assuming that all the coupons are reinvested at a rate equal to the YTM

The **Yield to Maturity** is calculated as mentioned below.

The CFE uses Newton-Rapshon method for the calculation of the Yield to Maturity. It starts with the CUR_NET_RATE in the instrument record, and iteratively calculates the YTM to a precision of 0.000001. Pseudo code follows:

<u>Initial Value</u>

Yield to Maturity = Current Net Rate

Delta = 0

Total MV=0

Total Derivative =0

Maximum Iterations =500

Conditions: absolute of Delta > 0.000001 and iteration > Max Iterations

Formula: *Yield to Maturity= Yield to Maturity-Delta*

Where:

$$Delta = \frac{Clean\ Market\ Value - Total\ MV\_1}{Total\ Derivative}$$

$$Total\ MV = PV\ (First\ FE\ 430) + PV\ (other\ FE\ 430s) + PV\ (FE\ 210)$$

$$Total\ MV\_1 = Total\ MV - Accrued\ Interest$$

$$Total\ MV = Total\ MV + Cashflow$$

$$Cashflow = \frac{Cash\ Flow}{(1 + \frac{Yield\ to\ Maturity}{Payment\ per\ year})^{\frac{Time\ Expired}{Active\ Payment\ Frequency}}}$$

$$Total\ Derivative = Total\ Derivative + \frac{\frac{\frac{Term\ Expired * Cashflow}{Active\ Payment\ Frequency}}{Payment\ per\ year}}{(1 + \frac{Yield\ to\ Maturity}{Payment\ per\ year})}$$

Iteration = Iteration + 1

Note: If it is a first Interest Cash Flow, then (First Interest Cashflow = ICF),

$$ICF = ICF - (ICF * \left(\frac{As\ of\ Date - Last\ Payment\ Date}{Next\ Payment\ Date - Last\ Payment\ Date}\right))$$

The **effective interest rate** is the true rate of interest earned. The Effective Interest Rate is calculated as follows:

Initial Value

Yield to Maturity = Current Net Rate

Delta = 0

Total MV=0

Total Derivative =0

Const Max Iterations =500

Iteration =0

The below calculations are in 'do – while' loop.

```
Conditions:  absolute of Delta > 0.000001 and  Iteration > Max
Iterations


Formula:

YTM = YTM – Delta.

Where, Delta = (Cur_book_bal + fees – Total MV ) / Total Derivative

        Total MV = Total MV + Cash Flow

Cash Flow = (Discount Type) / Power ( ( 1.0 + YTM / Payment Per
Year) , ( Expired Term / Active Payment Freq ) )

If,
'Calculate Market Value' is checked in the UI:
    Discount Type = Interest only, then FE 430
    Discount Type = Principal & Interest, then FE 210 + FE 430
    Discount Type = Reprice Balance, then FE 250

'Calculate Market Value' is not checked in the UI:
    Discount Type = Principal & Interest, then FE 210 + FE 430

Total Derivative = Total Derivative + (Expired Term / Active Payment
Freq ) * Cash Flow / PmtPerYear / ( 1.0 + YTM / Payment Per Year )

             Iteration = Iteration + 1

If it is a first Interest Cash Flow (FE 430) - ICF,
ICF = ICF  – (  ICF * ( Start Date in Julian – Last Pmt Date in
Julian ) / ( Next Pmt Date in Julian – Last Pmt Date in Julian ) )
```

The YTM calculated is as per the COMPOUNDING_FREQ of the instrument.

And, this YTM Calculated * CUR_PAR_BAL is written to the RES_MASTER YTM in the RES_MASTER = YTM calculated * CUR_PAR_BAL

7. Update instrument data.

   Within the ALM Deterministic Process user interfaces (Static and Dynamic), you can choose to write any of the preceding financial measures for a specified start date back to the instrument table. If this option has been selected, the financial measures are written to the following columns:

Financial Measure	Column Name
Market Value	MARKET_VALUE_C
Effective Interest Rate	EFF_INTEREST_RATE_C
Clean Price	MARKET_VALUE_CLEAN_C
Macaulay Duration	DURATION_C
Modified Duration	MODIFIED_DURATION_C
Convexity	CONVEXITY_C
Average Life	AVERAGE_LIFE_C
Yield to Maturity	CUR_YIELD

The CUR_YIELD column in the instrument record is the annualized yield to maturity of the instrument. And, it calculated as follows:

$$CUR\_YIELD = \left\{\left(1 + \frac{YTM}{No\ of\ Payments}\right)^{No\ of\ Payments} - 1\right\} * 100$$

Note: The difference in output between RES_MASTER and instrument table is that Instrument table outputs annualized results.

Consolidation of Results

If the Oracle ALM Process optionally specifies **Consolidate to Reporting Currency**, a cross-currency consolidation is performed. The local currency results are consolidated into a single reporting currency based on the historical and forecast exchange rates. Once translated into the reporting currency, common products are aggregated and output to a separate table. For example, if the Process system identifier is uniquely identified by Sys_ID_Num 99999, results would be held as follows:

	Results Table(s)	Calculations
Detailed Results	RES_DTL_99999 FSI_O_RESULT_MASTER, identified by Result_Sys_ID	The instrument is processed in its local currency and all results are accumulated in the local currency. Currency gain/loss is computed based on currency method defined for Product/Currency.
Consolidated Results	Cons_Dtl_99999 OFSA_Consolidated_Master, identified by Result_Sys_ID	Translates cash flows in each modeling bucket; consolidates across currencies.

The currency accounting method determines which exchange rates are used for the translation of each Financial Element. Typically, forecast exchange rates are used for the Temporal and Current Rate methods, and original exchange rates are used for the Historical Basis method. For the Temporal and Current Rate methods, some exceptions apply:

- Deferred Balances

  Deferred balances are balances that reflect prepaid amounts that are prepaid fees, amortized costs, premiums or discounts. Because the cash flows associated with these balances have already occurred, there is no currency risk associated with them. Therefore, these financial elements are reflected at cost, at the exchange rate at the time of origination.

- Interest Accruals

  The standard historical interest accrual financial element is reflected at the original exchange rate because the currency gain/loss account already reflects the change in interest due to currency. However, in order to calculate yields consistently with the average balance, a separate financial element is calculated for the current basis interest accrual; this reflects the interest accrual based on the current bucket exchange rate.

- Gap Financial Elements

  For more details, see Currency-Based Gap Modeling, page 5-59 .

For Consolidated Master results: Deferred balances are translated using the exchange rate in effect when the instrument was originated; all other balances are translated using the exchange rate in effect at the As-of-Date or future Start Date:

- As-of-Date values use the actual exchange rate in effect on the As-of-Date.
- Future originations and Dynamic Start Date values respectively use the forecast exchange rate in effect on the future origination date or Start Date.

If no exchange rate is found, the cash flow engine logs an error message and set the exchange rate equal to 1.

Currency-Based Gap Modeling

The processing steps depend on the relationship between the gap and cash flow modeling buckets. Because modeling buckets may differ between the cash flow results and the gap results and exchange rate forecasts are defined for cash flow modeling buckets only, additional processing steps are employed when there are bucket differences.

- **Compatible** (consistent) gap and cash flow buckets

  If cash flow modeling buckets and gap buckets are equal, or if multiple gap modeling buckets can evenly fit into one cash flow modeling bucket, the forecast exchange rates for the cash flow modeling buckets are used.

- **Unevenly-Overlapping** gap and cash flow buckets

  If cash flow modeling buckets are smaller than gap modeling buckets or if their respective start and end dates do not coincide, the engine must derive the forecast exchange rates for the gap modeling buckets.

  In this special case, the cash flow engine calculates a time-weighted exchange rate forecast for the gap bucket. Bear in mind that with a time-weighted rate, the consolidated gap results may be different than the consolidated cash flow results, even though they came from the same local balance. For example, a cash flow of $10 at an exchange rate of 1 and a cash flow of $20 at an exchange rate of 2 sums up to a translated cash flow of 20 (10 divided by 1, plus 20 divided by 2). If the weighted-average exchange rate is calculated as 1.5, the translated cash flow equals 20 (10 plus 20, divided by 1.5).

We recommend that consistent modeling and gap buckets be used in multicurrency simulations, to eliminate potential data inconsistency issues.

Detail Cash Flow Data

Column Name	Column Description	Column Type	Event use
ID_NUMBER	Unique identifier.	static	I

Column Name	Column Description	Column Type	Event use
COMMON_COA_ID	Leaf value used to determine financial account type of detail instrument.	static	I
ADJUSTABLE_TYPE_CD	Determines whether reprice occurs, and, if it occurs, whether it occurs according to reprice dates or bucket dates.	static	R
ACCRUAL_BASIS_CD	Method of accrual used in determining the rate per payment.	static	P
ACCRUED_INTEREST	For Non Maturity Behavior Pattern instruments, the Accrued Interest amount – up to the as-of-date, is added to the computed interest (As-of-date to next-payment-date) to arrive at the full interest amount on next payment date.	Static	P, I
AMRT_TERM & AMRT_TERM_MULT	Determines time over which principal is amortized; used in payment recalculation.	static	P, PC
AMRT_TYPE_CD	Determines method for amortizing principal. Used to match to payment pattern data.	static	PC
AMORT_METH_PDFC_CD	Determines the method for amortizing premiums and discounts.	Static	D, I

Column Name	Column Description	Column Type	Event use
BEHAVIOUR_TYPE_CD	Determines the type of Behavior Pattern, e.g. Non Maturity, Non Performing or Devolvement and Recovery.	Static	PC
BEHAVIOUR_SUB_TYPE_CD	Determines the sub type of the Behavior Pattern.	Static	PC
CUR_PAYMENT	Amount of current payment, meaning depends on amortization type code.	dynamic	P, PC
CUR_PAR_BAL	Balance on which principal runoff, interest cash flows, deferred runoff are based.	dynamic	P, PC, PP
CUR_NET_RATE	Interest rate that the financial institution pays/receives.	dynamic	P, R, PC, PP
CUR_GROSS_RATE	Interest rate that the customer pays/receives; used in determining payments and prepayments.	dynamic	P, R, PC, PP
DEFERRED_CUR_BAL	Holds current unamortized premium, discount, fees, costs, and so on.	dynamic	D
ISSUE_DATE	Date instrument is recognized as **on-the-books**. Used in dynamic gap and market value calculations.		I
INTEREST_RATE_CD	Code value that determines the forecasted rate to base repricing on.	static	R

Column Name	Column Description	Column Type	Event use
INT_TYPE	Determines how interest is calculated and accrued.	static	P
INSTRUMENT_TYPE_CD	Used to match a schedule instrument record to its scheduled payment dates and amounts.	static	I
LAST_PAYMENT_DATE	Date of last payment before the as-of-date, used to calculate days in first payment for interest in arrears instruments and to calculate accruals prior to first payment in interest in advance instruments.	static	P
LRD_BALANCE	Balance as of last reprice date.	static	I
LAST_REPRICE_DATE	Last date instrument rate repriced.	static	I
MARGIN	Pricing spread added to the IRC value for current net rate.	static	R
MATURITY_DATE	Date of final payment.	static	P, PP
NEG_AMRT_AMT	Amount of current balance due to negative amortization of interest payments.	dynamic	P
NEG_AMRT_EQ_DATE	Date that instrument fully re-amortizes, irrespective of payment caps.	event trigger	PC
NEG_AMRT_EQ_FREQ & NEG_AMRT_EQ_MULT	Frequency of neg am equalization events. 0 denotes neg-am equalization never occurs.	static	PC

Column Name	Column Description	Column Type	Event use
NEG_AMRT_LIMIT	Maximum amount that instrument can negatively amortize, stored as a percent of original balance.	event trigger	PC
NEXT_PAYMENT_DATE	Date of next payment.	event trigger	P, PC
NEXT_REPRICE_DATE	Date of next rate change.	event trigger	R, PP
ORG_PAYMENT_AMT	Payment used for cash flow transfer pricing of fixed rate records. Used by pattern instruments to calculate payment amount.	static	PC,I
ORG_PAR_BAL	Used in conjunction with neg am limit to determine the maximum amount that instrument can negatively amortize. Used for Rule of 78's schedules. Used by pattern instruments to calculate payment amount.	static	PC,I
ORG_TERM & ORG_ TERM_MULT	Time from origination date to maturity date. Used in determining whether an instrument balloons for payment recalculation purposes.	static	PC,PP
ORIGINATION_DATE	Determines age of instrument for prepayments. Used in calculating remaining amortization term. Used in determining payment number in pattern records.	static	PP,PC
PERCENT_SOLD	Determines net balance.	static	I

Column Name	Column Description	Column Type	Event use
PMT_ADJUST_DATE	Date of next scheduled payment recalculation for neg am instruments.	event trigger	PC
PMT_CHG_FREQ & PMT_CHG_FREQ_MULT	Frequency of regular payment change calculation for neg-am instruments only. 0 denote payment never changes.	static	PC
PMT_DECR_CYCLE	Maximum percent payment can decrease from its previous value.	static	PC
PMT_DECR_LIFE	Minimum payment amount; stored as a percent of original payment amount; can be overwritten on ngam equalization dates.	static	PC
PMT_FREQ & PMT_FREQ_MULT	Frequency of payments; should be set equal to original term if instrument is bullet (principal and interest at maturity date) or account type of other asset, other liability, interest income, interest expense, non-interest income, non-interest expense.	static	P, PC, PP
PMT_INCR_CYCLE	Maximum percent payment can increase from previous value.	static	PC
PMT_INCR_LIFE	Maximum payment amount; stored as a percent of original payment amount; can be overwritten on ngam equalization dates.	static	PC

Column Name	Column Description	Column Type	Event use
RATE_CAP_LIFE	Maximum value to which current rate can reprice.	static	R
RATE_CHG_MIN	Minimum amount that current rate must change before a rate change occurs.	static	R
RATE_CHG_RND_CD	Type of rounding to be applied to current rate.	static	R
RATE_CHG_RND_FAC	Precision of rounding; 0 denotes no rounding.	static	R
RATE_DECR_CYCLE	Maximum amount rate can decrease within a repricing period.	static	R
RATE_FLOOR_LIFE	Minimum value to which current rate can reprice.	static	R
RATE_INCR_CYCLE	Maximum amount rate can increase within a repricing period.	static	R
RATE_SET_LAG & RATE_SET_LAG_MULT	Time lag used when repricing. Used to determine rate set date on reprice event.	static	R
REMAIN_NO_PMTS_C	Number of payments left to be made on the instrument from the As of Date to the maturity date.	dynamic	P, PC
REPRICE_FREQ & REPRICE_FREQ_MULT	Frequency that instrument reprices; 0 denotes fixed rate.	static	R, PP
RESIDUAL_AMOUNT	For leases, indicates the residual amount.	Static	I,C

Column Name	Column Description	Column Type	Event use
ORG_PAYMENT_AMT	Payment used for cash flow transfer pricing of fixed rate records.	static	I
TEASER_END_DATE	Date that teased instrument begins repricing.	event trigger	R, PP
MARGIN_GROSS	Pricing spread added to IRC for current gross rate.	static	R
MARGIN_T_RATE	Pricing spread added to IRC for current transfer rate. Used in Balance Sheet Planning only.	static	R
TP_EFFECTIVE_DATE	Override date used by FTP to determine an alternate effective date for the historical TP Rate lookup.	static	I
T_RATE_INT_RATE_CD	Interest rate code used for determining transfer rate. Used in Balance Sheet Planning only.	Static	R
NET_MARGIN_CODE	Defines relationship between gross rate and net rate; 0 denotes floating net rate; 1 denotes constant net rate.	Static	R

Event Use Code Values

I = Initialization of record

P = Payment

PC = Payment Recalculation

PP = Prepayment

R = Reprice

D = Deferred amortization

Rule of 78's Example

Example: 12 month loan with current payment of $93.33 and original balance = $1,000.00

1. Sum all principal and interest payments made over the life of the instrument:

```
cash flow     = current payment * total number of payments
               = $93.33 * 12
               = $1,120.00
```

2. Determine total amount of interest paid over the life of the instrument:

```
interest       = cash flow - original par balance
                  = $1,120.00 - $1,000.00
                 = $120.00
```

3. Sum the payment numbers.

```
pmts           = total no. payments * (total no. payments + 1)/2
                  = 12 * 13/2
                  = 78
```

4. Calculate principal and interest amount at each payment.

```
interest        =  interest * (payments remaining/ pmts)
principal          = current payment - interest
```

Month	Interest Calculation	Interest	Principal	Remaining Balance
1	12/78 * 120	$18.46	$74.87	$925.13
2	11/78 * 120	$16.92	$76.41	$848.72
3	10/78 * 120	$15.38	$77.95	$770.77
4	9/78 * 120	$13.85	$79.48	$691.29
5	8/78 * 120	$12.31	$81.02	$610.27
6	7/78 * 120	$10.77	$82.56	$527.71
7	6/78 * 120	$9.23	$84.10	$443.61
8	5/78 * 120	$7.69	$85.64	$357.97
9	4/78 * 120	$6.15	$87.18	$270.79

Month	Interest Calculation	Interest	Principal	Remaining Balance
10	3/78 * 120	$4.61	$88.72	$182.07
11	2/78 * 120	$3.08	$90.25	$91.82
12	1/78 * 120	$1.54	$91.79	$0.00

6

Cash Flow Dictionary

This chapter includes a list of columns required for this processing as well as a list of columns required to run the Oracle Financial Services Analytical Applications (OFSAA) cash flow edits.

This chapter covers the following topics:

- Introduction to Cash Flow Dictionary
- Cash Flow Columns

Introduction to Cash Flow Dictionary

Cash flow processing is executed from Oracle Asset Liability Management (ALM) and Oracle Funds Transfer Pricing (FTP). This processing accesses specific fields from instrument tables to perform cash flow calculations.

The OFSAA cash flow edits are executed from either Oracle ALM or Oracle FTP and are used to correct data in the columns used in OFSAA cash flow processing.

> **Note:** Both lists overlap for a majority of the columns.

In order for OFSAA cash flow processing to generate appropriate results, the data within the accessed instrument tables must be appropriate and consistent. The OFSAA cash flow edits function provides a measure of validation for this data. However, the cash flow edits function cannot ensure that the input data from the instrument tables is correct and faithfully reflects reality. Therefore, this chapter contains detailed information necessary for correct data population, including field definitions, formulas used in the cash flow process calculations, and recommended default values for the cash flow processing fields.

Unless otherwise stated, when calculations refer to frequency or term fields in this chapter, the implication is that both the frequency (or term) and its associated multiplier field are used. For example, if PMT_FREQ is used in a formula, it refers to PMT_FREQ and PMT_FREQ_MULT to determine the true payment frequency.

Cash Flow Columns

This section describes in detail the cash flow processing and cash flow edit columns for OFSA.

Cash Flow Columns Listed

The OFSAA cash flow columns are listed in table format in this section along with information on whether they are associated with cash flow processing, cash flow edits or both.

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
ACCRUAL_BASIS_CD	Yes	Yes	Input
ACCRUED_INTEREST	Yes	Pending	Input
ADJUSTABLE_TYPE_CD	Yes	Yes	Input
ALL_IN_TP_RATE	Yes	No	Output
All Leaf Fields	Yes	No	Input/Output
AMORT_METH_PDFC_CD	Yes	Pending	Input
AMRT_TERM	Yes	Yes	Input
AMRT_TERM_MULT	Yes	Yes	Input
AMRT_TYPE_CD	Yes	Yes	Input
AS_OF_DATE	Yes	Yes	Input
AVERAGE_LIFE_C	Yes	No	Output
BASIS_RISK_CHARGE_CREDIT	Yes	No	Output
BASIS_RISK_COST_AMT	Yes	No	Output
BASIS_RISK_COST_RATE	Yes	No	Output

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
BEHAVIOUR_SUB_TYPE_CD	Yes	Pending	Input
BEHAVIOUR_TYPE_CD	Yes	Pending	Input
CHARGE_CREDIT_OCOST	Yes	No	Output
CHARGE_CREDIT_OCOST_REM_TERM	Yes	No	Output
CHARGE_CREDIT_TRATE	Yes	No	Output
CHARGE_CREDIT_TRATE_REM_TERM	Yes	No	Output
COMPOUND_BASIS_CD	Yes	Yes	Input
CONVEXITY_C	Yes	No	Output
CUR_BOOK_BAL	Yes	Yes	Input
CUR_GROSS_RATE	Yes	Yes	Input
CUR_NET_PAR_BAL_C	No	Yes	Input
CUR_NET_RATE	Yes	Yes	Input
CUR_OAS	Yes	No	Input/Output
CUR_PAR_BAL	Yes	Yes	Input
CUR_PAYMENT	Yes	Yes	Input
CUR_STATIC_SPREAD	Yes	No	Input
CUR_TP_PER_ADB	Yes	No	Input
CUR_YIELD	Yes	No	Output
DEFERRED_CUR_BAL	Yes	Yes	Input

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
DEFERRED_ORG_BAL	Yes	Yes	Input
DEVOLVEMENT_STATUS_CD	Yes	Pending	Input
DURATION_C	Yes	No	Output
EFF_INTEREST_RATE_C	Yes	No	Output
EXPECTED_BAL	Yes	Pending	Input
EXPECTED_BAL_GROWTH_PCT	Yes	Pending	Input
HISTORIC_OAS	Yes	No	Output
HISTORIC_STATIC_SPREAD	Yes	No	Output
HOLIDAY_ROLLING_CONVENTION_CD	Yes	Reserved for future use	Reserved for future use
ID_NUMBER	Yes	Yes	Input
IDENTITY_CODE	Yes	Yes	Input
INSTRUMENT_TYPE_CD	Yes	No	Input
INT_TYPE	Yes	Yes	Input
INTEREST_RATE_CD	Yes	Yes	Input
ISSUE_DATE	Yes	Yes	Input
LAST_PAYMENT_DATE	Yes	Yes	Input
LAST_REPRICE_DATE	Yes	Yes	Input
LIQUIDITY_PREM_CHARGE_CREDIT	Yes	No	Output
LIQUIDITY_PREMIUM_AMT	Yes	No	Ouput

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
LIQUIDITY_PREMIUM_RATE	Yes	No	Output
LRD_BALANCE	Yes	Yes	Input
MARGIN	Yes	No	Input
MARGIN_T_RATE	Yes	No	Input
MARGIN_GROSS	Yes	No	Input
MARKET_VALUE_C	Yes	No	Output
MARKET_VALUE_CLEAN_C	Yes	No	Output
MATCHED_SPREAD_ALT	Yes	No	Output
MATCHED_SPREAD_C	Yes	No	Input
MATURITY_AMOUNT	Yes	Pending	Input
MATURITY_DATE	Yes	Yes	Input
MINIMUM_BALANCE	Yes	Pending	Input
MODIFIED_DURATION_C	Yes	No	Output
NEG_AMRT_AMT	Yes	No	Output
NEG_AMRT_EQ_DATE	Yes	Yes	Input
NEG_AMRT_EQ_FREQ	Yes	Yes	Input
NEG_AMRT_EQ_MULT	Yes	Yes	Input
NEG_AMRT_LIMIT	Yes	Yes	Input
NET_MARGIN_CD	Yes	Yes	Input
NEXT_PAYMENT_DATE	Yes	Yes	Input

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
NEXT_REPRICE_DATE	Yes	Yes	Input
OFFSET_PERCENT	Yes	Pending	Input
ORG_BOOK_BAL	No	Yes	Input
ORG_MARKET_VALUE	Yes	No	Input
ORG_PAR_BAL	Yes	Yes	Input
ORG_PAYMENT_AMT	Yes	Yes	Input
ORG_TERM	Yes	Yes	Input
ORG_TERM_MULT	Yes	Yes	Input
ORIGINATION_DATE	Yes	Yes	Input
OTHER_ADJ_AMOUNT_ALT	Yes	No	Output
OTHER_ADJ_CHARGE_CREDIT	Yes	No	Output
OTHER_ADJ_RATE_ALT	Yes	No	Output
OTHER_ADJUSTMENTS_AMT	Yes	No	Output
OTHER_ADJUSTMENTS_RATE	Yes	No	Output
PERCENT_SOLD	Yes	Yes	Input
PMT_ADJUST_DATE	Yes	Yes	Input
PMT_CHG_FREQ	Yes	Yes	Input
PMT_CHG_FREQ_MULT	Yes	Yes	Input
PMT_DECR_CYCLE	Yes	Yes	Input

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
PMT_DECR_LIFE	Yes	Yes	Input
PMT_FREQ	Yes	Yes	Input
PMT_FREQ_MULT	Yes	Yes	Input
PMT_INCR_CYCLE	Yes	Yes	Input
PMT_INCR_LIFE	Yes	Yes	Input
PRICING_INC_CHARGE_CREDIT	Yes	No	Output
PRICING_INCENTIVE_AMT	Yes	No	Output
PRICING_INCENTIVE_RATE	Yes	No	Output
PRIOR_TP_PER_ADB	Yes	No	Input
PMT_SET_LAG	No	No	Input
PMT_SET_LAG_MULT	No	No	Input
RATE_CAP_LIFE	Yes	Yes	Input
RATE_CHG_MIN	Yes	Yes	Input
RATE_CHG_RND_CD	Yes	Yes	Input
RATE_CHG_RND_FAC	Yes	Yes	Input
RATE_DECR_CYCLE	Yes	Yes	Input
RATE_FLOOR_LIFE	Yes	Yes	Input
RATE_INCR_CYCLE	Yes	Yes	Input
RATE_SET_LAG	Yes	Yes	Input

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
RATE_SET_LAG_MULT	Yes	Yes	Input
RESIDUAL_AMOUNT	Yes	Pending	Input
REMAIN_NO_PMTS_C	Yes	Yes	Input
REMAIN_TERM_MULT_C	No	No	Input
REPRICE_FREQ	Yes	Yes	Input
REPRICE_FREQ_MULT	Yes	Yes	Input
TEASER_END_DATE	Yes	Yes	Input
TP_AVERAGE_LIFE	Yes	No	Input/Output
TP_DURATION	Yes	No	Input/Output
TP_EFFECTIVE_DATE	Yes	Pending	Input
TRAN_RATE_REM_TERM	Yes	No	Output
TRAN_RATE_REM_TERM_ALT	Yes	No	Output
TRANSFER_RATE	Yes	No	Input/Output
TRANSFER_RATE_ALT	Yes	No	Output
DERIVATIVE_ONLY_COLUMNS			
BINARY_RATE	Yes	Pending	Input
EXCHG_OF_PRINCIPAL	Yes	Pending	Input
INSTRUMENT_TYPE_CD	Yes	Pending	Input
LEG_TYPE	Yes	Pending	Input

Column Name	Cash Flow Processing	Cash Flow Edits	Usage
MIN_BAL_AMORT	Yes	Pending	Input
PURCHASE_SALE_LOGIC	Yes	Pending	Input
SWAP_CLASS_CD	Yes	Pending	Input

Field Definitions

This section describes, in detail, the usage of these columns. For each column, the following information is provided:

- Column name as it appears in the appendix (upper and lower case) and as it appears in the database (upper case with underscores).
- Which OFSAA products are affected (ALM or FTP)?
- Data verification requirements and suggested defaults.

Accrual Basis Code (ACCRUAL_BASIS_CD)

Definition

The basis on which the interest accrual is calculated.

Module Usage

Oracle ALM and Oracle FTP cash flow methodologies use ACCRUAL_BASIS_CD for calculating interest income (financial element 430).

The accrual basis values are represented by code values as follows.

Code Value	Accrual Basis
1	30/360
2	Actual/360
3	Actual/Actual

Code Value	Accrual Basis
4	30/365
5	30/Actual
6	Actual/365

Oracle ALM and Oracle FTP cash flow methodologies reference INT_TYPE in determining whether interest payments are made in arrears or in advance.

If INT_TYPE = 1, the record is considered **interest in arrears**. Interest payments are paid at the end of the payment period along with the principal payments. The following calculations assume that the interest is to be calculated as interest in arrears.

If INT_TYPE = 2, the record is considered **interest in advance**. See Interest Type Code (INT_TYPE), page 6-59 for a description of the formula used to calculate interest in advance.

For calculation purposes, the accrual basis codes can be grouped in the following manner.

> **Note:** The following calculations assume a monthly payment frequency.

1. If the ACCRUAL_BASIS_CD is 30/360, 30/365 or 30/Actual, OFSAA uses the following formula to calculate interest income on a payment date:

   Previous Period's Ending Balance * Cur Net Rate/100 * PMT_FREQ [number of months] * x * (Next Payment Date - Last Payment Date)/(Next Payment Date - Calculated Last Payment Date)

   Replace **x** with one of the three accrual basis values mentioned earlier.

   > **Note:** The actual denominator refers to the actual number of days in the year. Other than leap years, this equals 365 days. Also note that the Calculated Last Payment Date is the Next Payment Date rolled back by the number of months in PMT_ FREQ.

   The final portion of the earlier calculation,

   (Next Payment Date - Last Payment Date)/(Next Payment Date - Calculated Last Payment Date)

   is a ratio that calculates the percentage of the payment frequency period that should be applied to calculate the Interest Income amount. This adjustment is necessary because, for the first forecasted payment, the Last Payment Date is not necessarily equal to the Calculated Last Payment Date. This would be the case of a stub or extended payment at the origination or maturity of a record.

If the Last Payment Date is precisely equal to the Calculated Last Payment Date, then the ratio is equal to 1 and therefore does not impact the Interest Income calculation.

2. If the ACCRUAL_BASIS_CD is Actual/365, Actual/Actual, or Actual/360, OFSAA uses the following formula to calculate interest income on a payment date:

   Previous Period's Ending Balance * Cur Net Rate/100 * (Next Payment Date - Last Payment Date)/y

   Replace **y** with the denominator of one of the three accrual basis values mentioned earlier.

   > **Note:** The actual numerator refers to the actual number of days in the current month.

The preceding two equations represent Interest in Arrears income calculations. The interest in advance calculations are indicated in the INT_TYPE section.

> **Note:** If a compounding method has been chosen, OFSA derives the compounded rate before calculating the preceding interest income amounts. For further information, see Compounding Basis Code (COMPOUND_BASIS_ CD), page 6-30.

Data Verification Requirements and Suggested Defaults

- Must be equal to values 1 - 6.
- Suggested default depends on product characteristics of institution's data. Default to the most common ACCRUAL_BASIS_CD for the product leaf.
- If AMRT_TYPE_CD = 800, 801, or 802 (Schedule) or 1000 to 99999 (Pattern), the ACCRUAL_BASIS_CD cannot equal **1**, **4** or **5**.

Accrued Interest (ACCRUED_INTEREST)

Definition

Calculated interest due from the last payment date to the as of date.

Module Usage

For Oracle ALM, this field is used with Non-Maturity Behavior Pattern instruments to compute the first interest cash flow on the next_payment_date. If this information is available from the source system, the cash flow engine will be able to more accurately reflect the interest cash flow due on the next interest cash flow date (NEXT_PAYMENT_DATE). The cash flow engine combines the ACCRUED_INTEREST amount (up to the AS_OF_DATE) with a calculated interest from the AS_OF_DATE + 1day to the NEXT_PAYMENT_DATE.

> **Note:** For Non-Maturity Behavior Patterns, the principal payments are determined based on the pattern dates, and the interest payments are determined based on the NEXT_PAYMENT_DATE and PMT_FREQ from the instrument record.

Data Verification Requirements and Suggested Defaults

- Should be > 0 for non-maturity behavior pattern instruments.
- For non-maturity behavior pattern instruments, set LAST_PAYMENT_DATE = AS_OF_DATE.
- For non-maturity behavior pattern instruments, set NEXT_PAYMENT_DATE equal to the next expected interest payment date.
- For non-maturity behavior pattern instruments, set PMT_FREQ equal to the interest payment frequency.

Adjustable Type Code (ADJUSTABLE_TYPE_CD)

Definition

Identifies the repricing method and repricing characteristics of the record.

Module Usage

For Oracle ALM, this field works in conjunction with REPRICE_FREQ to determine the repricing characteristics of an instrument. An ADJUSTABLE_TYPE_CD must be specified if the record is expected to reprice.

Oracle ALM

1. The code values for this field are as follows:

Code Value	Definition	Repricing Frequency	Repricing Method
000	Fixed	0	No Repricing

Code Value	Definition	Repricing Frequency	Repricing Method
030	Administered Rate	> 0	Reprices when IRC (interest rate code) changes.*
050	Floating Rate	> 0	Reprices when IRC (interest rate code) changes.*
250	Adjustable	> 0	Last Reprice Date + Reprice Frequency.*
500-4999	Reprice Pattern	> 0	Reprices based on pattern definition

`*(if not in tease period)`

1. If the ADJUSTABLE_TYPE_CD = 0 and the REPRICE_FREQ = 0, then the record is fixed-rate.

2. If the ADJUSTABLE_TYPE_CD = 30 or 50 and the REPRICE_FREQ > 0, then the reprice dates are driven by forecasted yield curve rate changes rather than by the REPRICE_FREQ. For these codes, Oracle ALM reprices the record by referencing the Forecast Rate Assumption - interest rate code (IRC) when producing cash flow information at the beginning of each bucket. There is one reference to the IRC per modeling bucket.

   The database field, NEXT_REPRICE_DATE, is not used when the ADJUSTABLE_TYPE_CD = 30 or 50. The database field, REPRICE_FREQ, is used to determine the yield curve point when the IRC is a yield curve as opposed to a single rate IRC.

   > **Note:** Floating/Administered ADJUSTABLE_TYPE_CD should not be used for instruments with periodic caps and/or floors because periodic caps and floors infer a specific repricing frequency.

3. If the ADJUSTABLE_TYPE_CD = 250 and the REPRICE_FREQ > 0, then the repricing frequency of the record is determined by the REPRICE_FREQ and NEXT_REPRICE_DATE. See these fields for further explanations of the repricing process.

2. The value input into the ADJUSTABLE_TYPE_CD overrides the REPRICE_ FREQ

value. For instance, even though the REPRICE_FREQ > 0 and if the ADJUSTABLE_TYPE_CD = 0, OFSAA treats the record as a fixed-rate instrument.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing references REPRICE_FREQ to determine if the record is an adjustable rate instrument.

Data Verification Requirements and Suggested Defaults

- If the record is a floating rate or administered rate product, ADJUSTABLE_TYPE_CD = 30 or 50.
- If the record reprices contractually according to NEXT_REPRICE_DATE and REPRICE_FREQ information, ADJUSTABLE_TYPE_CD = 250.
- If the ADJUSTABLE_TYPE_CD > 0, REPRICE_FREQ > 0.
- If the record is fixed, ADJUSTABLE_TYPE_CD = 0.

All - in Transfer Rate (ALL_IN_TP_RATE)

Definition

The Transfer Pricing Engine outputs the All-in transfer rate that includes the base transfer pricing (TP) rate plus any defined transfer rate adjustments. The components of the All-in TP rate are defined in the Standard Transfer Pricing Process, Calculation Elements block.

Module Usage

This field is populated (optionally) by the FTP Process and represents the total of the base transfer rate plus any standard transfer rate adjustments.

Data Verification Requirements and Suggested Defaults

None

Amortization Method for Premiums and Discounts (AMORT_METH_PDFC_CD)

Definition

Code to indicate how the cash flow engine should amortize (or accrete) a discount or premium amount. The amount considered for amortization on the instrument record is defined by the DEFERRED_CUR_BAL.

Module Usage

Premium and discount instruments use this field to indicate the method to amortize/accrete. Values are: 0=Level Yield, 1=Straight Line. This input is read by the cash flow engine when the DEFERRED_CUR_BAL is > 0. See DEFERRED_CUR_BAL, page 6-45 for a description of the Level Yield calculation.

Data Verification Requirements and Suggested Defaults

- If the DEFERRED_CUR_BAL < > 0 then AMORT_METH_PDFC_CD must be populated with 0 (default) or 1.

Amortization Type Code (AMRT_TYPE_CD)

Definition

Defines the method by which an account's principal and interest is amortized.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies use AMRT_TYPE_CD to determine the calculation method of the record's amortization of principal and resulting calculation of interest. Following are the supported AMRT_TYPE code values.

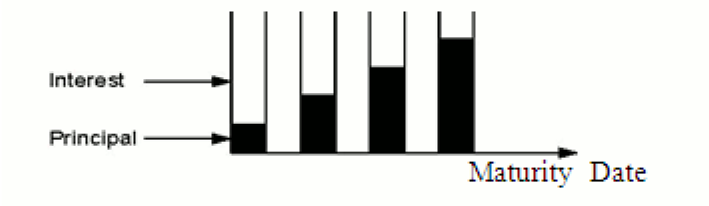
Conventionally Amortizing	
100	Conventional Fixed
400	Balloon
500	Adjustable Conventional
600	Adjustable Negative Amortizing
840	Lease

Non-Conventionally Amortizing	
700	Simple Interest

Non-Conventionally Amortizing	
710	Rule of 78s
800 - 802	Payment Schedules
820	Level Principal
850	Annuity
999	Default Value
1000 - 69999	User-Defined Payment Patterns

Following are the explanations of each of the amortization type codes.

100, 400, 500, 600, 840 - Conventionally Amortizing AMRT_TYPE_CDs



The conventional amortization loan types have loan payments that are unevenly divided between principal balance and interest owed. Total payment amount (principal + interest) is generally equal throughout the life of the loan. The interest portion (non-shaded portion) of each payment is calculated based on the record's interest rate and the remaining balance of the loan. Therefore, close to the loan's origination, a higher portion of the payment consists of interest rather than principal. As the loan is paid down, an increasing portion of each payment is allocated to principal until a zero balance is reached at maturity.

> **Note:** **For Lease amortization, the principal is reduced to the amount defined in the RESIDUAL_AMOUNT column rather than zero.**

For these five AMRT_TYPE_CDs, the amount in the CUR_PAYMENT field should equal principal plus interest.

Following is a breakout of these four conventionally amortizing AMRT_TYPE_CDs:

Conventional Loan Type	Description
100 Conventional Fixed	(Described earlier)
400 Balloon	A loan in which the amortization term (AMRT_ TERM) of an instrument exceeds the maturity term (ORG_TERM). For example, a loan with an original term of seven years is amortized conventionally as if it were a 30-year instrument. At the end of the 7th year, there exists a large balloon payment that represents 23 years of non-amortized loan balance.
500 Conventional Adjustable	Repricing instrument with conventional amortization.
600 Adjustable Neg Am	In a negatively amortizing instrument, the principal of a loan increases when the loan payments are insufficient to pay the interest due. The unpaid interest is added to the outstanding loan balance, causing the principal to increase rather than decrease, as payments are made. For more details, see NEG_AMRT_AMT, page 6-74.
840 Lease	A lease instrument amortizes down to the amount specified in the RESIDUAL_AMOUNT column. The CUR_PAYMENT for a fixed rate lease should be entered so the instrument will achieve the RESIDUAL_AMOUNT on the MATURITY_DATE of the lease contract. For an adjustable rate lease, the CUR_PAYMENT will be re-calculated by the cash flow engine considering the MATURITY_DATE , RESIDUAL_AMOUNT and current rate.

The OFSAA cash flow engine does not treat AMRT_TYPE_CD 100, 400 or 500 differently. For a given record, the use of any of these three types produces identical results. The division is simply for product distinction purposes. For instance, AMRT_TYPE_CD 100 can be used for a fixed-rate, adjustable-rate or a balloon record. However, only an AMRT_ TYPE 600 record uses the negative amortization fields.

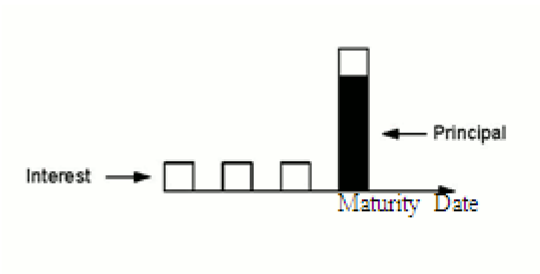
The cash flow engine does not use AMRT_TYPE_CD to identify whether a record is adjustable or not. It instead uses REPRICE_FREQ (and in Oracle ALM, ADJUSTABLE_TYPE_CD) for this purpose. Therefore, any amortization type can be adjustable.

The cash flow engine does not use AMRT_TYPE_CD to determine whether a record is a balloon or not. It instead uses AMRT_TERM and ORG_TERM for this purpose. Therefore, even a level principal AMRT_TYPE_CD could be treated as a balloon instrument.

A record must be AMRT_TYPE_CD 600 in order for the cash flow engine to process the record using the negative amortization fields.

A record must be AMRT_TYPE_CD 840 in order for the cash flow engine to process the record using the residual amount field.

700 Simple Interest (Non-amortizing)



For simple interest amortization type, no principal is paid until maturity. If NEXT_PAYMENT_DATE < MATURITY_DATE, OFSA calculates interim interest-only payments as shown in the earlier diagram. OFSA pays the entire record's principal balance on the maturity date along with the appropriate interest amount.

For this AMRT_TYPE_CD, the CUR_PAYMENT field should equal **0**.

710: Rule of 78s

An amortization type in which the following calculation is used in computing the interest rebated when a borrower pays off a loan before maturity.

For example, in a 12-month loan, the total is 78 (1 + 2 + ... 12 = 78). For the first month, 12/78 of the total interest is due. In the second month, this amount is 11/78 of the total.

For Rule of 78 AMRT_TYPE_CDs, the amount in the CUR_PAYMENT field should equal principal plus interest. For more details, see Cash Flow Calculations, page 5-1.

800 - 802: Payment Schedule

The key for matching the instrument record with its corresponding payment schedule record is the INSTRUMENT_TYPE_CD and ID_NUMBER.

800: Conventional Payment Schedule This AMRT_TYPE_CD conventionally amortizes a record whose cash flows are defined in the PAYMENT_SCHEDULE table. Payment amounts should contain both principal + interest.

801: Level Principal Payment Schedule This AMRT_TYPE_CD level principal amortizes a record whose cash flows are defined in the PAYMENT_SCHEDULE table. Payment amounts should contain principal only.

802: Simple interest Payment Schedule This AMRT_TYPE_CD simple interest amortizes a record whose cash flow dates are defined in the PAYMENT_ SCHEDULE table. Payment amounts should equal 0 (the engine calculates interest and ignores payment amount in the schedule records). Principal balance is paid on the maturity date as defined in the schedule.

820 Level Principal Payments

Level principal payment is the amortization type in which the principal portion of the loan payment remains constant for the life of loan. Interest (non-shaded portion) is calculated as a percentage of the remaining balance, and therefore, the interest portion decreases as the maturity date nears. Because the principal portion of payment is constant for life, the total payment amount (principal plus interest) decreases as the loan approaches maturity.

For this AMRT_TYPE_CD, the amount in the CUR_PAYMENT field should equal the principal portion only.

850 Annuities

The Annuity amortization type assumes the payment amount (CUR_PAYMENT) will increase the principal balance rather than reduce it.

For this AMRT_TYPE_CD, the amount in the CUR_PAYMENT field should equal the amount of the regular / scheduled principal contribution. The CUR_PAYMENT should be expressed as a positive value and the cash flow engine will multiply by -1 to reflect the principal contribution.

Users can also, optionally populate the MATURITY_AMOUNT column. If populated, this amount will be considered as the target amount and required periodic payment will be calculated to achieve the Maturity Amount at the maturity of the instrument.

1000 - 69999: User-Defined Payment Patterns

Records with this range of AMRT_TYPE_CDs are matched to the user-defined amortizations in the Payment Pattern user interface. The key for matching is the AMRT_TYPE_CD value. These records can only be defined as conventionally amortizing, level-principal, or simple interest.

999: Other Amortization Type

If this value is used, the cash flow engine uses the AMRT_TYPE_CD 700, simple interest amortization method. Using this AMRT_TYPE_CD for product identification purposes is not recommended. This should be reserved for indicating erroneous data extraction.

For calculation methods of the different AMRT_TYPE_CDs, see Current Payment (CUR_PAYMENT), page 6-37.

Data Verification Requirements and Suggested Defaults

- All accounts require a valid AMRT_TYPE_CD as listed earlier.
- If the AMRT TYPE <> 700 or is not a simple interest-amortizing schedule or pattern record, CUR_PAYMENT must be valid. If CUR_PAYMENT value is too small or equal to **0**, the cash flow engine may generate erroneous cash flows, depending on the AMRT_TYPE_CD selected.

- If the record is defaulted to AMRT_TYPE_CD 999, or if it cannot find a match in PAYMENT_SCHEDULE or PAYMENT_PATTERNS, the cash flow engine processes the record as AMRT_TYPE_CD 700.
- Suggested defaults in the following table are dependent on basic knowledge of product characteristics:

Loan Type	Suggested AMRT_TYPE_CD
Non-amortizing, such as Certificates of Deposit	700
Fixed amortizing, such as short term consumer loans	100
Variable rate amortizing, such as adjustable-rate mortgages	500

- Rule of 78s instruments are implicitly fixed.

  If AMRT_TYPE_CD = 710, REPRICE_FREQ = 0.

Amortization Term (AMRT_TERM)

Definition

Amortization term is used in conjunction with AMRT_TERM_MULT to define the term over which the payment is amortized.

Module Usage

AMRT_TERM is used by Oracle ALM and Funds Transfer Pricing for adjust-able-rate cash flow transfer-priced records.

Amortization Term

Amortization term has two main purposes:

- Identifies whether a record is a balloon and is used in calculation of payment amounts.
- Used when recalculating payment amounts for User-Defined Payment Pattern records that are defined as % Current Payment and have more than one payment frequency defined in the Payment Pattern user interface.

Balloon Check

As an initial step before processing a record's cash flows, the cash flow engine compares the record's ORG_TERM with its AMRT_TERM. If AMRT_TERM = ORG_TERM. The cash flow engine then uses the CUR_PAYMENT from the record. When appropriate, the cash flow engine later recalculates the CUR_PAYMENT if: 1) The record reprices; 2) The TEASER_END_ DATE is reached; or 3) A negative amortization-related recalculation date is reached.

If the AMRT_TERM > ORG_TERM, the cash flow engine recognizes the record as a balloon, and recalculates the payment amount. In order to perform this calculation (For the formula, see Current Payment (CUR_PAYMENT, page 6-37)), the cash flow engine must derive the remaining number of payments until the end of the amortization term. This is calculated by adding the AMRT_TERM to the ORIGINATION_DATE to determine the amortization end date. The remaining number of payments is calculated by determining how many payments can be made from and including the NEXT_PAYMENT_DATE and this date. Following is the formula used for determining the remaining number of payments:

(((ORIGINATION_DATE - Beginning of Payment Period Date) / 30.41667 + AMRT_TERM) / PMT_FREQ)

Beginning of Payment Period Date refers to the date of the current payment that the cash flow engine is calculating. This would equal the detail record's NEXT_PAYMENT DATE if the engine were calculating the first forecasted cash flow. After the remaining number of payments have been calculated, the cash flow engine derives the CUR_PAYMENT amount and applies it to the record's cash flows. Ignoring repricings and other recalculation events, this payment amount is paid until maturity, at which time the cash flow engine pays the record's balloon payment (the remaining principal portion).

User-Defined Pattern

Records that are defined as **% Current Payment** in the User-Defined Payment Pattern screen and have more than one payment frequency defined in the Payment Pattern user interface also recalculate the payment amount using the earlier formula.

The remaining number of payments on pattern records is calculated by rounding to the nearest number of payments when the remaining term is not exactly divisible by the payment frequency.

Data Verification Requirements and Suggested Defaults

- If the record is a balloon, AMRT_TERM > ORG_TERM.
- If the record is not a balloon, AMRT_TERM = ORG_TERM.
- AMRT_TERM should never be less than ORG_TERM.
- Do not default AMRT_TERM or ORG_TERM to **0**. Use **1**.
- The validation of AMRT_TERM should always be done in conjunction with AMRT_TERM_MULT.

Amortization Term Multiplier (AMRT_TERM_MULT)

Definition

Used in conjunction with AMRT_TERM to define the term over which the payment is amortized.

Module Usage

This field is the multiplier of the AMRT_TERM field. It is used in conjunction with AMRT_TERM to define the term over which the payment is amortized. Oracle ALM and Funds Transfer Pricing cash flow transfer-priced records reference AMRT_TERM_MULT when recalculating the current payment as defined under the AMRT_TERM section. AMRT_TERM_MULT determines the units (Months, Days or Years) of AMRT_TERM.

Data Verification Requirements and Suggested Defaults

Values are:

D= Days

M= Months

Y= Years

As of Date (AS_OF_DATE)

Definition

The date that the extracted data represents.

Module Usage

AS_OF_DATE is used for the following purposes:

- Application Preferences filter (Oracle ALM and Oracle FTP)
- Market value calculations (Oracle ALM)
  - **t** calculation
  - Term of cash flow from AS_OF_DATE is used for matching cash flow for discounting purposes.
- Payment Schedules and Patterns - used to determine where in the life of the loan the record is.

- Transfer Pricing Remaining Term Pricing Basis

Application Preferences

Oracle ALM and Oracle FTP use AS_OF_DATE as a primary data filter. When executing an ALM or Transfer Pricing processing run, the engines compare the AS_OF_DATE in Application Preferences, against the AS_ OF_DATE field of the detail instrument record. If AS_OF_DATE from the instrument record is the same date as that from Application Preferences, then engines process the instrument record. Otherwise, engines do not process the instrument record.

Market Value Calculations

- When calculating the market valuation of a daily paying record (PMT_FREQ_MULT = **D**), Oracle ALM uses the AS_OF_DATE when calculating the **t** period variable. For calculations, see Market Valuation Calculation, page 5-50.

- For the Oracle ALM Discount Methods - Spot Interest Rate Code and Forecast Remaining Term methods, the AS_OF_DATE is used in calculating the cash flow's discounting term. The cash flow date (payment date) is subtracted from the AS_OF_DATE to determine this term. This term is then applied to the appropriate discount yield curve in the Oracle ALM Forecast Rates assumption rule to determine the discount rate for the record (after applying the indicated interpolation method).

Payment Schedules and User-Defined Payment Patterns

- **Relative Patterns** - AS_OF_DATE is used to determine where in the life of an Payment Pattern a record is currently positioned. For relative patterns, the Payment Frequencies specified in the Pattern interface are rolled forward from the ORIGINATION_DATE until the rolled date is greater than the AS_OF_ DATE.

- **Absolute Patterns** - In determining the first forecasted payment date, the cash flow engine selects the payment date in the pattern that corresponds to the first date after the AS_OF_DATE.

  > **Note:** If NEXT_PAYMENT_DATE is different from the next defined payment date in the absolute pattern, the NEXT_ PAYMENT_DATE is used instead. Therefore, the cash flow engine requires that the NEXT_PAYMENT_DATE correspond to the appropriate date in the absolute payment pattern. For additional information, see Next Payment Date (NEXT_ PAYMENT_DATE), page 6-81.

- **Payment Schedules –** the cash flow engine makes the first forecasted payment based on the first date in the payment schedule table after the AS_OF_DATE.

Transfer Pricing Remaining Term Pricing Basis

When the Remaining Term Pricing Basis is selected for Transfer Pricing, transfer rates for the relevant methodologies are calculated from the AS_OF_DATE.

Data Verification Requirements and Suggested Defaults

Unless the record has a future ORIGINATION_DATE, the following conditions exist:

- AS_OF_DATE = MATURITY_DATE – REMAIN_TERM_C
- AS_OF_DATE < NEXT_REPRICE_DATE
- AS_OF_DATE < NEXT_PAYMENT_DATE
- AS_OF_DATE < MATURITY_DATE
- AS_OF_DATE >= ORIGINATION_DATE
- AS_OF_DATE >= ISSUE_DATE
- AS_OF_DATE < PMT_ADJUST_DATE
- AS_OF_DATE >= LAST_PAYMENT_DATE
- AS_OF_DATE >= LAST_REPRICE_DATE

Average Life Calculated (AVERAGE_LIFE_C)

Definition

The average life of a loan is the average number of years that principal is outstanding. The calculated amount represents the time necessary for principal to be reduced by one half of its current value.

Module Usage

This financial measure is calculated by Oracle ALM when the corresponding Calculation Element is selected and the result is written to FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER. Users can additionally update each individual instrument record with the calculation result by selecting the related measure in Output Preferences.

Oracle FTP computes the Average Life when the Average Life TP Method is applied to an instrument record. With this TP Method, the user has the additional option to update the instrument record with the calculation results (TP_AVERAGE_LIFE). For more information on the Average Life TP Method, see the *Oracle Financial Services Funds Transfer Pricing User Guide*.

The cash flow engine calculates the average life using the following formula:

$$Average\ Life = \sum_{i=1}^{n} \frac{P_i}{P}\ t_i$$

Where:

P is the principal

P_i is the principal repayment in coupon i, hence,

$$\frac{P_i}{P}$$

is the fraction of the principal repaid in coupon i and t_i is the time from the start of coupon *i*.

Data Verification Requirements and Suggested Defaults

None

Basis Risk Charge Credit (BASIS_RISK_CHARGE_CREDIT)

Definition

Oracle FTP calculates the funds charge or credit due to basis risk when the user selects the Adjustments, Charge/Credit option in the TP Process – Calculation Elements block.

Module Usage

This field is populated by the FTP process (optionally) if selected in the Calculation Elements block. The TP Engine calculates this amount as follows:

AVG_BOOK_BAL or CUR_BOOK_BAL X BASIS_RISK_COST_RATE/100 X ACCRUAL BASIS

> **Note:** The balance referenced in the preceding calculation is selected in the FTP Application Preferences screen.

Data Verification Requirements and Suggested Defaults

None

Basis Risk Cost Amount (BASIS_RISK_COST_AMT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Basis Risk Cost amount in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the basis risk adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the amount input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Basis Risk Cost Rate (BASIS_RISK_COST_RATE)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Basis Risk Cost rate or defines a formula based rate in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the basis risk adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the assumption input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Behavior Type Code (BEHAVIOUR_TYPE_CODE)

Definition

Behavior Type indicates the type of behavior pattern that is mapped to the instrument records.

Module Usage

Oracle ALM uses BEHAVIOUR_TYPE_CD to determine the type of behavior pattern being applied to the instrument record. Following are the supported BEHAVIOUR_TYPE_CODE values.

Code Value	Behavior Type
1	Non-Maturity
2	Non-Performing
3	Devolvement and Recovery

Data Verification Requirements and Suggested Defaults

- BEHAVIOR_TYPE_CD is mandatory when AMRT_TYPE_CD is populated with a Behavior Pattern code. Example: AMRT_TYPE_CD >= 70000 and <=99999.
- If AMRT_TYPE_CD is not in the Behavior Pattern range, then BEHAVIOR_TYPE_CD can be defaulted to NULL.

Behavior Sub Type Code (BEHAVIOUR_SUB_TYPE_CODE)

Definition

This field indicates the type of cash flow associated with Non-Performing or Devolvement and Recovery Behavior Pattern instruments.

Module Usage

Oracle ALM uses BEHAVIOUR_SUB_TYPE_CD to determine the SUB type of the behavior pattern being applied to the instrument record. Following are the supported BEHAVIOUR_SUB_TYPE_CODE values.

Code Value	Behavior Sub Type
201	Substandard
202	Doubtful
203	Loss
301	Sight Devolvement

Code Value	Behavior Sub Type
302	Sight Recovery
303	Usance Devolvement
304	Usance Recovery

Data Verification Requirements and Suggested Defaults

- BEHAVIOR_SUB_TYPE_CD is mandatory when AMRT_TYPE_CD is populated with a Behavior Pattern code. Example, AMRT_TYPE_CD >= 70000 and <=99999 AND the BEHAVIOR_TYPE_CD is 2 (Non-Performing) or 3 (Devolvement and Recovery).
- BEHAVIOR_SUB_TYPE_CD can be NULL when BEHAVIOR_TYPE_CD = 1 (Non-Maturity) or when the AMRT_TYPE_CD is not in the earlier mentioned Behavior Pattern Range.

Charge Credit Option Cost (CHARGE_CREDIT_OCOST)

Definition

This field is populated by the FTP Stochastic Process and holds the charge / credit amount for the original term Option Cost.

Module Usage

Oracle Funds Transfer Pricing populates this field (optionally), when the user selects the Option Cost – Charge / Credit option in the Stochastic FTP Process.

AVG_BOOK_BAL or CUR_BOOK_BAL X (HISTORIC_OAS – HISTORIC_STATIC_SPREAD)/100 X ACCRUAL BASIS

Data Verification Requirements and Suggested Defaults

None

Charge Credit Option Cost Remaining Term (CHARGE_CREDIT_OCOST_REM_TERM)

Definition

This field is populated by the FTP Stochastic Process and holds the charge / credit amount for the remaining term Option Cost.

Module Usage

Oracle Funds Transfer Pricing populates this field (optionally), when the user selects the Option Cost – Charge / Credit option in the Stochastic FTP Process.

AVG_BOOK_BAL or CUR_BOOK_BAL X (CUR_OAS – CUR_STATIC_SPREAD)/100 X ACCRUAL BASIS

Data Verification Requirements and Suggested Defaults

None

Charge Credit Transfer Rate (CHARGE_CREDIT_TRATE)

Definition

This field is populated by the FTP Standard Process and holds the charge / credit amount for the original term Transfer Rate Charge or Credit.

Module Usage

Oracle Funds Transfer Pricing populates this field (optionally), when the user selects the Transfer Rate – Charge / Credit option in the Standard FTP Process.

AVG_BOOK_BAL or CUR_BOOK_BAL X TRANSFER_RATE / 100 X ACCRUAL BASIS

Data Verification Requirements and Suggested Defaults

None

Charge Credit Transfer Rate Remaining Term (CHARGE_CREDIT_TRATE_REM_TERM)

Definition

This field is populated by the FTP Standard Process and holds the charge / credit amount for the remaining term Transfer Rate Charge or Credit.

Module Usage

Oracle Funds Transfer Pricing populates this field (optionally), when the user selects the Transfer Rate – Charge / Credit option in the Standard FTP Process.

AVG_BOOK_BAL or CUR_BOOK_BAL X TRAN_RATE_REM_TERM / 100 X ACCRUAL BASIS

Data Verification Requirements and Suggested Defaults

None

Compounding Basis Code (COMPOUND_BASIS_CD)

Definition

Indicates the compounding frequency used to calculate interest income.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow calculations reference the COMPOUND_BASIS_CD field when determining the detail record's compounding method to be applied during interest income (financial element 430) calculations.

1. The following table shows the code values for the COMPOUND_BASIS_CD and the interest calculation logic for an annual-paying instrument with 30/360 accrual basis code.

Code Value	Description	Annual Payment Calculation
110	Daily	Balance * [(1 + Rate/365)^365-1]
120	Monthly	Balance * [(1 + Rate/12)^12-1]
130	Quarterly	Balance * [(1 + Rate/4)^4-1]
140	Semi-annual	Balance * [(1 + Rate/2)^2-1]
150	Annual	Balance * [(1 + Rate/1)^1-1]
160	Simple	Balance * Rate (no compounding)
170	Continuous	$e^{(\text{Rate Per Payment})} -1$
200	At Maturity	Balance * Rate (no compounding)
999	Other	Balance * Rate (no compounding)

   The annualized rate that is applied to the record for interest income calculations is compounded according to one of the methods listed earlier.

2. OFSAA cash flow engine compounds the rate on the record at the time of the interest income calculation. If the record has repriced, the cash flow engine calculates the new rate, applies any rounding, caps/floors, or tease periods, and then applies the compounding calculation (COMPOUND_BASIS_CD) before calculating interest income (financial element 430).

3. Simple and At Maturity calculate interest in the same manner. These two codes do not compound the rate.

4. Compounded interest is calculated only when the compounding frequency is less than the PMT_FREQ. If the compounding frequency is greater than the PMT_FREQ, the model assumes simple compounding.

Data Verification Requirements and Suggested Defaults

1. A valid COMPOUND_BASIS_CD must be one of the values listed earlier.

2. Suggested default - If the record's compounding is unknown, default to COMPOUND_BASIS_CD = 160, Simple.

Convexity (CONVEXITY_C)

Definition

Convexity is a measure of the curvature or 2nd derivative of how the price of an instrument varies with interest rate, that is, how the duration of an instrument changes as the interest rate changes.

Module Usage

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output convexity when the Market Value option is selected from the Calculation Elements block. This result is written to FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with Convexity for each record if the related option is selected on the Output Preferences block. The instrument record will be updated for the scenario 1 result only.

The cash flow engine calculates convexity using the following formula:

$$Convexity = \frac{\sum_{t=1}^{n} Present\ Value\ of\ Cashflows_t \times Time\ Expired \times (1 + Time\ Expired)}{(1 + Yield\ to\ Maturity)^2}$$

Data Verification Requirements and Suggested Defaults

None

Current Book Balance (CUR_BOOK_BAL)

Definition

Current Gross Book Balance.

Module Usage

Oracle ALM

When there is deferred balance (DEFERRED_CUR_BAL), Oracle ALM uses CUR_BOOK_BAL to calculate accretion/amortization (financial element 540) and the deferred ending and average balances (financial element 520, 530).

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing uses CUR_BOOK_BAL during calculations for LEDGER_STAT transfer pricing and if selected in Application Preferences, for calculation of Charge / Credit amounts.

When Remaining Term Calculation Mode is selected in the Standard TP Process, and the Target Balance for the subject product leaf is Book Balance (in the TP Rule), Oracle Transfer Pricing Option Cost calculations use CUR_BOOK_BAL as the target balance to which the sum of future discounted cash flows is set equal.

Data Verification Requirements and Suggested Defaults

- Validate that CUR_BOOK_BAL = CUR_PAR_BAL + DEFERRED_CUR_BAL.

Current Gross Rate (CUR_GROSS_RATE)

Definition

Coupon rate of account, expressed in terms of an annualized rate.

Module Usage

When the Model with Gross Rates switch is turned on in the Oracle ALM Product Characteristics rule, or in the Transfer Pricing Rule, CUR_GROSS_ RATE is used to calculate forecasted cash flows. When switched on, the cash flow engine uses the record's CUR_GROSS_RATE for two calculations:

Amortization - When conventionally amortizing a record's balance with the Model with Gross Rates option selected, the cash flow engine uses the CUR_GROSS_RATE as the customer rate. If the option is not selected, then the cash flow engine uses CUR_NET_RATE as the customer rate for amortization calculation.

Prepayments - In order to determine the rate at which the customer prepays, the current customer rate must be compared to the market rate. If Model with Gross Rates is switched on, then the customer rate is represented by the CUR_GROSS_ RATE. If the switch is not turned on, the CUR_NET_RATE is used as the current customer rate.

If Model with Gross Rates is used, the Oracle ALM cash flow engine uses the CUR_GROSS_RATE for gross interest cash flow (financial element 435) calculations. This means that the record amortizes and prepays according to the CUR_GROSS_RATE, but the net cash flows associated interest income (financial element 430) are calculated from the CUR_NET_RATE.

> **Note:** Depending on the NET_MARGIN_CD value, interest income is calculated differently.

For a complete explanation of the relationship between NET_MARGIN_CD, CUR_GROSS_RATE, and CUR_NET_RATE, see Net Margin Code (NET_MARGIN_CD), page 6-79.

Following is an explanation of how Oracle ALM calculates CUR_GROSS_ RATE:

Before the NEXT_REPRICE_DATE, the cash flow engine uses the CUR_GROSS_RATE from the detail record as the gross rate.

At or beyond the NEXT_REPRICE_DATE, the cash flow engine matches the REPRICE_FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Oracle ALM Forecast Rates assumption rule. This is to assign a forecasted base rate. The MARGIN_GROSS is then added to this forecasted base rate. Any rounding, rate caps/floors, and tease periods are applied and the resulting rate is applied to the record as the gross rate.

> **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 does not reference Reprice Dates.

Data Verification Requirements and Suggested Defaults

- All term accounts require a valid CUR_GROSS_RATE.
- For non-interest earning/bearing accounts, CUR_GROSS_RATE = 0.
- For transaction accounts where the rate changes daily, based upon average balances, CUR_GROSS_RATE should be the spot rate at the time the extract program is run.
- CUR_GROSS_RATE >= 0.
- CUR_GROSS_RATE = MARGIN_GROSS + value of index (IRC) that the account is tied to (assuming periodic/lifetime caps/floors, tease periods do not apply and rounding is taken into consideration).
- CUR_GROSS_RATE should be validated while validating CUR_PAYMENT. See Current Payment (CUR_PAYMENT), page 6-37 for validation formulas.

Current Net Rate (CUR_NET_RATE)

Definition

Interest rate that interest income due to the bank is based upon.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow calculations reference CUR_NET_RATE for the following purposes:

Interest Income (Financial Element 430) Calculation CUR_NET_RATE is used to derive the interest cash flow (income/expense) that is due to the financial institution (referred to as net). The cash flow engine uses different Interest Income calculations depending on the ACCRUAL_BASIS_CD and INT TYPE. These calculations are presented under the field heading Accrual Basis Code (ACCRUAL_BASIS_CD) and Interest Type Code (INT_TYPE). Interest income is calculated on payment dates or the record's maturity date. As the calculations indicate, after referencing the ACCRUAL_BASIS_CD, the cash flow engine applies the CUR_NET_RATE to the entire payment period (last **Previous** payment date to next **Current** payment date). If any repricing occurred during the payment period, the cash flow engine uses the last repriced rate that occurred immediately before the next **Current** payment date.

> **Note:** Whether or not the Model with Gross Rates option has been selected in Product Characteristics (ALM) or the Transfer Pricing rule screen (Transfer Pricing), the cash flow engine always calculates the bank's income according to the CUR_NET_RATE.

Prepayments- As defined in the Prepayment rule interface, the cash flow engine compares the customer rate to the market rate when determining the prepayment rate. If the Model with Gross Rates option is not selected, the CUR_NET_RATE is the customer rate and therefore is used for prepayment calculations.

Amortization - When amortizing a record's balance, a key input is the record's customer rate. If Model with Gross Rates is not selected, then the cash flow engine uses the CUR_ NET_RATE for amortization purposes.

> **Note:** Depending on the NET_MARGIN_CD value, interest income is calculated differently. For a complete explanation of the relationship between NET_MARGIN_CD, CUR_GROSS_RATE, and CUR_NET_RATE see Net Margin Code (NET_MARGIN_CD), page 6-79.

Following is an explanation of how Oracle ALM calculates CUR_NET_RATE:

Before the NEXT_REPRICE_DATE, Oracle ALM uses the CUR_NET_RATE from the detail record as the net rate.

At or beyond the NEXT_REPRICE_DATE, Oracle ALM matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates rule. This is to assign a forecasted rate. The MARGIN is then added to this forecasted base rate. Any rounding, rate caps/floors, and tease periods are applied and the resulting rate is applied to the record as the net rate.

> **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 does not reference reprice dates.

Data Verification Requirements and Suggested Defaults

- All term accounts require a valid CUR_NET_RATE.
- For non-interest earning/bearing accounts, CUR_NET_RATE = 0.
- For transaction accounts where the rate changes daily based upon average balances, CUR_NET_RATE should be the spot rate at the time the extract program is run.
- For interest-bearing accounts, CUR_NET_RATE >= 0.
- CUR_NET_RATE = MARGIN + value of index that the account is tied to (assuming periodic/lifetime caps/floors do not apply and rounding is taken into consideration).

Current Option-Adjusted Spread (CUR_OAS)

Definition

The average spread over all stochastic rate paths that equate the discounted sum of future cash flows to the target balance at the As Of Date.

Module Usage

Transfer Pricing When Remaining Term Calculation Mode is selected in the Stochastic TP Process, the Oracle Funds Transfer Pricing option cost module writes the result of its option-adjusted spread calculations to this column.

Data Verification Requirements and Suggested Defaults

None

Current Par Balance (CUR_PAR_BAL)

Definition

Represents the starting balance from which Oracle ALM cash flows are generated.

Module Usage

Oracle ALM and Transfer Pricing Cash Flow Remaining Term Pricing Basis methodologies use the CUR_PAR_BAL field to derive the starting balance for amortization calculations. For amortizing accounts, CUR_PAR_BAL is the balance the cash flow engine amortizes over the remaining number of payments.

Oracle ALM

1. As the cash flow engine processes the record's payment dates and its maturity date, the CUR_PAR_BAL is reduced by the principal portion of the CUR_PAYMENT amount until the principal balance reaches **0**. Once the balance has been reduced to **0**, processing of the record ceases. The calculation method that defines how the CUR_PAR_BAL amount is reduced is represented by the AMRT_TYPE_CD and the CUR_PAYMENT fields.

2. The CUR_PAR_BAL is represented as Bucket 001 under financial element 60 (beginning balance in Oracle ALM Result Detail Table (RES_DTL_XXXXXX)) and as the CUR_PAR_BAL in the FSI_O_RESULT_MASTER Table. Generally the CUR_ PAR_BAL from the detail record matches the balances in the Result Detail and RESULT_MASTER Tables. However, there are three exceptions.

    1. **Exception 1** - If the record's PERCENT_SOLD > 0, the cash flow engine recalculates the balance to equal CUR_PAR_BAL * (100 - PERCENT_SOLD). This is because if any percentage of the balance is not actually owned by the financial institution, it is not included in the forecast. For details, see Percent Sold (PERCENT_ SOLD, page 6-105). In this case, the RESULT_MASTER CUR_NET_PAR_BAL and Result Detail balances would be smaller than the record's CUR_PAR_ BAL.

    2. **Exception 2** - This exception applies to Discount/Premium records where the DEFERRED CUR BAL <> 0. The cash flow engine reads in the CUR_PAR_BAL amount but processes only book balances when calculating cash flows. The cash flow engine calculates the book balance by adding the CUR_PAR_BAL and the DEFERRED_CUR_BAL. See Deferred Current Balance (DEFERRED_CUR_ BAL) , page 6-45 for details.

    3. **Exception 3** - This exception applies when the NEXT_PAYMENT_DATE < AS_OF_DATE. If the record's NEXT_PAYMENT_DATE is less than the AS_ OF_DATE and the record's AMRT_TYPE_CD <> 700, the cash flow engine reduces the CUR_PAR_BAL by the amount of the payments before the AS_OF_DATE.

        For example, if the record's CUR_PAR_BAL = $1000 and there are two payments (each worth $100 of principal), the cash flow engine's beginning balance (financial element 60) would be equal to $800.

3. For User-Defined Payment Patterns where the payment method is defined as %

Current Balance, Oracle ALM references the CUR_PAR_BAL field for all payment amounts beyond the first forecasted payment amount.

Transfer Pricing

1. For the cash flow **Remaining Term Pricing Basis** methodologies in Oracle Funds Transfer Pricing, as the cash flow engine processes the record's payment dates and its maturity date, the CUR_PAR_BAL is reduced by the principal portion of the CUR_ PAYMENT amount until the principal balance reaches **0**. Once the balance has been reduced to **0**, processing of the record ceases. The calculation method that defines how the CUR_PAR_BAL amount is reduced is represented by the AMRT_TYPE_CD and the CUR_PAYMENT fields.

2. For User-Defined Payment Patterns where the payment method is defined as **% Current Balance**, Oracle Funds Transfer Pricing references the CUR_PAR_BAL field for all payment amounts including the first one.

When Remaining Term Calculation Mode is selected in the TP Process, and the Target Balance for the subject product leaf in the TP Rule is Par Balance, Oracle Transfer Pricing Option Cost calculations use CUR_PAR_BAL as the target balance to which the sum of future discounted cash flows is set equal.

Data Verification Requirements and Suggested Defaults

- CUR_PAR_BAL requires a valid balance for all accounts. If CUR_PAR_BAL = 0, the cash flow engine does not process the record.

- CUR_PAR_BAL = CUR_BOOK_BAL - DEFERRED_CUR_BAL

- CUR_PAR_BAL should have the same sign as CUR_PAYMENT.

Current Payment (CUR_PAYMENT)

Definition

Represents the current periodic payment made against the outstanding balance.

Module Usage

For standard amortization types (those that are not non-patterned and non-scheduled), Oracle ALM and Funds Transfer Pricing use the CUR_PAYMENT from the detail record for the life of the record until a payment re-calculation occurs. A payment re-calculation occurs when the record is an:

- Adjustable record and a reprice date (NEXT_REPRICE_DATE) is reached.

- Adjustable record and the TEASER_END_DATE is reached (if TEASER_END_ DATE < NEXT_REPRICE_DATE, TEASER_END_DATE takes precedence).

- AMRT TYPE = 600 and the PMT_ADJUST_DATE, NEG_AMRT_EQ_DATE, or the NEG_AMRT_LIMIT is reached.

Depending on AMRT_TYPE_CD, CUR_PAYMENT may be composed of principal or interest or both.

AMRT_TYPE_CD 700 (Simple Interest) : CUR_PAYMENT equals interest only. The cash flow engine always calculates the interest component of any payment amount. Therefore, for extracting purposes, a simple interest record's CUR_PAYMENT = 0. If a repricing event (payment recalculation event) occurs, the interest amount of the payment (financial element 430, and if applicable, 435) is recalculated as indicated under the ACCRUAL_BASIS_CD. The only principal payment is made at maturity (Maturity Payment = financial element 195, 197).

AMRT_TYPE_CD 100, 400, 500, 600 (Conventionally Amortizing) : For extracting purposes, CUR_PAYMENT = principal + interest. If a payment recalculation event occurs, the cash flow engine re-calculates the total CUR_PAYMENT amount using the following formula:

$$\frac{\text{Current Par Balance}}{{}^{1}/_{\text{Current Rate}^{*}(1-(1+\text{ Current Rate})^{\wedge(\text{Remaining \# of Payments})}}}$$

For Leases (AMRT_TYPE_CD = 840), the amortization is conventional and the payment amount includes both principal and interest, but the equation is slightly different because it must incorporate the residual value. The following formula is used to compute the lease payment when re-calculation is required:

$$\left[-f - p(1+r)^{n} \times \left[\frac{r}{(1+r)^{n}-1}\right]\right]$$

Where:

f = Residual Amount

p = Current Par Balance

r = Current Rate

n = Number of payments

These calculations derive the total payment amount, principal, and interest. To determine the interest income (financial element 430) portion and the principal (financial element 190 or 192) portion of this payment amount, the cash flow engine calculates the interest income as indicated under the ACCRUAL_BASIS_CD. This income amount is then subtracted from the calculated total payment amount to determine the principal portion.

For additional information on AMRT_TYPE_CD 600, see Negative Amortization Amount (NEG_AMRT_AMT), page 6-74.

AMRT_TYPE_CD 820 (Level Principal) : For extracting purposes, CUR_PAYMENT = principal only. If a repricing event occurs, the cash flow engine recalculates the total CUR_PAYMENT amount using the following formula:

$$\frac{\text{Current Par Balance}}{\text{Remaining Number of Payments}}$$

This calculation yields the principal payment amount only (financial element 190 or 192). To derive the total payment amount, the interest amount (financial element 430) calculation (see Accrual Basis Code (ACCRUAL_BASIS_CD, page 6-9)) is applied and added to the principal portion.

- **AMRT_TYPE_CD 800** : Conventional payment schedule **Payment Amount** should contain both principal + interest. Payment recalculation is the same as for conventionally amortizing. The cash flow engine uses the CUR_PAYMENT from the detail record for the first forecasted payment. Therefore, the CUR_ PAYMENT on the detail record should equal the corresponding payment in the PAYMENT_SCHEDULE table.

- **AMRT_TYPE_CD 801** : Level principal payment schedule **Payment Amount** should contain principal only. Payment re-calculation is the same as under level principal AMRT_TYPE_CD 820. The cash flow engine uses the CUR_ PAYMENT from the detail record for the first forecasted payment. Therefore, the CUR_PAYMENT on the detail record should equal the corresponding payment in the PAYMENT_SCHEDULE table.

- **AMRT_TYPE_CD 802** : Simple interest payment schedule **Payment Amount** should be equal to zero (for simple interest, the engine ignores this field and just looks at the scheduled payment date). Interest recalculation is the same as indicated under simple interest AMRT_TYPE_ CD 700.

AMRT_TYPE_CD 1000 - 69999, User-Defined Payment Patterns : Depending on the payment method defined in the interface, Oracle ALM may or may not reference the CUR_PAYMENT field from the detail record.

Following is a grid that outlines the Oracle ALM use of the CUR_PAYMENT field depending on the payment method.

% Current Payment	% Original Payment	Interest Only	% Original Balance	%Current Balance
Never (always calculated)	ALM -Referenced for first cash flow only.	Never (always calculated)	ALM -Referenced for first cash flow only.	ALM -Referenced for first cash flow only.

Note: Oracle Funds Transfer Pricing does not reference CUR_PAYMENT when using User-Defined Payment Patterns.

The method of re-calculating payments for User-Defined Payment Patterns is dependent on the payment type that is defined for the payment pattern: conventional, level principal, or simple interest. Amortization re-calculations correspond to AMRT_TYPE_CDs 100, 820, and 700 respectively. Each is defined earlier.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD = 100, 400, 500, 600, 710, 800, 840 or is a conventionally-amortizing payment pattern, CUR_PAYMENT should include principal and interest.
- If AMRT_TYPE_CD = 820, 801 or is a level principal-amortizing payment pattern, CUR_PAYMENT should include principal only.
- If AMRT_TYPE_CD = 700, 802, 850 or is a simple interest-amortizing payment pattern, CUR_PAYMENT can be **0**.
- For AMRT_TYPE_CD <> Simple Interest AMRT_TYPE_CDs, CUR_PAYMENT must have the same sign as CUR_BOOK_BAL.
- CUR_PAYMENT must be the same sign as the CUR_BOOK_BAL and CUR_ PAR_BAL fields.
- If AMRT_TYPE_CD = 600 (Negative Amortization) and PMT_DECR_LF <> 0, CUR_PAYMENT should be greater than or equal to ORG_PAYMENT_AMT * (1 - PMT_DECR_LF/100).
- If AMRT_TYPE_CD = 600 (Negative Amortization) and PMT_ INCR_LF <> 0, CUR_PAYMENT should be less than or equal to ORG_PAYMENT_AMT * (1 + PMT_INCR_LF/100).
- CUR_PAYMENT can be validated by performing the following calculations:
  - For conventionally amortizing and Rule of 78s:

CUR_PAYMENT = (CUR_BOOK_BAL* (CUR_GROSS_ RATE/((12/PMT_FREQ[in months]) *100))) / (1 - ((1+(CUR_GROSS_ RATE /((12/ PMT_FREQ [in months])*100))) ^-(REMAIN_NO_PMTS_ C)))

- For leases:

  [(CUR_PAR_BAL x (1+r)^n) - RESIDUAL_AMOUNT] x [r/((1+r)^n -1)]

- For fixed rate accounts:

  CUR_PAYMENT = (ORG_BOOK_BAL* (CUR_GROSS_ RATE/((12/PMT_FREQ[in months]) *100))) / (1 - ((1+(CUR_GROSS_ RATE /((12/ PMT_FREQ [in months])*100))) ^-(ORG_TERM/PMT_ FREQ [in months])))

- For Level Principal records:

  CUR_PAYMENT = CUR_BOOK_BAL / REMAIN_NO_PMTS_C

- For Annuities:

  CUR_PAYMENT = (MATURITY_AMOUNT – CUR_PAR_BAL) / REMAIN_NO_PMTS_C

- For fixed-rate accounts level principal records, the following should also be true:

  CUR_PAYMENT = ORG_BOOK_BAL/ (ORG_TERM/ PMT_FREQ [in months])

- For balloon records, the calculated remaining number of payments in the amortization term (CRPAT) must be calculated first. This is demonstrated in the Remaining Number of Payments (REMAIN_NO_PMTS_C) section. The following calculation is used:

  CUR_PAYMENT = (CUR_BOOK_BAL* (CUR_GROSS RATE/((12/PMT_FREQ[in months]) *100))) / (1 - ((1+(CUR_GROSS_ RATE /((12/ PMT_FREQ [in months])*100))) ^-(CRPAT)))

- For fixed rate accounts, the following should be true:

  CUR_PAYMENT = (ORG_BOOK_BAL* (CUR_GROSS_ RATE/((12/PMT_FREQ[in months]) *100))) / (1 - ((1+(CUR_GROSS_ RATE /((12/ PMT_FREQ [in months])*100))) ^-(AMRT_TERM/PMT_ FREQ [in months])))

- **Validation for FE190:**
  - If Total Installments = 3,171,629.758 and Interest Cash Flow = 991324.79

    Payment Runoff-Positive = 3,171,629.758 - 991324.79 = 2,180,304.968

  - Time period = 24.5 months or 744 days (19 July 2013(As of Date +1 Day - 1 august-2015(Maturity Date))

- Compound frequency = Quarterly or 8.1667 times over the remaining life of the loan
- Interest Rate 12.08% or 3.02% per quarter
- Cur Par Balance = 28,885,597
- Prepayment Constant Prepayment at 25%

Following is the calculation for FE 190 in bucket 1:

Here, next reprice date(07/31/2013) is less than next payment date(08/01/2013). Hence, engine will re-calculate the current payment using the following formula.

Cur payment = 28885597.00/((1/0.0302)*(1-(1+0.0302)^(-1*9))) = 3713349.496

Where the rate of interest .0302 is arrived using (9.48+2.6)%*3/12 = .0302

Interest cash flow(FE430) on next payment date = 28885597.00 *(9.48+2.6)%*(360/365)*3/12*(106/92) = 991324.786

Here, the factor 106/92 is used to adjust the interest payment for actual payment period.

FE190 = 3713349.496 - 991324.786

= 2722024.71 - this is the Engine's output for Bucket 1.

Following is the calculation for FE 190 in bucket 3:

On the second reprice event (on 10/31/2013), the rate is not changing. Hence, engine will not re-calculate the current payment. So the cur payment still remains 3713349.496. Since the "Cash Flow Treatment" option selected is "Refinance", engine will adjust the cur payment using the below formula.

Adjusted cur payment = cur payment * (1-Adjusted prepay factor) = 3713349.496 * (1 - 0.079955271) = 3416447.631

FE190 = Adjusted cur payment - FE 430

= 3416447.631 - 717005.6233

= 2699442.007 - this is the Engine's output for Bucket 3.

Current Static Spread (CUR_STATIC_SPREAD)

Definition

The spread over the implied forward rates that equates the discounted sum of future cash flows to the target balance at the As Of Date.

Module Usage

Transfer Pricing- When Remaining Term Calculation Mode is selected in the Stochastic TP Process, the Oracle Funds Transfer Pricing Option Cost module writes the result of its static spread calculations to this column.

Data Verification Requirements and Suggested Defaults

None

Current Transfer Pricing Period Average Daily Balance (CUR_TP_PER_ADB)

Definition

The average balance at the LAST_REPRICE_DATE.

Module Usage

Oracle ALM

Oracle ALM does not reference CUR_TP_PER_ADB or PRIOR_TP_PER_ ADB.

Transfer Pricing

When processing with the mid-period repricing option, Oracle Funds Transfer Pricing references CUR_TP_PER_ADB as the average daily balance at the time of the last repricing event. This field is used in conjunction with the PRIOR_TP_PER_ADB field.

1. Mid-period repricing produces an average transfer rate over the current processing month if the LAST_REPRICE_DATE occurred since the beginning of the processing month. CUR_TP_PER_ADB and PRIOR_TP_PER_ADB are used as average balance weightings in the mid-period pricing equation.

   An example of a mid-period pricing scenario as follows. The table provides the scenario specifics, followed by an explanation.

Fields	Scenario
AS_OF_DATE	31-DEC
LAST_REPRICE_DATE	15-DEC
LAST_PAYMENT_DATE	15-DEC (balance was reduced on this date)
CUR_TP_PER_ADB	$10,000
PRIOR_TP_PER_ADB	$15,000

Fields	Scenario
TRANSFER_RATE	3% from 15-NOV to 15-DEC (prior period, 30 days in the period)
TRANSFER_RATE	5% from 15-DEC to 15-JAN (current period, 31 days in the period)

Without the mid-period repricing option, Transfer Pricing would assign a 5% transfer rate to the record for the month of December. However, this is the transfer rate only for the second half of December. The true transfer rate for the month should be a balance-weighted average transfer rate over the entire month. Mid-period repricing provides this by calculating the final transfer rate by weighting the transfer rate results (from current and previous repricing periods) by average balances and days. This final transfer rate is then applied to the detail record's TRANSFER_RATE field.

The equation used by Oracle Transfer Pricing for calculating Mid-Period Repricing is as follows:

((CUR_TP_PER_ADB * Current Period Transfer Rate * Current Period Days) + S(PRIOR_TP_PER_ADB * Prior Period Transfer Rate * Prior Period Days)) / ((CUR_TP_PER_ADB * Current Period Days) + S(PRIOR_TP_PER_ADB * Prior Period Days))

From the earlier example, the equation would be:

((10,000 * 5% * 31) + (15,000 * 3% * 30))/ ((10,000 * 31) + (15,000 * 30)) = 3.82%

Therefore, the correct transfer rate is 3.82%.

2. In reference to the earlier calculation, the CUR_TP_PER_ADB is used to determine the balance as of the LAST_REPRICE_DATE and PRIOR_TP_PER_ ADB is used to determine the balance as of the repricing dates prior to the LAST_REPRICE_DATE.

3. If the TEASER_END_DATE is greater than the AS_OF_DATE, the Mid-Period Repricing does not apply and the CUR_TP_PER_ADB and PRIOR_TP_PER_ ADB fields are not used.

4. See the *Oracle Financial Services Funds Transfer Pricing User Guide* for more information.

Data Verification Requirements and Suggested Defaults

- If the record is adjustable and repricing occurs within the month, CUR_TP_ PER_ADB = (average) balance at the time of the LAST_REPRICE_DATE.

- If the CUR_TP_PER_ADB and PRIOR_TP_PER_ADB are not available, use CUR_PAR_BAL as your default (otherwise mid-period repricing could result in a zero transfer rate).

Current Yield (CUR_YIELD)

Definition

The Yield to Maturity of the instrument record is computed using an internal rate of return calculation.

Module Usage

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output Yield to Maturity as a standard output. This result is written to FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with YTM for each record if the related option is selected on the Output Preferences block. The instrument record will be updated for the scenario 1 result only.

Data Verification Requirements and Suggested Defaults

None

Deferred Current Balance (DEFERRED_CUR_BAL)

Definition

The current, un-amortized deferred balance representing future income/expense, such as premium, discount, fees, and costs.

Module Usage

DEFERRED_CUR_BAL holds the amount of un-amortized discount or premium (fee or cost) associated with a bond or other instrument record.

Discounted Instrument A discount loan or instrument is one with interest deducted from the face amount (CUR_PAR_BAL) of the loan at its origination. The discount amount (DEFERRED_CUR_BAL) is the difference between the loan's current market price (CUR_BOOK_BAL) and its stated par value (CUR_PAR_BAL). The DEFERRED_CUR_BAL, which represents income, is amortized over the life of the instrument according to a constant yield (IRR) calculation. Therefore, as the instrument approaches maturity, the CUR_BOOK_BAL approaches the CUR_PAR_BAL.

For discounted instruments, the DEFERRED_CUR_BAL should be a negative balance. This indicates to the cash flow engine that the balance is income.

Premium Instrument A premium bond or instrument is one in which the face value is issued below the book value. The premium is represented by the DEFERRED_CUR_BAL field, and, as with discounts, the deferred portion is accreted over the life of the instrument. For a premium instrument the DEFERRED_ CUR_BAL represents an expense.

For instruments sold at a premium, DEFERRED_CUR_BAL should be positive, indicating that the balance is an expense.

The relationship between book, par and the deferred amount is as follows:

CUR_BOOK_BAL = CUR_PAR_BAL + DEFERRED_CUR_BAL

An example of this relationship for a discounted loan follows:

CUR_PAR_BAL = $10,000

CUR_BOOK_BAL = $9,000

DEFERRED_CUR_BAL = - $1,000

1. Standard behavior is for the cash flow engine to perform a constant-yield (IRR) amortization of the DEFERRED_CUR_BAL. This enables the deferred balance to be amortized evenly over the life of the instrument. This life-long amortization rather than a one-time realization of the deferred amount at the inception of the instrument is dictated by general accounting rules regarding discount or premium instruments. For certain fees and costs, as well as premiums and discounts, banks must recognize income/expense over the life of an account instead of at the inception of the account. Hence, some deferred balances are amortized over an account's maturity term even if the account itself does not amortize (Cash Flow Calculations, page 5-1, provides a textual explanation for the constant-yield calculation.)

   In addition to level yield amortization of the discount or premium, users also have the option to apply a simplified straight line amortization of the premium or discount. With this method, the discount or premium will be allocated in equal increments over the remaining life of the instrument. For related code values, see the description for AMORT_METH_PDFC_CD, page 6-14.

2. If the DEFERRED_CUR_BAL = 0, the cash flow engine recognizes this record as having no discount or premium (CUR_BOOK_BAL = CUR_PAR_BAL).

Data Verification Requirements and Suggested Defaults

- For deferred income, (fees, discount), DEFERRED_CUR_BAL < 0.
- For deferred expense, (costs, premium), DEFERRED_CUR_BAL > 0.
- For accounts with deferred balances, the following equation must hold true:

  CUR_BOOK_BAL = CUR_PAR_BAL + DEFERRED_CUR_BAL
- For accounts with no deferred balances, DEFERRED_CUR_BAL = 0

Deferred Original Balance (DEFERRED_ORG_BAL)

Definition

Original non-amortized deferred balance representing future income/expense, such as premium, discount, fees, and costs.

Module Usage

This field must exist on the instrument table for cash flow processing, but is not used in any of the cash flow calculations.

Data Verification Requirements and Suggested Defaults

None

Devolvement Status Code (DEVOLVEMENT_STATUS_CD)

Definition

Related to Letters of Credit, devolvement status can be either not devolved or devolved. Not devolved is the default state. Devolvement occurs when the purchaser of the LC is not able to pay for receivables. When the LC moves into devolvement status, the bank begins the recovery process.

Module Usage

This field is required by the ALM process when the instrument uses a Devolvement/Recovery Behavior Pattern (BEHAVIOR_TYPE_CD = 3). The following values are possible for DEVOLVEMENT_STATUS_CODE.

0 – Not Devolved

1 – Devolved

In addition to DEVOLVEMENT_STATUS_CD, users are also required to populate the BEHAVIOR_SUB_TYPE_CD for Devolvement and Recovery Behavior Patterns. The following relationships are possible based on DEVOLVEMENT_STATUS_CD:

```
If DEVOLVEMENT_STATUS_CD = 0 (not devolved)
 BEHAVIOR_SUB_TYPE_CD = 301 (Sight Devolvement) or
 BEHAVIOR_SUB_TYPE_CD = 303 (Usance Devolvement) or
 BEHAVIOR_SUB_TYPE_CD = 302 (Sight Recovery) or
 BEHAVIOR_SUB_TYPE_CD = 304 (Usance Recovery)

If DEVOLVEMENT_STATUS_CD = 1 (devolved)
 BEHAVIOR_SUB_TYPE_CD = 302 (Sight Recovery)
 BEHAVIOR_SUB_TYPE_CD = 304 (Usance Recovery)
```

Data Verification Requirements and Suggested Defaults

- When BEHAVIOR_TYPE_CD = 3 then DEVOLVEMENT_STATUS_CD must be **0** or **1**.
- If BEHAVIOR_TYPE_CD <> 3, then DEVOLVEMENT_STATUS_CD can be NULL.

Duration (DURATION_C)

Definition

The DURATION_C column holds the calculated Macaulay Duration output.

Module Usage

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output Macaulay Duration when the Market Value option is selected from the Calculation Elements block. This result is written to FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with Macaulay Duration for each record if the related option is selected on the Output Preferences block. Each instrument record will be updated for the scenario 1 result only.

The cash flow engine calculates the average life using the following formula:

$$\text{Macaulay Duration} = \frac{\sum_{n=1}^{n}(\text{MV}_{\text{cf}} * \text{t})}{\sum_{n=1}^{N}\text{MV}_{\text{cf}} * \frac{\text{pmts}}{\text{year}}}$$

Data Verification Requirements and Suggested Defaults

None

Effective Interest Rate (EFF_INTEREST_RATE_C)

Definition

Effective interest rate represents the Yield to Maturity of the instrument inclusive of fees and costs.

Module Usage

This financial measure is calculated by Oracle ALM and Oracle Hedge Management and IFRS Valuations when the corresponding Calculation Element is selected and the result is written to each individual instrument record selected in the process. Effective Interest Rate is not written to FSI_O_RESULT_MASTER or FSI_O_CONSOLIDATED_MASTER.

The Effective Interest Rate calculation is similar to Yield to Maturity, with the main difference being that it additionally considers the amount input into the FEES_EIR column.

Data Verification Requirements and Suggested Defaults

None

Expected Balance (EXPECTED_BAL)

Definition

This field represents the expected balance of a mortgage offset account (MOA). It is a data input (optionally) for each mortgage record and represents an offset to the outstanding loan balance for purposes of calculating interest.

Module Usage

Used for modeling Mortgage Offset Accounts (MOA's). If populated, the cash flow engine will reduce the loan's principal balance in calculating interest.

The following example illustrates how Mortgage Offset Account work:

Mortgage Offset Account Example	
Current Balance	100,000
Rate of Interest (fixed)	10%
Interest Type	Arrears
Amortization term (Days)	366
Accrual Basis	1
Compounding Basis	160
Payment frequency (Months)	3
Current payment	25,000

Mortgage Offset Account Example	
Maturity Date	12/31/2010
Last Payment Date	12/31/2009
Remaining number of payments	4
MOA Expected Balance	60,000
MOA Expected Balance Growth %	5.00%
Mortgage Offset Percentage	40%
Minimum Balance of MOA	3,000

As depicted in the earlier sample data record, the MOA Expected Balance is 60,000 at the last payment date (12/31/2009). The MOA Expected Balance is changed at each payment date by a growth percentage factor. This amount could be positive, negative or zero. Each of the remaining payment dates would have an MOA expected balance of :

3/31/2010	60,750.00	= 60,000 * (1+(.05/12))*3)
6/30/2010	61,509.38	
9/30/2010	62,278.24	
12/31/2010	63,056.72	

Derived Information:

Accrual Basis **(1 = 30/360)**	**0.0833** **3333**
Compounding per payment frequency	3

Accrual Basis (1 = 30/360)	**0.0833 3333**				
Payment frequency (Days)	91	91	92	92	90
Payment Dates		3/31/2010	6/30/2010	9/30/2010	12/31/2010
Remaining number of payments		3	2	1	0
MOA Expected Balance on every Payment date		60,750.00	61,509.38	62,278.24	63,056.72

A) Interest Cashflow Calculation without Offset				
Payment Date	3/31/2010	6/30/2010	9/30/2010	12/31/2010
Opening Balance	100,000.00	76,908.31	53,226.86	28,940.66
Interest Rate per payment	0.025	0.025	0.025	0.025
Compounded Interest Rate	0.025	0.025	0.025	0.025
Interest cash flow	2,520.89	1,938.77	1,341.79	729.56
Actual Current payment amount	25,000.00	25,000.00	25,000.00	25,000.00
Principal Runoff	22,479.11	23,061.23	23,658.21	24,270.44
Ending Balance	77,520.89	53,847.09	29,568.65	4,670.22
MOA Prepay Runoff	612.58	620.23	627.99	635.84
Maturity Runoff	-	-	-	4,670.22

A) Interest Cashflow Calculation without Offset				
B) Interest Cashflow Calculation with Offset				
Payment Date	3/31/2010	6/30/2010	9/30/2010	12/31/2010
Remaining Balance after offset (only for Interest calculations)	75,700.00	52,304.56	28,315.56	3,717.97
Interest cash flow	1,908.31	1,318.54	713.8	93.73
New Financial Elements related to MOA				
MOA Prepayment Runoff (FE 184)	612.58	620.23	627.99	635.84
Timing of MOA Prepayment Runoff (FE 185)	---> MOA Prepayment Runoff * (Start date of the bucket - Payment date)/((Start date of the bucket - Payment date)+((Payment date+1)-End date of the bucket))			

The amount of interest calculated is derived as follows:

The Remaining Balance after Offset = Opening balance – (**Expected balance** on every payment date * Mortgage offset %)

For 3/31/2010, remaining balance for interest calculation = 100,000 – (60,750 * 40%) = 75,700.00

Interest cash flow considering the offset for the period is 75,700 * .025 = 1908.31

Data Verification Requirements and Suggested Defaults

For Mortgage Offset Account instruments, EXPECTED_BAL > 0

Expected Balance Growth Percentage (EXPECTED_BAL_GROWTH_PCT)

Definition

This input field represents the expected growth of the offset account balance, used to calculate interest for Mortgage Offset Accounts (MOA). This growth can be positive, negative or zero. Mortgage offset account assumptions are optional when computing periodic interest.

Module Usage

For sample calculations, see Expected Balance (EXPECTED_BAL), page 6-49.

Data Verification Requirements and Suggested Defaults

For Mortgage Offset Account instruments, EXPECTED_BAL_GROWTH_PCT = 0 (Default)

Gross Margin (MARGIN_GROSS)

Definition

The contractual spread that is added to the pricing index, which results in the customer (Gross) rate, for adjustable rate accounts.

Module Usage

Oracle ALM

1. If the Oracle ALM Product Characteristics, Model with Gross Rates option is switched on, MARGIN_GROSS is used by the cash flow engine during cash flow generation.

2. For adjustable-type records, MARGIN_GROSS is the contractual spread above/below the index that is applied throughout the instrument's life. The customer's gross rate (CUR_GROSS_RATE) is equal to the index to which the record is tied to plus a spread, which is defined by the MARGIN_GROSS field.

3. The events of a repricing involving MARGIN_GROSS are as follows:

   At a repricing event (or a TEASER_END_DATE) for an adjustable-rate record, the cash flow engine matches the INTEREST_RATE_CD, REPRICE_FREQ and repricing date of the detail record to the Forecast Rates assumption attached to the ALM Process.

   After matching the rate from the Forecast Rates assumption rule, the cash flow engine adds the MARGIN_ GROSS amount and applies any teases, rate caps/floors, and rounding to derive the rate that is applied to the record.

   > **Note:** As explained in the ADJUSTABLE_TYPE_CD section, the cash flow engine does not reference repricing date information for ADJUSTABLE_ TYPE_CD = 30 or 50.

4. The repriced rate, defined in Step 3, equals the coupon rate that is used for amortization and prepayment purposes only. Interest income (financial element

430) is still derived from the CUR_NET_RATE + MARGIN.

> **Note:** If the Model with Gross Rates option switched off (it is typically off), the cash flow engine uses the CUR_NET_RATE and MARGIN for amortization, prepayment, and interest income calculation purposes.
>
> However, there is one exception to this, which is described in the Net Margin Code (NET_MARGIN_CD) section.

For additional information, see Current Gross Rate (CUR_GROSS_RATE), page 6-32

Transfer Pricing

MARGIN_GROSS is not used by Oracle Funds Transfer Pricing.

Data Verification Requirements and Suggested Defaults

- For fixed rate accounts, MARGIN_GROSS = 0.
- For adjustable rate accounts with no contractual margin, MARGIN_GROSS = 0.
- For administered rate accounts, MARGIN_GROSS = 0.

Historic Option-Adjusted Spread (HISTORIC_OAS)

Definition

The average spread over all stochastic rate paths that equates the discounted sum of future cash flows to the target balance at origination.

Module Usage

Transfer Pricing When Standard Calculation Mode is selected in the Stochastic TP Process, the Oracle Funds Transfer Pricing option cost module writes the result of its option-adjusted spread calculations to this column.

Historic Static Spread (HISTORIC_STATIC_SPREAD)

Definition

The spread over the implied forward rates that equates the discounted sum of future cash flows to the target balance at origination.

Module Usage

Transfer Pricing- When Standard Calculation Mode is selected in the Stochastic TP Process, the Oracle Funds Transfer Pricing option cost module writes the result of its static spread calculations to this column.

Holiday Calculation Option Code (HOLIDAY_CALC_OPTION_CD)

Definition

Holiday Calculation Option Code

Module Usage

Oracle ALM and FTP

There are two options for holiday calendar calculation. They are:

1. Shift Dates Only

2. Recalculate Payment

Holiday Calendar Code (HOLIDAY_CALENDAR_CODE)

Definition

Holiday Calendar Code.

Module Usage

Oracle ALM and FTP

The Holiday Calendar Code stores the Holiday Calendar Definition which contains the details such as name, description, weekend holidays, fixed and moving holidays. The Holiday Calendar Code should be a unique number ranging from 1 to 99999.

Holiday Rolling Convention Code (HOLIDAY_ROLLING_CONVENTION_CD)

Definition

Holiday Rolling Convention Code

Module Usage

Oracle ALM and FTP There are five types for holiday rolling convention code. They are:

1. Following business day

2. Modified following business day

3. Previous business day

4. Modified previous business day

5. Unadjusted

Data Verification Requirements and Suggested Defaults

None

ID Number (ID_NUMBER)

Definition

Account number or other unique identifier used for identifying individual customer accounts.

Module Usage

ID_NUMBER identifies the individual customer accounts in the instrument tables. The ID_NUMBER should be unique for a given IDENTITY_CODE within an instrument table. Cash flow processing uses ID_NUMBER to identify each account as it is processed.

It is also important for instruments with Payment Schedules (AMRT_TYPE_CD 800, 801, 802) because the cash flow engine uses the INSTRUMENT_TYPE_CD and ID_NUMBER to determine the payment dates and amounts from the FSI_D_PAYMENT_SCHEDULE table.

Data Verification Requirements and Suggested Defaults

- ID_NUMBER is loaded into instrument tables from the source data. Because the database ensures that ID_NUMBER is unique for each IDENTITY_CODE, there are no edits or defaults for this field.

Identity Code (IDENTITY_CODE)

Definition

Data identifier.

Module Usage

IDENTITY_CODE is an identifier for sets of data loaded into an instrument table. IDENTITY_CODE identifies the data source for the individual customer account. The cash flow engine uses IDENTITY_CODE to uniquely identify individual customer accounts.

The combination of ID_NUMBER and IDENTITY_CODE must be unique.

Data Verification Requirements and Suggested Defaults

- IDENTITY_CODE is loaded into instrument tables from the source data. Because the database ensures that the combination of IDENTITY_CODE and ID_NUMBER is unique, there are no edits or defaults for this field.

Instrument Type Code (INSTRUMENT_TYPE_CD)

Definition

Code identifying the instrument table / category of the customer account.

Module Usage

INSTRUMENT_TYPE_CD identifies the source table of instrument record. The following table lists each of the available INSTRUMENT_TYPE_CD values:

Code Value	Description
110	Commercial loans
120	Consumer loans
130	Mortgages
140	Investments
141	MBS
145	Money Market Contracts
150	Credit card
160	Credit Lines
170	Leases
180	Loan Contracts
205	Annuity Contracts
210	Deposits

Code Value	Description
215	Checking and Savings
220	Wholesale Funding
230	Term Deposits
240	Retirement Accounts
310	Break Funding
410	Ledger Instruments
510	Merchant Cards
520	Mutual Funds
530	Other Services
540	Trusts
910	Derivatives
920	Forward Rate Agreements
925	Futures
930	FX Contracts
940	Caps / Floors / Collars
950	Guarantees
960	Swaps
970	Rate Lock Commitments

The cash flow engine uses the INSTRUMENT_TYPE_CD to determine the instrument of an account when accounts of different instruments are grouped together (on a report or other query).

It is also important for instruments with Payment Schedules (AMRT_TYPE_CD 800, 801, 802) because the cash flow engine uses the INSTRUMENT_TYPE_CD and ID_NUMBER to determine the payment dates and amounts from the FSI_D_PAYMENT_SCHEDULE table.

Data Verification Requirements and Suggested Defaults

The INSTRUMENT_TYPE_CD value for an individual account should match the instrument type of the table in which it is stored. For example, all account records stored in the TERM DEPOSITS table should be assigned INSTRUMENT_TYPE_CD = 230.

Interest Type Code (INT_TYPE)

Definition

Determines whether interest cash flows are paid in advance or in arrears.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference INT_TYPE in determining whether interest payments are made in arrears or in advance. INT_TYPE impacts the calculation of interest income (financial element 430, 435).

1. If INT_TYPE = 1, the record is considered **interest in arrears**. Interest payments are paid at the end of the payment period along with the principal payments. For interest income formulas, see Accrual Basis Code (ACCRUAL_BASIS_CD), page 6-9.

2. If INT_TYPE = 2, the record is considered **interest in advance**. Interest payments are paid at the beginning of the payment period starting from the ORIGINATION_DATE. Payments are made on every payment date except for the MATURITY_DATE.

3. If INT_TYPE = 3, the record is considered as rate **set in arrears** and the interest is also paid in arrears. This is unique case where both the rate is set in arrears and the interest is paid in arrears. This setting will typically be used only for modeling Set in Arrears SWAPS.

4. The calculation used to determine interest income (financial element 430) for an interest in advance record depends also on the ACCRUAL_BASIS_CD. Following are the relevant equations for an **interest in advance** calculation:

   For ACCRUAL_BASIS_CD 30/360, 30/365 and, 30/Actual the interest income calculation, when PMT_FREQ_MULT = M (assuming no compounding), is:

   Current Period's Ending Balance * Cur Net Rate/100 * PMT_FREQ [number of months] * [accrual basis] * (Following Payment Date - Next Payment Date)/(Calculated Following Payment Date -Next Payment Date) where:

- **Following Payment Date** is the payment after **Next Payment Date**.

- **Calculated Following Payment Date** is the **Next Payment Date** rolled forward by the number of months in PMT_FREQ.

  In most cases (a) would be the same as (b); however, if there is a short or extended maturity, (a) <> (b), and therefore the last interest cash flow (in other words, the payment just prior to maturity) would need to consider this factor.

  > **Note:** The **Following Payment Date** is the payment that follows the one currently being calculated.

  For ACCRUAL_BASIS_CD Actual/365, Actual/Actual, Actual/360 (the example is for an Actual/365 record), the interest income calculation is:

  Current Period's Ending Balance * Cur Net Rate/100 * (Following Payment Date -Current Payment Date)/365.

5. Even though the cash flow engine pays **interest in advance** on every payment date except for the MATURITY_DATE, the REMAIN_NO_PMTS_C field should count MATURITY_DATE as a payment date because principal is still paid on this date.

Data Verification Requirements and Suggested Defaults

- If INT_TYPE = 2, AMRT_TYPE_CD = 700, 820, 850, 801, 802 or non-conventionally amortizing User-Defined Payment Patterns.

- INT_TYPE valid values are **1** and **2**.

- If INT_TYPE = 2, REMAIN_NO_PMTS_C should still count MATURITY_DATE as a payment date.

- If INT_TYPE = 2 and ORIGINATION_DATE > AS_OF_DATE, NEXT_PAYMENT_DATE and LAST_PAYMENT_DATE = ORIGINATION_DATE.

- If the ORIGINATION_DATE > AS_OF_DATE, the NEXT_PAYMENT_DATE and LAST_PAYMENT_DATE should both equal the ORIGINATION_DATE.

- For conventionally amortizing records, interest in advance is not a valid INT_TYPE code. Interest in advance functions with simple interest and level principal AMRT_TYPE_CDs only.

- If a compounding method has been chosen, the cash flow engine derives the compounded rate before calculating the preceding interest income amounts. For more information, see Compounding Basis Code (COMPOUND_BASIS_CD), page

6-30.

Interest Rate Code (INTEREST_RATE_CD)

Definition

Identifies the index to which adjustable and floating rate accounts are tied.

Module Usage

Oracle ALM

The cash flow engine references INTEREST_RATE_CD when calculating a forecasted interest rate for an ALM process.

1. The detail record's INTEREST_RATE_CD has a corresponding reference in the Oracle ALM Forecast Rate assumption rule and related Rate Management Interest Rate UI. The Rate Management Interest Rate code screen is where the term structure and historical rate values are managed. The Forecast Rates UI is where the forecast rates are input for each IRC across all required scenarios.

2. At a repricing event (or a TEASER_END_DATE) for an adjustable-rate record, the cash flow engine matches the INTEREST_RATE_CD, REPRICE_FREQ and repricing date of the detail record to the Forecast Rates assumption rule attached to the ALM Process. After matching the rate from the Forecast Rates rule, the cash flow engine adds the MARGIN amount and applies any teases, rate caps/floors, and rounding to derive the rate that is applied to the record.

   > **Note:** As explained in the Adjustable Type Code (ADJUSTABLE_TYPE_ CD) section, the cash flow engine does not reference repricing date information for ADJUSTABLE_TYPE_CD = 30 or 50.

3. The exact value of the INTEREST_RATE_CD is user-defined, but it must be within the range of 1 - 99999.

   Note, fixed-rate records do not reference an INTEREST_RATE_CD and can be defaulted to a value of 0.

Transfer Pricing

Oracle Funds Transfer Pricing does not reference INTEREST_RATE_CD because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- If ADJUSTABLE_TYPE_CD = 0, INTEREST_RATE_CD can be defaulted to 0.

- If ADJUSTABLE_TYPE_CD <> 0, INTEREST_RATE_CD = 001 - 99999. Be sure that the Interest Rate Code value mapped to data records has a corresponding entry in the Rate Management Interest Rate Codes user interface.

Issue Date (ISSUE_DATE)

Definition

Date the account was originated (issued) by the originating institution. For derivative instruments, would be equal to the Trade Date.

Module Usage

The cash flow engine references ISSUE_DATE for future originating accounts.

ISSUE_DATE is referenced by the cash flow engine in Market Value and GAP calculations when processing records with future origination dates. If ISSUE_DATE <= AS_OF_DATE and ORIGINATION_DATE > AS_OF_DATE, the record is an account which has been traded but not settled. Therefore, the record is included in the Oracle ALM Market Value and GAP results if these processing options have been selected.

However, if ISSUE_DATE > AS_OF_DATE and ORIGINATION_DATE > AS_OF_DATE, the record has been neither traded nor originated. Therefore the balance associated with the record is not included in static Market Value or static GAP results (static means as of the AS_OF_DATE).

> **Note:** If the dynamic start dates in the Oracle ALM Time Buckets definition are set on or beyond the future ORIGINATION_ DATE of the record, Market Value and GAP results are calculated for the related Dynamic Start Date (START_DATE_INDEX).

Data Verification Requirements and Suggested Defaults

- For accounts originated by the current institution, ISSUE_DATE = ORIGINATION_DATE.
- For accounts acquired through acquisition of another institution, or purchase of a pool of accounts, ISSUE_DATE < ORIGINATION_DATE.
- If ORIGINATION_DATE from the original institution is not available, ISSUE_DATE = ORIGINATION_DATE.
- ISSUE_DATE <= LAST_REPRICE_DATE
- ISSUE_DATE <= ORIGINATION_DATE. ISSUE_DATE cannot be greater than ORIGINATION_DATE.

- For adjustable-rate records, when the LAST_REPRICE_DATE is less than the ISSUE_DATE, transfer pricing does not occur. If ADJUSTABLE_TYPE_CD = 0, LAST_REPRICE_DATE can be defaulted to be less than or equal to the ISSUE_DATE.

Last Payment Date (LAST_PAYMENT_DATE)

Definition

The date on which the record's last (previous) payment was made.

Module Usage

Oracle ALM

Oracle ALM uses LAST_PAYMENT_DATE to calculate the payment period and interest income (financial element 430) for the first forecasted cash flow.

1. The first forecasted cash flow from the AS_OF_DATE references NEXT_PAYMENT_DATE minus LAST_PAYMENT_DATE in order to determine the payment period for interest income calculations. The use of LAST_PAYMENT_DATE rather than (NEXT_PAYMENT_DATE - PMT_FREQ) provides for short or extended first period payments. Beyond the first forecasted cash flow (NEXT_PAYMENT_DATE), the cash flow engine rolls forward by PMT_FREQ until MATURITY_DATE. For more information, see Next Payment Date (NEXT_PAYMENT_DATE), page 6-81.

2. For instruments that have been originated in the past (AS_OF_DATE >= ORIGINATION_DATE), the LAST_PAYMENT_DATE should always be greater than or equal to the ORIGINATION_DATE.

3. For future originations (AS_OF_DATE < ORIGINATION_DATE), LAST_PAYMENT_DATE should always be equal to the ORIGINATION_DATE.

4. Even though the first forecasted cash flow may be extended, the PMT_FREQ should always be extracted as the records general frequency of payment.

5. For interest income calculation examples that reference LAST_PAYMENT_ DATE, see Accrual Basis Code (ACCRUAL_BASIS_CD), page 6-9 and Interest Type Code (INT_TYPE), page 6-59.

Transfer Pricing

Oracle Funds Transfer Pricing does not reference LAST_PAYMENT_DATE.

Data Verification Requirements and Suggested Defaults

- Default to NEXT_PAYMENT_DATE - PMT_FREQ if actual LAST_PAYMENT_DATE is not available.

- LAST_PAYMENT_DATE < NEXT_PAYMENT_DATE
- If ORIGINATION_DATE > AS_OF_DATE, LAST_PAYMENT_DATE = ORIGINATION_DATE.
- If ORIGINATION_DATE <= AS_OF_DATE, LAST_PAYMENT_DATE >= ORIGINATION_DATE.
- If INT_TYPE = 2 and ORIGINATION_DATE > AS_OF_DATE, NEXT_PAYMENT_DATE and LAST_PAYMENT_DATE = ORIGINATION_DATE. (e.g. for FRA's)

Last Repricing Date (LAST_REPRICE_DATE)

Definition

For adjustable rate accounts, the last date that the current interest rate changed.

Module Usage

Oracle ALM

Oracle ALM does not reference the LAST_REPRICE_DATE field for either fixed or adjustable-rate records. All rate information in Oracle ALM context is generated in the future from the AS_OF_DATE. CUR_NET_RATE, CUR_GROSS_RATE, and TRANSFER_RATE information from the detail record are referenced to obtain the rate information from the LAST_REPRICE_DATE.

> **Note:** In addition to REPRICE_FREQ, Oracle ALM also uses the ADJUSTABLE_TYPE_CD to determine if a record is adjustable.

Transfer Pricing

1. For an adjustable-rate straight term transfer-priced record, Oracle Funds Transfer Pricing uses the LAST_REPRICE_DATE to identify the assignment date (TP Reference Date). The Interest Rate Code's reference date just before (or equal to) the record's LAST_REPRICE_DATE is used as the transfer pricing yield curve.

   For example, if the record's LAST_REPRICE_DATE = 1/15/2011 and the Rate Management interest rate code (IRC) is defined at monthly intervals and only at month-end, the reference date would be 12/31/2010. The REPRICE_ FREQ is then matched to the same term on the transfer pricing yield curve (IRC) defined in Rate Management.

2. For an adjustable-rate cash flow transfer-priced record, FTP cash flow transfer prices all payments that occur from the LAST_REPRICE_DATE to the NEXT_REPRICE_DATE. In this case, the term and date as defined by these two fields are not used directly to define the transfer rate. They are the starting and ending points

within which FTP applies cash flow transfer pricing.

3. For fixed-rate records, the LAST_REPRICE_DATE and NEXT_REPRICE_DATE are not referenced. ORIGINATION_DATE and MATURITY_DATE are used instead.

Data Verification Requirements and Suggested Defaults

- Oracle Funds Transfer Pricing specific:
  - If REPRICE_FREQ = 0, LAST_REPRICE_DATE = ORIGINATION_DATE
  - If REPRICE_FREQ <> 0, LAST_REPRICE_DATE >= ORIGINATION_DATE
- If REPRICE_FREQ <> 0, LAST_REPRICE_DATE <= AS_OF_DATE
- If REPRICE_FREQ <> 0 and TEASER_END_DATE <= AS_OF_DATE, LAST_REPRICE_DATE + REPRICE_FREQ = NEXT_REPRICE_DATE
- LAST_REPRICE_DATE < NEXT_REPRICE_DATE

Last Reprice Date Balance (LRD_BALANCE)

Definition

Balance as of the record's previous repricing event.

Module Usage

Oracle ALM

Oracle ALM does not reference the LRD_BALANCE field.

Transfer Pricing

Oracle Funds Transfer Pricing cash flow methodologies reference LRD_BALANCE when transfer-pricing adjustable-rate records. LRD_BALANCE holds the balance as of the LAST_REPRICE_DATE.

When transfer pricing adjustable-rate records, LRD_BALANCE is used as the starting balance from the LAST_REPRICE_DATE. When cash flow transfer pricing an adjustable-rate record, FTP calculates the payment events from the LAST_REPRICE_DATE to the NEXT_REPRICE_DATE. If payments (amortization) occurred in between the LAST_REPRICE_DATE and the AS_OF_DATE, the record's existing CUR_PAR_BAL is smaller than it was on the LAST_REPRICE_DATE. Therefore, in order to provide an accurate balance amount at the time of the LAST_ REPRICE_DATE, the LRD_BALANCE is provided. Oracle Funds Transfer Pricing amortizes the LRD_BALANCE from the LAST_REPRICE_DATE until the NEXT_ REPRICE_DATE.

Data Verification Requirements and Suggested Defaults

- If the record is fixed-rate, LRD_BALANCE = ORG_PAR_BAL.
- If the record is adjustable-rate, LRD_BALANCE = balance as of the last reprice date.
- If the balance as of the last reprice date is not known, default LRD_BALANCE to CUR_PAR_BAL.

Liquidity Premium Charge Credit (LIQUIDITY_PREM_CHARGE_CREDIT)

Definition

Oracle FTP calculates the funds charge or credit due to liquidity risk when the user selects the Adjustments, Charge/Credit option in the TP Process – Calculation Elements block.

Module Usage

This field is populated by the FTP process (optionally) if selected in the Calculation Elements block. The TP Engine calculates this amount as follows:

AVG_BOOK_BAL or CUR_BOOK_BAL X LIQUIDITY_PREMIUM_RATE/100 X ACCRUAL BASIS

> **Note:** The balance referenced in the earlier calculation is selected in the FTP Application Preferences screen.

Data Verification Requirements and Suggested Defaults

None

Liquidity Premium Amount (LIQUIDITY_PREMIUM_AMT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Liquidity Adjustment amount in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the Liquidity Adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the amount input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Liquidity Premium Rate (LIQUIDITY_PREMIUM_RATE)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Liquidity Adjustment rate or defines a formula based rate in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the Liquidity Adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the assumption input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Margin (MARGIN)

Definition

MARGIN is the contractual spread in Oracle ALM that is added to the pricing index and results in the financial institution's retention (net) rate, for adjust-able-rate accounts.

Module Usage

Oracle ALM

1. MARGIN is used during cash flow generation.

2. For adjustable-type records, MARGIN is the contractual spread above/below the index that is applied throughout the instrument's life. The financial institution's retention rate (CUR_NET_RATE) is equal to the index that the record is tied to plus a spread, which is defined by the MARGIN field.

3. The events of a repricing involving MARGIN are as follows:

   At a repricing event (or a TEASER_END_DATE) for an adjustable-rate record, the cash flow engine matches the INTEREST_RATE_CD, REPRICE_FREQ and repricing date of the detail record to the Forecast Rates assumption. After matching the rate from the Forecast Rates assumption, the cash flow engine adds the MARGIN amount and applies any tease amount, rate caps/floors, and rounding to derive the rate that is applied to the record.

> **Note:** As explained in the Adjustable Type Code (ADJUSTABLE_ TYPE_CD) section, the cash flow engine does not reference repricing date information for ADJUSTABLE_TYPE_CD = 30 or 50.

4. The repriced rate defined in Step 3 equals the coupon rate that is used for amortization, prepayment, and interest income (financial element 430) calculations.

   > **Note:** If the Oracle ALM Product Characteristics rule has the Model with Gross Rates option switched on, the cash flow engine uses the CUR_ GROSS_RATE and MARGIN_GROSS for amortization and prepayment purposes.

   For more detailed information on this topic see the Current Gross Rate (CUR_ GROSS_RATE), page 6-32 section.

Transfer Pricing

MARGIN is used only when mid-period repricing is selected for spread from Note Rate to compute the rate from a prior period.

Data Verification Requirements and Suggested Defaults

- For administered rate accounts, MARGIN = 0.
- For adjustable rate accounts with no contractual margin, MARGIN = 0.
- For applicable accounts, margin can be positive or negative.

Market Value (MARKET_VALUE_C)

Definition

Market Value, calculated by Oracle ALM, or populated by the institution for use by Oracle Transfer Pricing Option Cost calculations.

Module Usage

Oracle ALM

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output Market Value when the Market Value option is selected from the Calculation Elements block. This result is written to the FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with Market Value for each record if the related option is selected on the Output Preferences block. The instrument record will be updated for the scenario 1 result only.

On the instrument record only, Market Value is stored as:

Market Value / (Current Par Balance * Percent Sold) * 100

Transfer Pricing

Oracle Funds Transfer Pricing does not use MARKET_VALUE_C (from the instrument record), except in Option Costing. When Remaining Term Calculation Mode is selected in the TP Process, and the Target Balance for the subject product leaf is Market Value, Oracle Funds Transfer Pricing Option Cost calculations use the product of CUR_PAR_BAL and MARKET_ VALUE_C as the target balance to which the sum of future discounted cash flows is set equal.

Data Verification Requirements and Suggested Defaults

Because Oracle ALM calculates MARKET_VALUE_C, the default can be set to 0 or NULL for organizations that use Oracle ALM only. Organizations that use Oracle Funds Transfer Pricing option cost calculations may need to populate the column during the extract and record loading process.

Market Value Clean (MARKET_VALUE_CLEAN_C)

Definition

Market Value Clean (or Clean Price) is equal to the standard market value less the accrued interest.

Module Usage

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output Market Value Clean when the Market Value option is selected from the Calculation Elements block. This result is written to the FSI_O_RESULT_MASTER and FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with Market Value Clean for each record if the related option is selected on the Output Preferences block. The instrument record will be updated for the scenario 1 result only.

Data Verification Requirements and Suggested Defaults

None

Matched Spread (MATCHED_SPREAD_C)

Definition

Interest margin on a product, calculated by Oracle Funds Transfer Pricing.

Module Usage

Oracle ALM

Oracle ALM does not use MATCHED_SPREAD_C.

Transfer Pricing

Oracle Funds Transfer Pricing computes MATCHED_SPREAD_C when processing using the standard pricing basis (when the remaining term pricing basis switch in the Standard Transfer Pricing Process is off).

It is calculated as follows:

For assets: Current Net Rate - Transfer Rate

For liabilities: Transfer Rate - Current Net Rate

Data Verification Requirements and Suggested Defaults

None

Matched Spread Alternate (MATCHED_SPREAD_ALT)

Definition

Interest margin on a product, calculated by Oracle Funds Transfer Pricing when an Alternate Rate Output Mapping Rule is attached to the Standard TP process and the target for the alternate Transfer Rate is TRANSFER_RATE_ALT.

Module Usage

Alternate Rate Output Mapping rules allow the user to map transfer pricing results to alternate columns rather than standard output columns. This functionality allows the user to calculate and output more than one Transfer Rate and the related Matched Spread for all records included in the TP process.

Oracle ALM

Oracle ALM does not use MATCHED_SPREAD_ALT.

Transfer Pricing

Oracle Funds Transfer Pricing computes MATCHED_SPREAD_ALT when processing using the standard TP process with an Alternate Rate Output Mapping Rule attached. The Standard TP Process calculates and outputs the MATCHED_SPREAD_ALT as follows:

For assets: Current Net Rate - Transfer Rate Alternate

For liabilities: Transfer Rate Alternate - Current Net Rate

Data Verification Requirements and Suggested Defaults

None

Maturity Amount (MATURITY_AMOUNT)

Definition

This optional column is used when AMRT_TYPE_CD = 850 (Annuity). Maturity Amount is an optional input, and represents the value (principal and accrued interest) of the record at the maturity date for annuities.

Module Usage

Maturity Amount allows the user to define the target ending balance for the account to reach by maturity date.

Data Verification Requirements and Suggested Defaults

None

Maturity Date (MATURITY_DATE)

Definition

Contractual date on which the principal balance of an earning asset or debt instrument is due and payable to the holder.

Module Usage

For both Oracle ALM and Funds Transfer Pricing, MATURITY_DATE defines the final date of payment for the record. The MATURITY_DATE signals the end of processing for a given record.

Oracle ALM

1. As described in the NEXT_PAYMENT_DATE section, the cash flow engine processes a record until the MATURITY_DATE has been reached. This occurs in one of two ways:

   - The PMT_FREQ is rolled until it finally reaches the MATURITY_DATE.

   - The REMAIN_NO_PMTS_C is reduced to **1**, in which case, the record immediately moves to its MATURITY_DATE. For details, including special considerations for payment-patterned records, see Remaining Number of Payments (REMAIN_NO_PMTS_C), page 6-119 and Amortization term (AMRT_ TERM, page 6-20)

2. MATURITY_DATE is considered the final payment date. Any remaining principal balance that was not reduced by a scheduled payment date is paid on the MATURITY_DATE (regularly scheduled principal runoff is financial element 190 or

192 and maturity principal runoff is financial element 195 or 197). Therefore, at the MATURITY_DATE, the record's balance is reduced to **0**.

3. If the record has a balloon amortization, the maturity payment includes the balloon or large final payment.

4. User-Defined Payment Schedules are an exception. Payment Schedules make their final payment on the last day as defined in the PAYMENT_SCHEDULE table. MATURITY_DATE is not referenced.

5. User-Defined Payment Patterns reference the MATURITY_DATE as the final payment date. In addition, in order to calculate the remaining number of payments, if the payment pattern is **split** or if the balance is a new business record, Oracle ALM references the payment pattern payment frequencies and counts the number of payments from the AS_OF_DATE to the MATURITY_DATE (new business records reference the future date of origination rather than the AS_OF_DATE).

Transfer Pricing

1. MATURITY_DATE is referenced for fixed-rate straight term transfer pricing methodologies. When defining the record's transfer pricing term, FTP subtracts the ORIGINATION_DATE from the MATURITY_DATE. The term is then matched to the relevant Interest Rate Code (IRC) in Rate Management. The derived rate is applied to the record as the TRANSFER_RATE.

2. MATURITY_DATE is referenced by cash flow transfer pricing methodologies for both adjustable and fixed-rate records. For adjustable records, FTP transfer prices all cash flows on payment dates within the LAST_REPRICE_DATE and NEXT_REPRICE_DATE. The MATURITY_DATE is used to determine the last payment of a record. Its use is the same as described in the following, for Oracle ALM records.

3. The MATURITY_DATE is also referenced in order to determine the remaining number of payments for user-defined payment pattern records. FTP references the payment pattern payment frequencies and counts the number of payments from the ORIGINATION_DATE to the MATURITY_DATE.

Data Verification Requirements and Suggested Defaults

- For non-term accounts, MATURITY_DATE = AS_OF_DATE + 1 Day. If MATURITY_DATE is defaulted to 01-JAN-1900, the record's balance is not processed by Oracle ALM.

- For term accounts, MATURITY_DATE is required.

- If the record is not past due or defaulted, MATURITY_DATE > AS_OF_DATE.

- MATURITY_DATE = ORIGINATION_DATE + ORG_TERM

- MATURITY_DATE = AS_OF_DATE + REMAIN_TERM_C
- MATURITY_DATE <= NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_C * PMT_FREQ).

  Assume that MATURITY_DATE is less than NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_C * PMT_FREQ). This implies that an account's calculated final payment date differs from MATURITY_DATE (the condition results in what is called a stub payment). In this case, the cash flow engine forces the true last payment to be made on the maturity date. If condition **6** is not met (most likely caused because REMAIN_NO_PMTS_C is too low), the cash flow engine skips scheduled payments.

Minimum Balance (MINIMUM_BALANCE)

Definition

Used for modeling Mortgage Offset Accounts (MOA's), this field represents the minimum balance required in the MOA (before and after potential offsets) for any offsets to occur. This is an input field that should be present in the consumer, mortgage or commercial loan table records that are tied to deposit/savings accounts. If the balance of the MOA is less than the MINIMUM_BALANCE, no offsets occur.

Module Usage

The cash flow engine uses MINIMUM_BALANCE to apply Mortgage Offset Account product parameters to the mortgage offset procedure. Each financial institution can set limit minimums appropriate for each loan record. For an example of the MOA calculation, see EXPECTED_BAL, page 6-49.

Data Verification Requirements and Suggested Defaults

- Minimum Balance > 0 when modeling Mortgage Offset Accounts

Modified Duration (MODIFIED_DURATION_C)

Definition

Calculated by ALM Deterministic Processes (optionally), this output field represents the modified duration.

Module Usage

Both the Static Deterministic and Dynamic Deterministic ALM processes will calculate and output Modified Duration when the Market Value option is selected from the Calculation Elements block. This result is written to FSI_O_RESULT_MASTER and

FSI_O_CONSOLIDATED_MASTER tables for each scenario defined in Forecast Rates. Additionally users can choose to update the instrument data with Modified Duration for each record if the related option is selected on the Output Preferences block. The instrument record will be updated for the scenario 1 result only.

The cash flow engine calculates the average life using the following formula:

$$Modified\ Duration = \frac{Duration}{\left(1 + \frac{Yield\ to\ Maturity}{Payment\ per\ year}\right)}$$

Data Verification Requirements and Suggested Defaults

None

Negative Amortization Amount (NEG_AMRT_AMT)

Definition

The total amount of principal added to outstanding principal, resulting from payments which were not large enough to cover interest due.

Module Usage

The Oracle ALM and Funds Transfer Pricing adjustable-type cash flow methodologies reference NEG_AMRT_AMT in calculating the current payment for negative amortization-type accounts. This is relevant only for adjustable-rate accounts with AMRT_TYPE_CD = 600.

1. In a negatively amortizing record, the CUR_PAYMENT is less than the principal and interest that is due on the payment date. The interest portion that is not included in the payment goes to two places. It is added to the NEG_AMRT_ AMT field and is added back to the principal amount. Because the NEG_ AMRT_AMT balance is already included in the outstanding principal balance, NEG_AMRT_AMT is not explicitly used when the cash flow engine fully re-amortizes the account.

2. NEG_AMRT_AMT is used by the cash flow engine to keep track of negative amortization separately from non-negative amortization (normal) principal balance. It is separate for two reasons:

   - Because the cash flow engine pays down the negatively amortized portion before the principal portion, a separation of the two amounts must be done to enable the application to identify what portion of the principal balance is negatively amortized.

- When calculating the current payment, the cash flow engine uses NEG_ AMRT_AMT in its check to see if NEG_AMRT_LIMIT has been exceeded. For more details, see the Negative Amortization Limit (NEG_AMRT_LIMIT), page 6-78.

Following is the process of events in regards to NEG_AMRT_AMT and related negative amortization fields:

1. Record is currently negatively amortizing because the payment amount, as defined by CUR_PAYMENT, is not enough to cover the principal and interest portion. The unpaid interest at each payment date goes into the NEG_AMRT_ AMT field and back into the principal.

2. While calculating a payment event (payment date), if the cash flow engine calculates negative principal runoff, it additionally checks the negative amortization limit (NEG_AMRT_LIMIT) in order to ensure that the current NEG_AMRT_ AMT is not exceeding its limit. NEG_AMRT_LIMIT is defined as a percentage of the original principal balance. If NEG_AMRT_AMT is exceeding this limit, the cash flow engine recalculates the payment amount in order to fully amortize the instrument.

3. However, when deriving the recalculated payment amount after a NEG_ AMRT_LIMIT has been exceeded, the cash flow engine also applies payment decrease/increase limits per period (PMT_DECR_CYCLE, PMT_INCR_CYCLE) and payment decrease/increase limits for the life of the record (PMT_DECR_ LIFE, PMT_INCR_LIFE). Because these fields limit how much the CUR_ PAYMENT can be changed, it is possible that the record continues to negatively amortize even after a NEG_AMRT_LIMIT has been exceeded. If negative amortization does continue, the NEG_AMRT_AMT continues to grow.

4. The cash flow engine also attempts to recalculate the negatively amortizing payment amount on a PMT_ADJUST_DATE. Just like a payment recalculation for a NEG_AMRT_ LIMIT, a payment recalculation on the PMT_ADJUST_DATE takes into account the effects of payment decrease/increase limits per period and payment decrease/increase limits for the life of the record. This provides for additional negative amortization to occur even after the PMT_ADJUST_DATE has recalculated the payment amount. PMT_ADJUST_DATE is incremented forward by the PMT_CHG_FREQ field until the maturity.

5. In addition to PMT_ADJUST_DATE and NEG_AMRT_LIMIT, the record can experience a payment recalculation on the negative amortization equalization date (NEG_AMRT_EQ_DATE). On this date, the record's CUR_PAYMENT will be fully re-amortized. NEG_AMRT_EQ_DATE will ignore payment decrease/increase limits per period and payment decrease/increase limits for the life of the record. Therefore, after the payment recalculation of a NEG_ AMRT_EQ_DATE, the record

will no longer be negatively amortizing and the NEG_AMRT_EQ_DATE is incremented forward by the NEG_AMRT_EQ_FREQ until maturity.

Data Verification Requirements and Suggested Defaults

- NEG_AMRT_LIMIT must be within the range of 0 to 100.
- For AMRT_TYPE_CD <> 600, NEG_AMRT_AMT = 0.
- If AMRT_TYPE_CD = 600, 0 <= NEG_AMRT_AMT <= NEG_AMRT_ LIMIT/100 * ORG_PAR_BAL.
- If not applicable, default to 0.

Negative Amortization Equalization Date (NEG_AMRT_EQ_DATE)

Definition

The next date that a negative amortization-type account will fully re-amortize, regardless of payment caps and floors.

Module Usage

The Oracle ALM and Funds Transfer Pricing adjustable-type cash flow methodologies reference NEG_AMRT_EQ_DATE when calculating the current payment for negative amortization-type accounts. NEG_AMRT_EQ_DATE is relevant only for adjustable-rate accounts with AMRT_TYPE_CD = 600.

1. On the NEG_AMRT_EQ_DATE, a negatively amortizing record's payment will be recalculated. On this date, the record's CUR_PAYMENT will be fully re-amortized. NEG_AMRT_EQ_DATE will ignore payment decrease/increase limits per period and payment decrease/increase limits for the life of the record. Therefore, after the payment recalculation of a NEG_AMRT_EQ_DATE, the record will no longer be negatively amortizing.
2. NEG_AMRT_EQ_DATE is incremented forward by the NEG_AMRT_EQ_FREQ until the maturity date is reached.
3. For an explanation of NEG_AMRT_EQ_DATE's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

> **Note:** If NEG_AMRT_EQ_FREQ = 0, once the modeling date is past the NEG_AMRT_EQ_DATE, the cash flow engine does not attempt to re-amortize the negative amortized amount.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD <> 600, NEG_AMRT_EQ_DATE = 01-JAN-1900.
- If AMRT_TYPE_CD = 600, NEG_AMRT_EQ_DATE > ORIGINATION_DATE.
- If AMRT_TYPE_CD = 600, NEG_AMRT_EQ_DATE < MATURITY_DATE.

Negative Amortization Equalization Frequency (NEG_AMRT_EQ_FREQ)

Definition

Used in conjunction with NEG_ AMRT_EQ_MULT to define the frequency that negatively amortizing accounts are fully re-amortized.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference NEG_AMRT_EQ_FREQ in calculating the current payment for negative amortization-type accounts. This is relevant only for adjustable-rate accounts with AMRT_TYPE_CD = 600.

1. From the NEG_AMRT_EQ_DATE, the cash flow engine rolls forward by the NEG_AMRT_ EQ_FREQ until the maturity date.
2. At each instance of a NEG_AMRT_EQ_FREQ, the cash flow engine recalculates the payment as it did for the NEG_AMRT_EQ_DATE. On these roll dates, the record's CUR_ PAYMENT will be fully re-amortized. NEG_AMRT_EQ_FREQ will ignore payment decrease/increase limits per period and payment decrease/increase limits for the life of the record. Therefore, after the payment recalculation of a NEG_AMRT_EQ_FREQ, the record will no longer be negatively amortizing.
3. If NEG_AMRT_EQ_FREQ = 0, once the modeling date is past the NEG_AMRT_ EQ_DATE, the cash flow engine will not attempt to re-amortize the negative amortized amount. In this case, any negative amortized balance will balloon at maturity.
4. For an explanation of NEG_AMRT_EQ_FREQ's relationship with other related negative amortization fields, see NEG_AMRT_AMT, page 6-74.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD = 600, NEG_AMRT_EQ_FREQ must be either 0 or a positive value. NEG_AMRT_EQ_FREQ cannot be negative.
- If AMRT_TYPE_CD <> 600, NEG_AMRT_EQ_FREQ = 0.

- If AMRT_TYPE_CD = 600, 0 <= NEG_AMRT_EQ_FREQ < ORG_TERM.
- If AMRT_TYPE_CD = 600, AS_OF_DATE < NEG_AMRT_EQ_DATE <= MATURITY_DATE.
- Validation of NEG_AMRT_EQ_FREQ should always be done in conjunction with NEG_AMRT_EQ_MULT.

Negative Amortization Equalization Frequency Multiplier (NEG_AMRT_EQ_MULT)

Definition

Used in conjunction with NEG_AMRT_EQ_FREQ to define the frequency that negatively amortizing accounts are fully re-amortized.

Module Usage

This field is the multiplier of the NEG_AMRT_EQ_FREQ field. It is used in conjunction with NEG_AMRT_EQ_FREQ to define the frequency that negatively amortizing accounts are fully re-amortized. Oracle ALM and Funds Transfer Pricing cash flow calculations reference NEG_AMRT_EQ_MULT when recalculating the current payment as defined under the NEG_AMRT_EQ_FREQ section. NEG_AMRT_EQ_MULT determines the units (Months, Days or Years) of NEG_ AMRT_EQ_FREQ.

Data Verification Requirements and Suggested Defaults

- Valid values are:
  - D - Days
  - M - Months
  - Y - Years
- For more information on NEG_AMRT_EQ_ MULT validation, refer to the validation discussion for Negative Amortization Equalization Frequency (NEG_AMRT_EQ_FREQ, page 6-77).
- Suggested default is **M**.

Negative Amortization Limit (NEG_AMRT_LIMIT)

Definition

Maximum negative amortization allowed as a percentage of the original balance.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference NEG_AMRT_LIMIT when determining if the NEG_AMRT_AMT is exceeding its defined limits. This is relevant only for adjustable-rate accounts with AMRT_ TYPE_D = 600.

1. While calculating a payment event (payment date), if the cash flow engine calculates negative principal runoff, the CFE also checks the negative amortization limit (NEG_AMRT_LIMIT) in order to ensure that the current NEG_AMRT_ AMT is not exceeding its limit. NEG_AMRT_LIMIT is defined as a percentage of the original principal balance.

   For example, NEG_AMRT_LIMIT = 25 means that the negative amortization amount should never exceed 25% of the original principal balance (principal balance should never exceed 125% of the original balance). The formula for this check is:

   (-1 * calculated (negative) principal runoff + negative amortization balance > NEG_AMRT_LIMIT/100 * ORG_PAR_BAL)

   If NEG_AMRT_AMT is exceeding this limit, cash flow engine recalculates the payment amount in order to fully amortize the instrument.

2. When deriving the recalculated payment amount after a NEG_AMRT_LIMIT has been reached, the cash flow engine also applies payment decrease/increase limits per period (PMT_DECR_CYCLE, PMT_INCR_CYCLE) and payment decrease/increase limits for the life of the record (PMT_DECR_LIFE, PMT_ INCR_LIFE). Because these fields limit how much the CUR_PAYMENT can be changed, it is possible that the record will continue to negatively amortize even after a NEG_AMRT_LIMIT has been exceeded. If negative amortization does continue, the NEG_AMRT_AMT will continue to grow.

3. For an explanation of NEG_AMRT_LIMIT's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD <> 600, NEG_AMRT_LIMIT = 0.
- If AMRT_TYPE_CD = 600, NEG_AMRT_LIMIT >= 0.

Net Margin Code (NET_MARGIN_CD)

Definition

NET_MARGIN_CD defines the relationship between CUR_GROSS_RATE and

CUR_NET_RATE for the cash flow engine.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference NET_MARGIN_CD when calculating a forecasted interest rate. NET_ MARGIN_CD has the following two, valid values:

- Floating Net Rate = **0**
- Fixed Net Rate = **1**

These are described as follows:

Floating Net Rate (NET_MARGIN_CD = 0): This is the default value for the Net Margin Code. If the detail data's NET_MARGIN_CD field is set to **0** (floating), the Existing Business record will reprice at the relevant reprice dates, as described under the NEXT_REPRICE_DATE section. If the Product Characteristics' Net Margin Flag is set to Floating Net Rate, the New Business record will also reprice at the relevant reprice dates. Interest Income (FE 430) will be based off of the CUR_NET_ RATE.

Fixed Net Rate (NET_MARGIN_CD = 1): This setting is used by financial institutions that maintain the loans of other financial institutions. For example, Bank A may service (operate and process) the loans of Bank B. Bank B pays Bank A a fixed spread or margin as payment for maintaining the loans. Because Bank A receives a guaranteed fixed spread, only Bank B gains or loses when the actual loan reprices. For this reason, if the record reprices, Bank A should not experience any change in interest income.

If the detail record's NET_MARGIN_CD field is set to **1** (fixed) and the Model With Gross Rates switch is turned on in the ALM Product Characteristics rule, the existing business record does not reprice even if the record is an adjustable-rate product (CUR_NET_RATE does not reprice). It is assumed that the rate received by the bank (Bank A) equals the fixed spread that the bank is receiving as payment for maintaining the loans. The record's CUR_NET_RATE field represents this fixed spread and is used for interest income (financial element 430) calculations while the record's CUR_GROSS_RATE is used for prepayments and amortization.

With regards to new business (business originating beyond the AS_OF_DATE), if the Product Characteristics rule's Net Margin Flag is set to Fixed Net Rate and the Model With Gross Rates switch is turned on, the rate used to derive new business interest income is taken from the Pricing Margin rule Net Margin and not from the Forecast Rates. This is because it is assumed that the Pricing Margin assumption contains the fixed spread that the bank is receiving as payment for maintaining the loans. The Net Margin from the Pricing Margin rule is used for interest income calculations while the gross rate, which is derived from the Forecast Rates assumption, is used for prepayment and amortization calculations.

> **Note:** The CUR_GROSS_RATE is used for amortization and prepayment calculation purposes and reflects the correct repriced rate

for all such calculations.

If the NET_MARGIN_CD is set to Fixed Net Rate, but the Model with Gross Rates switch is not turned on, the cash flow engine treats the records as if they were Floating Net Rate.

Data Verification Requirements and Suggested Defaults

- NET_MARGIN_CD must be equal to **0** or **1**.
- For fixed-rate accounts and adjustable accounts that reprice, NET_MARGIN_ CD = 0 (default setting).
- For adjustable-rate accounts that represent records being serviced for a fixed fee, set NET_MARGIN_CD = 1.

Next Payment Date (NEXT_PAYMENT_DATE)

Definition

Due date of the next scheduled payment. Otherwise known as cash flow date or date of runoff.

Module Usage

NEXT_PAYMENT_DATE is used to define the next scheduled (forecasted) payment date.

Oracle ALM

Processing order in regards to NEXT_PAYMENT_DATE is as follows:

1. From the AS_OF_DATE, the first cash flow event processed by the cash flow engine is the NEXT_PAYMENT_DATE. The cash flow engine references the NEXT_PAYMENT_DATE for the first forecasted payment date only. This applies to payment patterns (relative and absolute), but not for Payment Schedules. See the following details.
2. From the NEXT_PAYMENT_DATE, the cash flow engine increments forward by the PMT_ FREQ until the MATURITY_DATE is reached.

> **Note:** For User-Defined Payment Schedules or Patterns, the cash flow engine does not reference the PMT_FREQ field. Instead, it references the schedule or pattern information to define the additional forecasted payment dates.

3. On the payment date the cash flow engine calculates the interest payments, principal payments, current deferred payments, prepayments, and negative amortization, if applicable. For an adjustable-type record where REPRICE_FREQ < PMT_ FREQ, the cash flow engine applies only the last repriced rate for the purpose of payment calculation. If the record is not an AMRT_TYPE_CD 700 (non-amortizing), AMRT_TYPE_CD 850 (Annuity) or AMRT_TYPE_CD 600 (negatively amortizing), the record's principal balance is reduced at each payment date.

4. As each payment is made, the cash flow engine reduces the REMAIN_NO_PMTS_C by 1. If the newly calculated REMAIN_NO_PMTS_C = 1, the next payment date is set to MATURITY_DATE.

   > **Note:** The use of REMAIN_NO_PMTS_C varies depending on record type. See the Remaining Number of Payments (REMAIN_ NO_PMTS_C) , page 6-119 and Amortization Term (AMRT_TERM), page 6-20 sections for details, including special considerations for payment-patterned records.

5. MATURITY_DATE is the final payment date. If the record's principal was not reduced by the payment amounts, the remaining principal balance is paid on the MATURITY_DATE.

   > **Note:** For Payment Schedules, the cash flow engine does not use the NEXT_ PAYMENT_DATE field. For these records, it makes the next payment on the first date in the schedule after the AS_OF_DATE. However, for Payment Schedules and User-Defined Payment Patterns, the NEXT_PAYMENT_DATE from the detail record should correspond to the next defined payment date after the AS_ OF_DATE in the Schedule or Pattern interface.

Transfer Pricing

NEXT_PAYMENT_DATE is used for adjustable-rate cash flow transfer-priced records. In defining the transfer rate for an adjustable rate record, the cash flow engine produces cash flows for each payment date from the LAST_REPRICE_DATE to the NEXT_ REPRICE_DATE. In order to define all the payment events within this period, it rolls back from the NEXT_PAYMENT_DATE by the PMT_FREQ until just before the LAST_REPRICE_DATE. From this calculated payment date, the cash flow engine again rolls forward by the PMT_FREQ, but this time cash flows are produced. The cash flows produced are used by one of the four cash flow transfer pricing methodologies in order to derive the transfer rate.

Data Verification Requirements and Suggested Defaults

Required Conditions:

- NEXT_PAYMENT_DATE > AS_OF_DATE
- Also included in NEXT_PAYMENT_DATE is the modeling of past due, delinquent, or non-term accounts, if they are to be processed by the cash flow engine.

  For example, if the NEXT_PAYMENT_DATE is defaulted to 01-JAN-1900, the cash flow engine rolls the record by PMT_FREQ from that date until the maturity date.

  > **Note:** This consumes considerable processing time, so should be avoided.

- NEXT_PAYMENT_DATE <= MATURITY_DATE
- If REMAIN_NO_PMTS_C > 1, NEXT_PAYMENT_DATE < MATURITY_DATE
- If REMAIN_NO_PMTS_C = 1, NEXT_PAYMENT_DATE = MATURITY_DATE
- MATURITY_DATE <= NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_C * PMT_FREQ)
- If the amortization type is an absolute pattern or payment schedule, the payment dates are predefined. The NEXT_PAYMENT_DATE on the detail record should always correspond to the relevant (next payment date after the AS_OF_DATE) predefined payment date information of the Absolute Pattern interface or PAYMENT_SCHEDULE table.
- Suggested default:

  If next payment date is unknown, set to either:

  - AS_OF_DATE + PMT_FREQ , or
  - MATURITY_DATE

Next Repricing Date (NEXT_REPRICE_DATE)

Definition

Date of next scheduled interest rate change for adjustable rate accounts.

Module Usage

NEXT_REPRICE_DATE defines the first forecasted repricing event from the AS_OF_DATE.

Oracle ALM

In Oracle ALM the processing and use of NEXT_REPRICE_DATE is as follows:

1. If the record is defined as ADJUSTABLE_TYPE_CD = 250 and REPRICE_FREQ > 0, the cash flow engine references NEXT_REPRICE_DATE when calculating the first forecasted interest rate change.

   > **Note:** AMRT_TYPE_CD definition does not impact whether the record is adjustable or not.
   >
   > If ADJUSTABLE_TYPE_CD = 30 or 50, the cash flow engine does not reference the NEXT_ REPRICE_DATE. For more details, see Adjustable Type Code (ADJUSTABLE_TYPE_CD), page 6-12.

2. The cash flow engine rolls forward from NEXT_REPRICE_DATE to define the record's remaining forecasted Reprice Dates. Rolling by the REPRICE_FREQ continues until MATURITY_DATE.

3. In defining the customer rate, on each reprice date, the cash flow engine matches the record's INTEREST_RATE_CD, the reprice date and the REPRICE_FREQ to the appropriate term point on the forecasted Interest Rate Code (IRC) in the Forecast Rates assumption rule. To this derived rate, the cash flow engine adds the MARGIN (or MARGIN_ GROSS, if applicable) to arrive at the full note rate.

4. The cash flow engine then applies interest rate rounding and periodic/lifetime rate caps/floors. If the record's TEASER_END_DATE is less than or equal to the AS_OF_DATE, the cash flow engine applies the calculated forecasted rate to the record. Otherwise it applies the defined teased rate.

5. On a payment date, the forecasted rate derived on the repricing date is used for recalculating payment amounts.

   > **Note:** If multiple reprice dates exist within one payment period (that is, if REPRICE_FREQ < PMT_FREQ), only the forecasted rate of the reprice date immediately preceding the payment date is used for payment calculation purposes. Oracle ALM stores **Before Reprice** and **After Reprice**, **Gross Rates** and **Net Rates** as financial elements 260 - 290.

6. If the Process with Transfer Rates option has been selected in the ALM Deterministic Process under Calculation Elements, the cash flow engine produces transfer rates by referring to the Transfer Pricing Rule attached to the ALM Process. No rounding, rate caps/floors or tease checks are applied to the calculated transfer rate.

Transfer Pricing

1. NEXT_REPRICE_DATE is used by adjustable-rate straight term transfer-priced records when using the Remaining Term Pricing Basis. The AS_OF_DATE and

NEXT_REPRICE_DATE define the term of the transfer pricing period. This term is matched to the relevant Interest Rate Code (IRC) in Rate Management to derive a transfer rate.

2. Adjustable-rate cash flow transfer-priced records use LAST_REPRICE_DATE and NEXT_REPRICE_DATE as the starting and ending points of the transfer-pricing period. In order to define all the payment events within this period, FTP rolls back from the NEXT_PAYMENT_DATE by the PMT_FREQ until just after the LAST_REPRICE_DATE. From this calculated payment date, FTP again rolls forward by the PMT_FREQ until just before the NEXT_ REPRICE_DATE. As FTP rolls forward, cash flows are produced. The cash flows produced are used by one of the four cash flow transfer pricing methodologies in order to derive the transfer rate.

For additional information also see Teaser-rate End Date (TEASER_END_ DATE), page 6-123.

Data Verification Requirements and Suggested Defaults

1. For fixed-rate accounts, NEXT_REPRICE_DATE = MATURITY_DATE.
2. For administered rate accounts and floating rate accounts, use ADJUSTABLE_ TYPE_CD 30 or 50, which does not reference NEXT_REPRICE_DATE. Set the default to NEXT_REPRICE_DATE = NEXT_PAYMENT_DATE or MATURITY_ DATE.
3. If ADJUSTABLE_TYPE_CD = 250 and repricing information is available, then:
   - NEXT_REPRICE_DATE > AS_OF_DATE
   - NEXT_REPRICE_DATE <= MATURITY_DATE

Offset Percentage (OFFSET_PERCENT)

Definition

Used in the Mortgage Offset Account (MOA) process, the OFFSET_PERCENT is applied to the MOA expected balance at every payment date to calculate the remaining balance after offset for interest calculation purposes.

Module Usage

The cash flow engine calculates the interest due on each loan payment date applying the OFFSET_PERCENT to the expected MOA loan balance assuming that the MOA meets minimum balance criteria. The cash flow engine calculates loan interest due using offsets for MOA for consumer, mortgage, and commercial data table instruments that have these columns populated. For more information see EXPECTED_BAL, page 6-49.

Data Verification Requirements and Suggested Defaults

For Mortgage Offset Accounts, OFFSET_PERCENT > 0

Original Market Value (ORG_MARKET_VALUE)

Definition

The market value of the instrument at origination, expressed as a percentage of the Original Par Balance.

Module Usage

Transfer Pricing

Oracle Funds Transfer Pricing uses ORG_MARKET_VALUE when calculating Option Costs.

When Standard Calculation Mode is selected in the Stochastic TP Process, and the Target Balance for the subject product leaf is Market Value, Oracle Funds Transfer Pricing Option Cost calculations use the product of ORG_PAR_BAL and ORG_MARKET_VALUE as the target balance to which the sum of future discounted cash flows is set equal.

Data Verification Requirements and Suggested Defaults

None

Original Payment Amount (ORG_PAYMENT_AMT)

Definition

The original payment amount at the date of origination.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference ORG_PAYMENT_AMT when referencing the payment amount at the time of the record's origination.

Oracle ALM

For User-Defined Payment Patterns with the payment method defined as **% Original Payment**, the cash flow engine uses the record's ORG_PAYMENT_AMT for all payments beyond the first forecasted one. The first forecasted one uses CUR_PAYMENT. For all other payment pattern payment methods, the cash flow engine uses the ORG_PAYMENT_ AMT.

Transfer Pricing

1. Cash flow transfer pricing methodologies for fixed-rate records use ORG_

PAYMENT_AMT as the payment amount for amortization purposes. For a fixed-rate record, the cash flow engine rolls forward from the ORIGINATION_DATE by PMT_ FREQ when defining payment dates up to the record's MATURITY_DATE. From origination, the cash flow engine amortizes the original balance (ORG_PAR_BAL) by the ORG_PAYMENT_AMT. The precise method of this amortization depends on the AMRT_TYPE_CD.

2. ORG_PAYMENT_AMT is also used if the User-Defined Payment Pattern payment method **% Original Payment** is designated.

Oracle ALM and Funds Transfer Pricing

ORG_PAYMENT_AMT is also used in determining if the NEG_AMRT_AMT is exceeding its defined limits. This is relevant only for adjustable-rate accounts where AMRT_TYPE_CD = 600.

1. For negative amortization-type accounts, the cash flow engine uses ORG_ PAYMENT_AMT in determining whether a recalculated payment increase or decrease exceeds PMT_INCR_LIFE and PMT_DECR_LIFE.

2. On a recalculation date caused by a NEG_AMRT_LIMIT or PMT_ADJUST_ DATE, the cash flow engine recalculates the payment amount in order to create a fully amortized record. After the recalculation, the cash flow engine references payment life increases/decreases fields (PMT_INCR_LIFE, PMT_DECR_LIFE). These fields limit the amount that the recalculated payment amount can change from the original payment amount (ORG_PAYMENT_AMT).

   For example, if PMT_INCR_LIFE = 25.00, the recalculated payment amount is not allowed to increase by more than 25% of the ORG_PAYMENT_AMT.

   For additional information, see Payment Increase Limit - Life (PMT_INCR_LIFE), page 6-104 and Payment Decrease Limit - Life (PMT_DECR_LIFE), page 6-99.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD <> 600 ORG_PAYMENT_AMT = 0.

- If AMRT_TYPE_CD = Payment Pattern which references ORG Payment, ORG_PAYMENT_AMT should be a valid non-zero value.

- If adjustable-rate and AMRT_TYPE_CD = 600, ORG_PAYMENT should be a valid non-zero value.

- ORG_PAYMENT may be validated using the following formula:

  ORG_PAYMENT = (ORG_BOOK_BAL* (CUR GROSS RATE/((12/PMT_ FREQ[in months]) *100))) / (1 - ((1+(CUR_GROSS_RATE /((12/ PMT_ FREQ [in months])*100))) ^-(ORG_TERM/PMT_FREQ [in months])))

- If ORG_PAYMENT_AMT is unknown, default to CUR_PAYMENT.

Original Par Balance (ORG_PAR_BAL)

Definition

Represents the starting balance from which all fixed rate transfer pricing cash flows are generated, including principal run-off, prepayments, and interest cash flows.

Module Usage

Oracle ALM

ORG_PAR_BAL is referenced when processing User-Defined Payment Patterns in which the payment method is **%Original Balance**. When this payment pattern is selected, the cash flow engine applies payment amounts throughout the life of the loan that are a percentage of the value in the detail record's ORG_PAR_BAL field.

> **Note:** If **% Original Balance** is selected, the first forecasted cash flow references the CUR_PAYMENT, but all remaining cash flows use the ORG_PAR_BAL.

Transfer Pricing

Cash flow transfer pricing methodologies for fixed-rate records use ORG_PAR_BAL as the starting balance for all cash flow generation. For the treatment of adjustable rate records, see Last Reprice Date Balance (LRD_BALANCE), page 6-65.

The Oracle cash flow engine bases interest cash flows and principal runoff on ORG_PAR_BAL when transfer pricing cash flow methodology fixed-rate accounts. During processing, the cash flow engine rolls forward from the ORIGINATION_DATE by PMT_FREQ when defining payment dates up until the record's MATURITY_DATE. From origination, the cash flow engine amortizes the original balance (ORG_PAR_BAL) by the ORG_PAYMENT_AMT. The precise method of this amortization depends on the AMRT_TYPE_CD.

When Standard Calculation Mode is selected in the TP Process, and the Target Balance for the subject product leaf is Par Balance, Oracle Funds Transfer Pricing Option Cost calculations use ORG_PAR_BAL as the target balance to which the sum of future discounted cash flows is set equal.

Data Verification Requirements and Suggested Defaults

- ORG_PAR_BAL requires a valid balance for all accounts.

- ORG_PAR_BAL = ORG_BOOK_BAL - DEFERRED_ORG_BAL.

- ORG_PAR_BAL must have the same sign as ORG_PAYMENT_AMT and CUR_PAYMENT.

  For the transfer pricing of fixed-rate instruments, the original balance should be populated. If REPRICE_FREQ = 0, ORG_PAR_BAL <> 0.

- Original balance on Rule of 78 instruments should not be greater than the current balance. If AMRT_TYPE_CD = 710, ORG_PAR_BAL should be less than CUR_PAR_BAL.

Original Term (ORG_TERM)

Definition

Used in conjunction with ORG_TERM_MULT to define the contractual term at origination date.

Module Usage

The ORG_TERM of the instrument is referenced by Oracle Funds Transfer Pricing and Oracle ALM as the period from ORIGINATION_DATE to MATURITY_ DATE.

Transfer Pricing

ORG_TERM is referenced by Oracle Funds Transfer Pricing when calculating cash flows for fixed-rate cash flow methodologies. Additionally, FTP Adjustment Rules refer to Org Term when the Reference Term is set to "Original Term".

Transfer Pricing and ALM

1. Oracle Funds Transfer Pricing and Oracle ALM cash flow methodologies use ORG_TERM when calculating the current payment for adjustable rate accounts and in transfer pricing fixed rate accounts. For adjustable rate accounts, the cash flow engine compares ORG_TERM against AMRT_TERM, checking to see if the account is a balloon-type account.

2. If ORG_TERM < AMRT_TERM, the cash flow engine recognizes the record as a balloon. The cash flow engine amortizes the outstanding principal balance over the calculated number of payments, based upon the amortization maturity date. For a detailed explanation, see the Amortization Term (AMRT_TERM), page 6-20 section

Data Verification Requirements and Suggested Defaults

- ORG_TERM > 0
- ORG_TERM <= ISSUE_TERM
- ORG_TERM <= AMRT_TERM
- ORG_TERM >= REPRICE_FREQ
- ORG_TERM >= PMT_FREQ

- ORG_TERM >= REMAIN_TERM_C
- ORG_TERM + ORIGINATION_DATE = MATURITY_DATE

> **Note:** Validation of ORG_TERM should always be done in conjunction with ORG_TERM_MULT. Setting ORG_TERM to **0** may result in Oracle ALM processing errors.

Original Term Multiplier (ORG_TERM_MULT)

Definition

Used in conjunction with ORG_TERM to define the contractual term at origination date.

Module Usage

The cash flow engine references ORG_TERM_MULT when calculating current payments for adjustable rate accounts and in transfer pricing fixed-rate accounts. ORG_TERM MULT determines the units (Months, Days or Years) of ORG_TERM.

Data Verification Requirements and Suggested Defaults

- Valid values are:
  - D - Days
  - M - Months
  - Y - Years
- For non-term accounts, default to **M**.
- For information on ORG_TERM_MULT validation, refer to the validation discussion for Original Term (ORG_TERM), page 6-89.

Origination Date (ORIGINATION_DATE)

Definition

The date the current institution originated or acquired the instrument.

Module Usage

Both Oracle ALM and Funds Transfer Pricing reference the ORIGINATION_DATE as the start date of the record. For additional information, see Issue Date (ISSUE_DATE), page 6-62.

Prepayment assumptions also reference ORIGINATION_DATE. Separate assumptions can be defined for ORIGINATION_DATE ranges.

Additionally, when using a Prepayment Models:

- If the prepayment rate is driven by the Expired Term, the ORIGINATION_ DATE is used to determine the age of the instrument using the following formula:

  (ROUND(Current Bucket Date - ORIGINATION_DATE)/30.42, 0)

- If the prepayment rate is driven by the ORIGINATION_DATE and the instrument is still in its tease period (that is, TEASE_END_DATE > Current Bucket Date), then the REPRICE_FREQ is calculated as:

  (ROUND(TEASE_END_DATE - ORIGINATION_DATE)/30.42,0)

Oracle ALM

1. ORIGINATION_DATE is used to determine the last payment date. If the LAST_ PAYMENT_DATE is erroneously before the ORIGINATION_DATE, the cash flow engine uses the ORIGINATION_DATE value instead. This is used during interest income calculations where the LAST_PAYMENT_DATE is referenced for the first forecasted interest income cash flow (financial element 430). For details, see Last Payment Date (LAST_PAYMENT_DATE), page 6-63.

2. User-Defined amortization payment patterns that are defined as **% Current Payment** and have multiple payment frequencies (as defined in the Payment Pattern user interface) use ORIGINATION_DATE when calculating amortization in order to determine the payment amount. This calculation is defined in the AMRT_ TERM, page 6-20 section.

Transfer Pricing

1. Fixed rate cash flow transfer priced records reference ORIGINATION_DATE in order to calculate the payment dates for amortization purposes.

   > **Note:** For adjustable rate accounts, transfer pricing cash flow calculations begin on LAST_REPRICE_DATE.

   When defining the record's payment dates, the cash flow engine starts from the record's ORIGINATION_DATE and rolls forward by PMT_FREQ until the MATURITY_ DATE is reached. The cash flow engine bases interest cash flows and principal runoff on ORG_PAR_BAL when transfer pricing cash flow methodology fixed rate accounts. From ORIGINATION_DATE, the cash flow engine amortizes the original balance (ORG_PAR_BAL) by the ORG_PAYMENT_AMT. The precise method of this amortization depends on the AMRT_TYPE_CD.

2. Straight term methodology references ORIGINATION_DATE when defining the transfer pricing term that is matched to the term on the yield curve (Rate

Management Interest Rate Code). For fixed-rate instruments, the term defined by (MATURITY_DATE - ORIGINATION_DATE) is matched to the relevant Interest Rate Code (IRC).

For adjustable rate instruments in their tease period, the term is figured as the (TEASE_END_DATE - ORIGINATION_DATE). The transfer pricing assignment date for the IRC is also determined by the ORIGINATION_DATE of the record. That is, the date of the yield curve (IRC) is matched to the date of the record's origination.

> **Note:** IF TP_EFFECTIVE_DATE is populated, it will be used in place of ORIGINATION_DATE to determine the effective date for lookup of historical rates from the TP Yield Curve.

3. If the record is transfer-priced using a **Spread From Interest Rate Code** or **Redemption Curve** methodology, the option of choosing the IRC's assignment date is available. If the **Origination Date** is chosen as the assignment date, or if the assignment date is the **Last Repricing Date** and the instrument is fixed rate, the date of the IRC used for transfer rate calculations is the same as the detail record's ORIGINATION_DATE. If an IRC of the same date does not exist the cash flow engine uses the closest preceding date's yield curve information.

4. For records that reference the User-Defined Payment Patterns, the cash flow engine derives the remaining number of payments by counting the number of payments from the ORIGINATION_DATE to the MATURITY_DATE.

Data Verification Requirements and Suggested Defaults

- For all accounts, a valid ORIGINATION_DATE is required.
  - ORIGINATION_DATE >= ISSUE_DATE
  - ORIGINATION_DATE <= AS_OF_DATE (unless future originations booked on system)
  - ORIGINATION_DATE <= LAST_REPRICE_DATE
- For term accounts only:
  - ORIGINATION_DATE + ORG_TERM = MATURITY_DATE
  - ORIGINATION_DATE <= TEASER_END_DATE (if TEASER_END_DATE is valid)

Other Adjustment Amount Alternate (OTHER_ADJ_AMOUNT_ALT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies an Other Adjustment amount in the TP Adjustment Rule and additionally attaches an Alternate Rate Output Mapping rule to the TP Process. The **Other Adjustment** type can be used for defining any type of TP Add-on Rate that does not fit into one of the other predefined Adjustment Type classifications.

Module Usage

This field is related to FTP Adjustment Rules. When the **Other Adjustment** adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the amount input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Other Adjustment Charge Credit (OTHER_ADJ_CHARGE_CREDIT)

Definition

Oracle FTP calculates the funds charge or credit due to the **Other Adjustment** adjustment type when the user selects the Adjustments, "Charge/Credit" option in the TP Process – Calculation Elements block.

Module Usage

This field is populated by the FTP process (optionally) if selected in the Calculation Elements block. The TP Engine calculates this amount as follows:

AVG_BOOK_BAL or CUR_BOOK_BAL X OTHER_ADJUSTMENTS_RATE/100 X ACCRUAL BASIS

> **Note:** The balance referenced in the earlier calculation is selected in the FTP Application Preferences screen

Data Verification Requirements and Suggested Defaults

None

Other Adjustment Rate Alternate (OTHER_ADJ_RATE_ALT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies an **Other Adjustment** rate or defines a formula based rate in the TP Adjustment Rule and additionally attaches an Alternate Rate Output Mapping rule to the TP Process. The **Other Adjustment** type can be used for defining any type of TP Add-on Rate that does not fit into one of the other pre-defined Adjustment Type classifications.

Module Usage

This field is related to FTP Adjustment Rules. When the **Other Adjustment** adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the assumption input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Other Adjustments Amount (OTHER_ADJUSTMENTS_AMT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies an Other Adjustment amount in the TP Adjustment Rule. The **Other Adjustment** type can be used for defining any type of TP Add-on Rate that does not fit into one of the other pre-defined Adjustment Type classifications.

Module Usage

This field is related to FTP Adjustment Rules. When the **Other Adjustment** adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the amount input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Other Adjustments Rate (OTHER_ADJUSTMENTS_RATE)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies an **Other Adjustment** rate or defines a formula based rate in the TP Adjustment Rule. The **Other Adjustment** type can be used for defining any type of TP Add-on Rate that does not fit into one of the other pre-defined Adjustment Type classifications.

Module Usage

This field is related to FTP Adjustment Rules. When the **Other Adjustment** adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the assumption input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Payment Adjustment Date (PMT_ADJUST_DATE)

Definition

Date of next payment adjustment for adjustable rate, negative amortization type accounts.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_ADJUST_DATE when processing and calculating the current payment for negative amortization type accounts. This is relevant only for adjustable rate accounts where AMRT_TYPE_CD = 600.

1. The cash flow engine recalculates negatively amortizing payment amounts on a PMT_ ADJUST_DATE. Just like a payment recalculation for a NEG_AMRT_LIMIT, a payment recalculation on the PMT_ADJUST_DATE takes into account the effects of payment decrease/increase limits per period (PMT_DECR_CYCLE, PMT_INCR_CYCLE) and payment decrease/increase limits for the life of the record (PMT_DECR_LIFE, PMT_INCR_LIFE). This provides for additional negative amortization to occur even after PMT_ADJUST_DATE has recalculated the payment amount.

2. PMT_ADJUST_DATE is incremented forward by the PMT_CHG_FREQ field until maturity.

3. PMT_ADJUST_DATE differs from NEG_AMRT_EQ_DATE because on PMT_ ADJUST_DATE, the calculated payment is constrained by payment decrease/increase limits per period and payment decrease/increase limits for the life of the record. However, on the NEG_AMRT_EQ_DATE, the calculated payment overrides these payment change limits.

4. For an explanation of the relationships among the relevant negative amortization fields, see the Negative Amortization Amount (NEG_AMRT_AMT) section, page 6-74.

Data Verification Requirements and Suggested Defaults

- For fixed rate and non-term accounts, PMT_ADJUST_DATE = 01-JAN-1900.
- For adjustable rate accounts with AMRT_TYPE_CD <> 600, PMT_ADJUST_ DATE = NEXT_REPRICE_DATE.
- For adjustable rate accounts with AMRT_TYPE_CD = 600, the following conditions should exist:
  - PMT_ADJUST_DATE > AS_OF_DATE
  - PMT_ADJUST_DATE > ORIGINATION_DATE
  - PMT_ADJUST_DATE <= AS_OF_DATE + PMT_CHG_FREQ) PMT_ ADJUST_DATE <= MATURITY_DATE

Payment Change Frequency (PMT_CHG_FREQ)

Definition

Used in conjunction with PMT_CHG_FREQ_MULT to define the frequency at which an account's payment adjusts.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_CHG_FREQ when processing and calculating the current payment for negative amortization type accounts. This is relevant only for adjustable-rate accounts where AMRT_TYPE_CD = 600.

1. The PMT_CHG_FREQ is used to increment forward from the PMT_ADJUST_ DATE.
2. The cash flow engine recalculates negatively amortizing payment amounts on a PMT_CHG_ FREQ. Just like a payment recalculation for a NEG_AMRT_LIMIT, a payment recalculation on the PMT_CHG_FREQ takes into account the effects of payment decrease/increase limits per period (PMT_DECR_CYCLE, PMT_INCR_CYCLE) and payment decrease/increase limits for the life of the record (PMT_DECR_ LIFE, PMT_INCR_LIFE). This provides for additional negative amortization to occur even after the PMT_CHG_FREQ event has recalculated the payment amount.

3. For an explanation of the relationships among the relevant negative amortization fields, see the Negative Amortization Amount (NEG_AMRT_AMT), page 6-74 section.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD = 600, PMT_CHG_FREQ must be either 0 or a positive value. PMT_CHG_FREQ cannot be negative.
- For fixed-rate and non-term accounts, PMT_CHG_FREQ = 0.
- For adjustable-rate accounts with AMRT_TYPE_CD <> 600, PMT_CHG_FREQ = REPRICE_FREQ.
- For adjustable-rate accounts with AMRT_TYPE_CD = 600, the following conditions should exist:
  - PMT_CHG_FREQ <> 0
  - PMT_CHG_FREQ <= PMT_ADJUST_DATE - AS_OF_DATE

    Note: Validation of PMT_CHG_FREQ should always be done in conjunction with PMT_CHG_FREQ_MULT.

Payment Change Frequency Multiplier (PMT_CHG_FREQ_MULT)

Definition

Used in conjunction with PMT_CHG_FREQ to define the frequency at which an account's payment adjusts.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_CHG_FREQ_MULT when processing and calculating the current payment for negative amortization-type accounts. PMT_CHG_FREQ_MULT determines the units (Months, Days or Years) of PMT_CHG_FREQ.

Data Verification Requirements and Suggested Defaults

- Valid values are:
  - D - Days
  - M - Months

- Y - Years

- For information on PMT_CHG_FREQ_MULT validation, refer to the validation discussion for Payment Change Frequency (PMT_CHG_ FREQ), page 6-96

Payment Decrease Limit - Cycle (PMT_DECR_CYCLE)

Definition

Maximum payment decrease allowed during a payment change cycle of an adjust-able-rate instrument.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_DECR_CYCLE when re-calculating the current payment for negative amortization events such as NEG_AMRT_LIMIT, PMT_ADJUST_DATE, and PMT_ CHG_FREQ. This is relevant only for adjustable rate accounts where AMRT_TYPE_ CD = 600.

1. For negative amortization-type accounts, the cash flow engine uses PMT_ DECR_CYCLE to calculate the maximum decrease in the payment amount allowed from the previous payment change to the next.

2. PMT_DECR_CYCLE is defined in terms of a percentage. The cash flow engine performs the following check:

   Previous Current Payment - Newly calculated payment > (PMT_DECR_ CYCLE/100 * Previous Current Payment)

   If the newly calculated payment satisfies the earlier equation, the cash flow engine limits the decrease to the amount = (PMT_DECR_CYCLE * Previous Current Payment).

   For example, if PMT_DECR_CYCLE = 5.00, the calculated current payment is not allowed to decrease by more than 5% of the previous current payment.

3. If PMT_DECR_CYCLE = 0, the cash flow engine assumes that there is no payment decrease limit per payment change period.

4. The PMT_DECR_CYCLE is referenced when the following negative amortization events occur: NEG_AMRT_LIMIT, PMT_ADJUST_DATE, and PMT_CHG_FREQ.

5. For an explanation of PMT_DECR_CYCLE's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

Data Verification Requirements and Suggested Defaults

- For accounts with AMRT_TYPE_CD <> 600, PMT_DECR_CYCLE = 0.
- For accounts with AMRT_TYPE_CD = 600, 0 <= PMT_DECR_CYCLE < 100

Payment Decrease Limit - Life (PMT_DECR_LIFE)

Definition

Maximum payment decrease allowed during the life of an adjustable-rate instrument.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_DECR_LIFE when re-calculating the current payment for negative amortization events such as NEG_AMRT_LIMIT, PMT_ADJUST_DATE, and PMT_CHG_ FREQ. This is relevant only for adjustable-rate accounts where AMRT_TYPE_CD = 600.

1. For negative amortization-type accounts, the cash flow engine uses PMT_DECR_LIFE to calculate the maximum decrease in the payment allowed during the life of the account. PMT_DECR_LIFE is defined in terms of a percentage of ORG_ PAYMENT. The cash flow engine performs the following check:

   ORG_PAYMENT - Newly calculated payment > (PMT_DECR_LIFE/100 * ORG_PAYMENT)

   If the newly calculated payment satisfies the earlier equation, the cash flow engine limits the decrease to the amount = (PMT_DECR_LIFE * ORG_ PAYMENT).

   For example, if PMT_DECR_LIFE = 25.00, the calculated current payment is not allowed to decrease by more than 25% of ORG_PAYMENT.

2. If PMT_DECR_LIFE = 0, the cash flow engine assumes that there is no lifetime payment decrease limit.

3. The PMT_DECR_LIFE field is referenced when the following negative amortization events occur:

   - NEG_AMRT_LIMIT
   - PMT_ADJUST_DATE
   - PMT_CHG_FREQ

4. For an explanation of PMT_DECR_LIFE's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

Data Verification Requirements and Suggested Defaults

1. For accounts with AMRT_TYPE_CD <> 600, PMT_DECR_LIFE = 0.

2. For accounts with AMRT_TYPE_CD = 600, 0 <= PMT_DECR_LIFE < 100.

Payment Frequency (PMT_FREQ)

Definition

Used in conjunction with PMT_FREQ_MULT to define the payment frequency of an account.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_FREQ for calculating and processing payments, prepayments, and market valuation. The cash flow engine forecasts future next payment dates by incrementing NEXT_PAYMENT_DATE forward by PMT_FREQ.

Oracle ALM

PMT_FREQ is used in the Market Value calculation. The calculation is as follows:

$MV = \text{Cash Flow}/(1 + r)^{t}$

PMT_FREQ is used to derive the **r** and **t** value for records when PMT_FREQ_MULT = M.

Following is an explanation of how the calculations that use the PMT_FREQ.

For records with PMT_FREQ MULT = M, **r** is defined as the discount rate divided by the number of payments per period. The number of payments per period is calculated as (12/PMT_FREQ).

r = Discount Rate/12/PMT_FREQ

The **t** value is defined as the number of days in the payment period from the AS_OF_DATE divided by 30.41667. This amount is then rounded to a whole integer and then divided by the PMT_FREQ. The equation is as follows:

t = (Round((Current Payment Date - AS_OF_DATE) / 30.41667))/PMT_FREQ.

For user-defined Payment Schedules and Patterns, the Market Value formula is the same as used for PMT_FREQ = D records. That formula is:

t = (Current Payment Date - AS_OF_DATE/PMT_FREQ).

> **Note:** For User-Defined Payment Schedules and Patterns, the Market Value formula is the same as used for PMT_FREQ = D records. That formula is t = (Current Payment Date - AS_OF_ DATE/PMT_FREQ).

Funds Transfer Pricing

PMT_FREQ is referenced by the cash flow transfer pricing methodologies when deriving discounted cash flows. For calculation details, see Cash Flow Calculations, page 5-1.

Oracle ALM and Funds Transfer Pricing

Both Oracle ALM and Funds Transfer Pricing use PMT_FREQ in defining payment dates.

1. Oracle ALM defines the date of payment using PMT_FREQ in the following manner:

   During cash flow generation, from the NEXT_PAYMENT_DATE, cash flow engine increments forward by the PMT_FREQ until the MATURITY_DATE. Oracle ALM does not use NEXT_PAYMENT_DATE for payment schedule records. For a full description of processing, see the Next Payment Date (NEXT_PAYMENT_DATE), page 6-81 section.

   > **Note:** If the record uses user-defined Payment Schedules or Patterns, PMT_FREQ is not referenced for payment date information. Instead, the Payment Schedules and Pattern information are used to define all future payments.

2. Oracle Funds Transfer Pricing defines the date of payment using PMT_FREQ in the following ways:

   Adjustable-rate Cash Flow Transfer-priced Records: In defining the transfer rate for an adjustable-rate record, cash flow engine uses the PMT_FREQ to define the payment dates from the LAST_REPRICE_DATE to the NEXT_REPRICE_DATE. In order to define all the payment events within this period, FTP rolls back from the NEXT_PAYMENT_DATE by the PMT_FREQ until just before the LAST_REPRICE_DATE. From this calculated payment date, the cash flow engine again rolls forward by the PMT_FREQ, but this time cash flows are produced. The cash flows produced are used by one of the four cash flow transfer pricing methodologies in order to derive the transfer rate.

   Fixed-rate Cash Flow Transfer-priced Records: In defining the payment dates for cash flow transfer-priced fixed-rate records, the cash flow engine starts from the record's ORIGINATION_DATE and rolls forward by the PMT_FREQ until the MATURITY_DATE.

   > **Note:** If the transfer pricing record uses User-Defined Payment Schedules or Patterns, PMT_FREQ is not referenced for payment date information.

3. Oracle ALM and Funds Transfer Pricing both use the PMT_FREQ in the following, similar manner:

- On date of payment, the cash flow engine calculates the interest payments, principal payments, current deferred payments, prepayments, unscheduled prepayments, and negative amortization, if applicable. For an adjustable-type record where REPRICE_FREQ < PMT_FREQ, Oracle ALM applies only the last repriced rate for the purposes of payment calculation.

- During the payment calculation processing, PMT_FREQ is used for interest income calculations (financial element 430) for records with ACCRUAL_ BASIS_CDs of 30/360, 30/365, 30/ACTUAL. See Accrual Basis Code (ACCRUAL_BASIS_CD), page 6-9 and Interest Type Code (INT_TYPE), page 6-59 for formulas.

- PMT_FREQ is used in the **Remaining Number of Payments** calculation when calculating the payment amounts for balloon records and specific user-defined payment pattern instances. For the formula and further details, see the Amortization Term (AMRT_TERM), page 6-20 section.

- As each payment is made on the PMT_FREQ, the cash flow engine reduces the REMAIN_ NO_PMTS_C by **1**. If the newly calculated REMAIN_NO_PMTS_C = 1, the next payment date is set to MATURITY_DATE.

  Note: The use of REMAIN_NO_PMTS_C varies depending on record type. For more details and special considerations for payment-patterned records, see the Remaining Number of Payments (REMAIN_ NO_PMTS_C), page 6-119 and Amortization Term (AMRT_TERM), page 6-20 sections

Data Verification Requirements and Suggested Defaults

- PMT_FREQ > 0 in all cases. PMT_FREQ = 0 can cause incorrect calculations to occur.

- PMT_FREQ <= ORG_TERM

- MATURITY_DATE <= NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_C * PMT_FREQ).

  PMT_FREQ = **1** and PMT_FREQ MULT = **D** require the cash flow engine to perform cash flow calculations for every day of the modeling horizon. This slows processing significantly.

  Note: Validation of PMT_FREQ should always be done in conjunction with PMT_FREQ_MULT.

Payment Frequency Multiplier (PMT_FREQ_MULT)

Definition

Used in conjunction with PMT_FREQ to define the payment frequency of an account.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_FREQ_MULT for calculating and processing payments. PMT_FREQ_ MULT determines the units (Months, Days or Years) of PMT_FREQ.

Data Verification Requirements and Suggested Defaults

- Valid values are:
  - D - Days
  - M - Months
  - Y - Years
- For information on PMT_FREQ_MULT validation, refer to the validation discussion for Payment Frequency (PMT_FREQ), page 6-100 .

Payment Increase Limit - Cycle (PMT_INCR_CYCLE)

Definition

Maximum payment increase allowed during a payment change cycle of an adjustable-rate instrument.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_INCR_CYCLE when re-calculating the current payment for negative amortization events such as NEG_AMRT_LIMIT, PMT_ADJUST_DATE, and PMT_ CHG_FREQ. This is relevant only for adjustable-rate accounts where AMRT_TYPE_ CD = 600.

1. For negative amortization-type accounts, the cash flow engine uses PMT_ INCR_CYCLE to calculate the maximum increase in the payment amount allowed from the previous payment change to the next.

2. PMT_INCR_CYCLE is defined in terms of a percentage. The cash flow engine performs the following check:

   Newly calculated payment - Previous Current Payment > (PMT_INCR_CYCLE/100

* Previous Current Payment)

If the newly calculated payment satisfies the earlier equation, the cash flow engine limits the increase to the amount = (PMT_INCR_CYCLE * Previous Current Payment).

For example, if PMT_INCR_CYCLE = 5.00, the calculated current payment is not allowed to increase by more than 5% of the previous current payment.

3. If PMT_INCR_CYCLE = 0, the cash flow engine assumes that there is no payment increase limit per payment change period.

4. The PMT_INCR_CYCLE is referenced when the following negative amortization events occur:

   - NEG_AMRT_LIMIT

   - PMT_ADJUST_DATE

   - PMT_CHG_FREQ

5. For an explanation of PMT_INCR_CYCLE's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD <> 600, PMT_INCR_CYCLE = 0

- If AMRT_TYPE_CD = 600, 0 <= PMT_INCR_CYCLE < 100

Payment Increase Limit - Life (PMT_INCR_LIFE)

Definition

Maximum payment increase allowed during the life of an adjustable-rate instrument.

Module Usage

Oracle ALM and Funds Transfer Pricing cash flow methodologies reference PMT_INCR_LIFE when re-calculating the current payment for negative amortization events such as NEG_AMRT_LIMIT, PMT_ADJUST_DATE, and PMT_CHG_FREQ. This is relevant only for adjustable-rate accounts where AMRT_ TYPE_CD = 600.

1. For negative amortization-type accounts, the cash flow engine uses PMT_ INCR_LIFE to calculate the maximum increase in the payment allowed during the life of the account.

2. PMT_INCR_LIFE is defined in terms of a percentage of ORG_PAYMENT. The cash flow engine performs the following check:

   Newly calculated payment - ORG_PAYMENT > (PMT_INCR_LIFE/100 * ORG_PAYMENT)

   If the newly calculated payment satisfies the earlier equation, the cash flow engine limits the increase to the amount = (PMT_INCR_LIFE * ORG_ PAYMENT).

   For example, if PMT_INCR_LIFE = 25.00, the calculated current payment is not allowed to increase by more than 25% of ORG_PAYMENT.

3. The PMT_INCR_LIFE is referenced when the following negative amortization events occur:

   - NEG_AMRT_LIMIT
   - PMT_ADJUST_DATE
   - PMT_CHG_FREQ

4. For an explanation of PMT_INCR_LIFE's relationship with other related negative amortization fields, see Negative Amortization Amount (NEG_ AMRT_AMT), page 6-74.

Data Verification Requirements and Suggested Defaults

1. If PMT_INCR_LIFE = 0, the cash flow engine assumes that there is no lifetime payment increase limit.

2. If AMRT_TYPE_CD <> 600, PMT_INCR_LIFE = 0.

3. If AMRT_TYPE_CD = 600, 0 <= PMT_INCR_LIFE < 100.

Percent Sold (PERCENT_SOLD)

Definition

Percent of balance sold to investors.

Module Usage

PERCENT_SOLD is used to calculate net balance and net payment when the instrument is partially participated out to another financial institution. In the case of a participated loan, the bank partially owns the loan. A bank would participate out the balance of the loan to another financial institution if it was not able to lend the entire amount or if lending the entire amount would exceed the bank's legal lending limits. By selling most of the financing to another financial institution, the bank would earn fee income from servicing the loan and would be able to retain other banking relationships, such as checking accounts.

1. Oracle ALM and Funds Transfer Pricing cash flow methodologies use PERCENT_SOLD to perform all net balance calculations.

2. The cash flow engine performs all of the cash flow calculations on a gross balance basis, but must net out the principal and interest portions not owned/owed by the bank. PERCENT_SOLD defines the percent of the balances participated (sold) by the bank.

3. The calculation performed to net out participations sold is:

   [Net Balance] = [Gross Balance] * (100 - PERCENT_SOLD)/100

4. PERCENT_SOLD does not apply if an account has been sold to another subsidiary of the same company. In the case of accounts that have had portions sold from one legal entity of a holding company to another, PERCENT_SOLD = 0.

5. For wholly (100%) owned accounts, PERCENT_SOLD = 0.

6. Oracle ALM only writes out Result Detail balance results net of participations.

Data Verification Requirements and Suggested Defaults

For accounts with PERCENT_SOLD <> 0, the following conditions should exist:

- 0 < PERCENT_SOLD <= 100.
- CUR_NET_BOOK_BAL_C = CUR_BOOK_BAL * (100 - PERCENT_SOLD) / 100.
- CUR_NET_PAR_BAL_C = CUR_PAR_BAL * (100 - PERCENT_SOLD) / 100.
- ORG_NET_BOOK_BAL_C = ORG_BOOK_BAL * (100 - PERCENT_SOLD) / 100.
- ORG_NET_PAR_BAL_C = ORG_PAR_BAL * (100 - PERCENT_SOLD) / 100.
- If not applicable, PERCENT_SOLD = 0.

Pricing Incentive Charge Credit (PRICING_INC_CHARGE_CREDIT)

Definition

Oracle Funds Transfer Pricing calculates the funds charge or credit due to Pricing Incentives when the user selects the Adjustments, Charge/Credit option in the TP Process – Calculation Elements block.

Module Usage

This field is populated by the FTP process (optionally) if selected in the Calculation Elements block. The TP Engine calculates this amount as follows:

AVG_BOOK_BAL or CUR_BOOK_BAL X PRICING_INCENTIVE_RATE/100 X ACCRUAL BASIS

> **Note:** The balance referenced in the earlier calculation is selected in the FTP Application Preferences screen.

Data Verification Requirements and Suggested Defaults

None

Pricing Incentive Amount (PRICING_INCENTIVE__AMT)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Pricing Incentive Adjustment amount in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the Pricing Incentive Adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the amount input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Pricing Incentive Rate (PRICING_INCENTIVE_RATE)

Definition

Oracle Funds Transfer Pricing populates this field (optionally), when the user specifies a Pricing Incentive rate or defines a formula based rate in the TP Adjustment Rule.

Module Usage

This field is related to FTP Adjustment Rules. When the Pricing Incentive adjustment type is defined and included in the TP process, the TP Engine calculates and populates this field according to the assumption input by the user through the Adjustment Rule user interface.

Data Verification Requirements and Suggested Defaults

None

Prior Transfer Pricing Period Average Daily Balance (PRIOR_TP_PER_ADB)

Definition

The average balance at the reprice date(s) prior to the LAST_REPRICE_DATE.

Module Usage

Oracle ALM

Oracle ALM does not reference PRIOR_TP_PER_ADB or CUR_TP_PER_ ADB.

Oracle Funds Transfer Pricing

When processing with the mid-period repricing option, Oracle Funds Transfer Pricing references PRIOR_TP_PER_ADB as the average daily balance at the time of the last repricing event prior to the LAST_REPRICE_DATE. This field is used in conjunction with the CUR_TP_PER_ADB field.

1. Mid-period repricing produces an average transfer rate over the current processing month if the LAST_REPRICE_DATE occurred since the beginning of the processing month. PRIOR_TP_PER_ADB and CUR_TP_PER_ADB are used as average balance weightings in the mid-period pricing equation. PRIOR_TP_ PER_ADB is used to determine the balance on the reprice date prior to the LAST_REPRICE_DATE and CUR_TP_PER_ADB is used to determine the balance as of the LAST_REPRICE_DATE.

   For an example and explanation of the relationship between PRIOR_TP_ PER_ADB and CUR_TP_PER_ADB, see Current Transfer Pricing Period Average Daily Balance (CUR_TP_PER_ ADB), page 6-43.

   For additional information. See *Oracle Financial Services Funds Transfer Pricing User Guide*.

2. If the CUR_TP_PER_ADB and PRIOR_TP_PER_ADB are not available, use CUR_PAR_BAL as the default.

3. If the TEASER_END_DATE is greater than the AS_OF_DATE, the mid-period repricing does not apply and the CUR_TP_PER_ADB and PRIOR_TP_PER_ ADB fields are not used.

Data Verification Requirements and Suggested Defaults

- If the record is adjustable and repricing occurs within the month, PRIOR_TP_ PER_ADB = (average) balance at the time of the reprice date prior to the LAST_ REPRICE_DATE.

- If the average balance at the time of the reprice date prior to the LAST_REPRICE_DATE is not available, use CUR_PAR_BAL.

Rate Cap Life (RATE_CAP_LIFE)

Definition

Maximum interest rate allowed during life of an adjustable-rate instrument.

Module Usage

Oracle ALM

Oracle ALM references RATE_CAP_LIFE when calculating a forecasted interest rate for adjustable rate records.

Existing business adjustable-rate records: The cash flow engine matches the REPRICE_FREQ, INTEREST_RATE_CD, and the reprice date to the information contained in the Forecast Rates assumption rule to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding, rate caps (RATE_CAP_LIFE) or floors and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

> **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

For accounts beyond the tease period: After the rounding and rate change minimum are applied to the forecast rate plus margin, if the resulting rate is > RATE_CAP_LIFE, the cash flow engine sets the record's forecasted rate = RATE_CAP_LIFE.

> **Note:** For details of the repricing process see the Next Repricing Date (NEXT_REPRICE_DATE), page 6-83 section and Cash Flow Calculations , page 5-1.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_CAP_LIFE because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- If the record is fixed-rate, RATE_CAP_LIFE = 0.
- If the record is adjustable without a rate cap, RATE_CAP_LIFE = 0.
- If RATE_CAP_LIFE <> 0, CUR_GROSS_RATE and CUR_NET_RATE <= RATE_CAP_LIFE.

- If RATE_CAP_LIFE <> 0, RATE_CAP_LIFE = ORG_RATE + RATE_INCR_LIFE.

Rate Change Minimum (RATE_CHG_MIN)

Definition

Minimum change in the repricing index that is necessary for a change to be made to the interest rate.

Module Usage

Oracle ALM

Oracle ALM references RATE_CHG_MIN when calculating a forecasted interest rate.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD, and the reprice date to the information contained in the Forecast Rates assumption rule to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding is applied, followed by a check of the RATE_ CHG_MIN, when determining the fully indexed rate.

   > **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. If the absolute value of [(forecasted rate + margin) - (previous rate + margin)] < RATE_CHG_MIN, the cash flow engine sets the new forecasted rate = previous rate. The cash flow engine does not change the previous rate to the new forecasted rate. The previous rate is defined as either the rate on the detail record (CUR_ NET_RATE or CUR_GROSS_RATE) or the previous forecasted rate from the Forecast Rates assumption rule. After the RATE_CHG_MIN is calculated any other cap/floor, rounding, and tease periods are then applied.

   For details of the repricing process see the Next Repricing Date (NEXT_ REPRICE_DATE), page 6-83 section and Cash Flow Calculations, page 5-1.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_CHG_MIN because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- If the record is fixed-rate, RATE_CHG_MIN = 0.

- If the record is adjustable, without a minimum rate change amount, RATE_ CHG_MIN = 0.

- If RATE_CHG_MIN <> 0, 0 < RATE_CHG_MIN <= 1.

Rate Change Rounding Code (RATE_CHG_RND_CD)

Definition

Method used for rounding of the interest rate change.

Module Usage

Oracle ALM

Oracle ALM uses RATE_CHG_RND_CD to determine the rounding method that is applied to the current rate after a repricing event. RATE_CHG_ RND_CD is used in conjunction with RATE_CHG_RND_FAC.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding (RATE_CHG_RND_CD and RATE_ CHG_RND_FAC), rate caps/floors, and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

   Note: ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. The RATE_CHG_RND_CD accepts values 1 - 4. Depending on the value input, the value of the forecasted rate as calculated earlier will differ.

   Following are explanations of the possible values:

Rate Change Rounding Code	Description
RATE_CHG_RND_CD = 1	**Truncate** The cash flow engine truncates the forecasted rate to a whole value. For example, if unrounded forecasted = 8.65, the truncated forecasted rate =8.00.
RATE_CHG_RND_CD = 2	**Round Up**

Rate Change Rounding Code	Description
	The cash flow engine rounds the rate up to the nearest multiple of the RATE_CHG_RND_FAC. For example, if the unrounded forecasted rate is 8.65 and the RATE_CHG_RND_FAC = 0.5, the rounded forecasted rate = 9.00.
RATE_CHG_RND_CD = 3	**Round Down**
	The cash flow engine rounds the rate down to the nearest multiple of RATE_CHG_RND_FAC. For example, if the unrounded forecasted rate is 8.65 and the RATE_CHG_RND_FAC = 0.25, the rounded forecasted rate = 8.50.
RATE_CHG_RND_CD = 4	**Round Nearest**
	The cash flow engine rounds the rate to the nearest multiple of RATE_CHG_RND_FAC. For example, if the unrounded forecasted rate is 8.65 and the RATE_CHG_RND_FAC = 0.25, the rounded forecasted rate = 8.75.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_CHG_RND_CD because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- RATE_CHG_RND_CD must be within the range of **0** to **4**.
- For fixed-rate accounts, RATE_CHG_RND_CD = 4 and RATE_CHG_RND_FAC = 0.
- For variable-rate accounts with rates that are not rounded, RATE_CHG_RND_ CD = 4 and RATE_CHG_RND_FAC = 0.

Rate Change Rounding Factor (RATE_CHG_RND_FAC)

Definition

Factor to which the rate change on an adjustable instrument is rounded.

Module Usage

Oracle ALM

Oracle ALM references RATE_CHG_RND_FAC when calculating a forecasted interest rate, and contains the value to which forecasted interest rates are rounded. RATE_CHG_RND_FAC is used in conjunction with RATE_CHG_RND_CD.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding (RATE_CHG_RND_CD and RATE_ CHG_RND_FAC), rate caps/floors, and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

   Note: ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. For an explanation of usage and the relationship between RATE_CHG_RND_ FAC and RATE_CHG_RND_CD, see the Rate Change Rounding Code (RATE_ CHG_RND_CD), page 6-111 section.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_CHG_RND_FAC because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- For fixed-rate accounts, RATE_CHG_RND_CD = 4 and RATE_CHG_RND_FAC = 0.

- For variable-rate accounts with rates that are not rounded, RATE_CHG_RND_ CD = 4 and RATE_CHG_RND_FAC = 0.

- For variable-rate accounts with RATE_CHG_RND_FAC <> 0, 0 < RATE_CHG_ RND_FAC <= 1.

Rate Decrease Limit - Cycle (RATE_DECR_CYCLE)

Definition

Maximum rate decrease allowed during a repricing cycle for an adjustable-rate instrument.

Module Usage

Oracle ALM

Oracle ALM references RATE_DECR_CYCLE when calculating a forecasted interest rate. RATE_DECR_CYCLE sets the maximum amount (in terms of basis points) that the interest rate may decrease in a given REPRICE_FREQ.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding, rate caps/floors (RATE_DECR_ CYCLE), and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

   Note: ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. When applying the RATE_DECR_CYCLE, the cash flow engine checks for the following:

   - Previous Current Rate > Calculated forecasted rate.
   - Previous Current Rate - Calculated forecasted rate > RATE_DECR_CYCLE.

   If both equations are true, the rate change during the repricing period has exceeded RATE_DECR_CYCLE. In this case, the new forecasted rate is limited to the previous current rate - RATE_DECR_CYCLE.

   This is illustrated in the following example:

```
RATE_DECR_CYCLE       = 2.00 (200 basis points)
Previous Current Rate   = 10.00
Calculated (raw) rate   = 7.75
              10.00 > 7.75
             10.00 - 7.75 (= 2.25) > 2.00
New current rate        = 10.00 - 2.00 = 8.00
```

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_DECR_CYCLE because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- For fixed-rate accounts, RATE_DECR_CYCLE = 0.
- For variable-rate accounts without a maximum rate decrease, RATE_DECR_CYCLE = 0.
- For RATE_DECR_CYCLE <> 0, 0 <= RATE_DECR_CYCLE <= RATE DECR LIFE.

Rate Floor Life (RATE_FLOOR_LIFE)

Definition

Minimum interest rate allowed during life of an adjustable-rate instrument.

Module Usage

Oracle ALM

Oracle ALM references RATE_FLOOR_LIFE when calculating a forecasted interest rate for adjustable-rate records.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding, rate caps/floors (RATE_FLOOR_ LIFE), and tease periods are applied, and the resulting rate is applied to the record as the repriced rate.

   > **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. If the forecast rate plus margin < RATE_FLOOR_LIFE, the cash flow engine sets the record's forecasted rate = RATE_FLOOR_LIFE. Rounding and tease periods are then applied.

   For details of the repricing process see the Next Repricing Date (NEXT_ REPRICE_DATE), page 6-83 section.

3. For any forecasted rate changes throughout the life of the instrument, the cash flow engine references RATE_FLOOR_LIFE.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_FLOOR_LIFE because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- If the record is fixed-rate, RATE_FLOOR_LIFE = 0.
- If the record is adjustable without a rate floor, RATE_FLOOR_LIFE = 0.
- If RATE_FLOOR_LIFE <> 0, CUR_GROSS_RATE and CUR_NET_RATE >= RATE_FLOOR_LIFE .
- If RATE_FLOOR_LIFE <> 0, RATE_FLOOR_LIFE = ORG_RATE - RATE_ DECR_LIFE.

Rate Increase Limit - Cycle (RATE_INCR_CYCLE)

Definition

Maximum rate increase allowed during a repricing cycle for an adjustable-rate instrument.

Module Usage

Oracle ALM

Oracle ALM references RATE_INCR_CYCLE when calculating a forecasted interest rate. RATE_INCR_CYCLE sets the maximum amount (in terms of basis points) that the interest rate may increase in a given REPRICE_FREQ.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the reprice date to the information contained in the Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding, rate caps (RATE_INCR_CYCLE) or floors, and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

   > **Note:** ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. When applying the RATE_INCR_CYCLE, the cash flow engine checks for the following:

   - Calculated forecasted rate > Previous Current Rate.
   - Calculated forecasted rate - Previous Current Rate > RATE_INCR_CYCLE.

   If both equations are true, the rate change during the repricing period has exceeded RATE_INCR_CYCLE. In this case, the new forecasted rate is limited to the previous current rate + RATE_INCR_CYCLE.

This is illustrated in the following example:

```
RATE_INCR_CYCLE        = 2.00 (200 basis points)
    Previous Current Rate = 10.00
    Calculated rate       = 12.25
             □ 12.25 > 10.00
             □ 12.25 -10.00 (= 2.25) > 2.00
  New current rate =     10.00 + 2.00 = 12.00
```

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_INCR_CYCLE because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- For fixed-rate accounts, RATE_INCR_CYCLE = 0.
- For variable-rate accounts without a maximum rate increase, RATE_INCR_ CYCLE = 0.
- For RATE_INCR_CYCLE <> 0, 0 <= RATE_INCR_CYCLE <= RATE INCR LIFE.

Rate Set Lag (RATE_SET_LAG)

Definition

Used in conjunction with RATE_SET_LAG_MULT to define the period by which repricing lags the current interest rate changes.

Module Usage

Oracle ALM

Oracle ALM references RATE_SET_LAG when calculating a forecasted interest rate.

1. For existing business adjustable-rate records, the cash flow engine matches the REPRICE_ FREQ, INTEREST_RATE_CD and the lagged reprice date (after referencing RATE_SET_LAG) to the information contained in the Oracle ALM Forecast Rates assumption rule. This is to assign a forecasted rate. The margin is then added to this forecasted rate. Any rounding, rate caps/floors, and tease periods are applied, and the resulting rate is applied to the record as the record's repriced rate.

   Note: ADJUSTABLE_TYPE_CD = 30 or 50 will not reference reprice dates.

2. If RATE_SET_LAG > 0, the cash flow engine does not assign a forecasted interest rate based upon NEXT_REPRICE_DATE. Instead, the cash flow engine assigns the account an interest rate based upon the date NEXT_ REPRICE_DATE -

RATE_SET_LAG.

This is illustrated in the following example:

REPRICE_FREQ = 3

REPRICE_FREQ_MULT = M

NEXT_REPRICE_DATE = 01-APR

RATE_SET_LAG = 1

RATE_SET_LAG_ MULT = M

MARGIN = 1.00

In this example the account is tied to the Treasury Yield Curve. Because of the RATE_SET_LAG, Oracle ALM references the Treasury Yield Curve one month before the NEXT_REPRICE_DATE. The 3-Month point on the Treasury Yield Curve on 01-MAR equals 5%. Therefore the repriced rate equals 6% (5% plus the 1% margin).

3. If the RATE_SET_LAG > (Cash Flow Date - AS_OF_DATE), the cash flow engine uses the base rate from the Forecast Rates assumption rule. The cash flow engine does not reference the rates historically. For instance, if a 3-month RATE_SET_LAG is applied to a repricing event that is two months from the AS_OF_DATE, the cash flow engine does not reference the Historical Rates database, to obtain the yield curve information that is one month before the AS_ OF_DATE. Instead, the cash flow engine applies the current position rate from the Oracle ALM Forecast Rates assumption rule.

Oracle Funds Transfer Pricing

Oracle Funds Transfer Pricing does not reference RATE_SET_LAG because it processes cash flows within repricing periods.

Data Verification Requirements and Suggested Defaults

- RATE_SET_LAG must be either 0 or a positive value.
- If the record is a fixed-rate account, RATE_SET_LAG = 0.
- If the record is an adjustable-rate account, RATE_SET_LAG >= 0

> **Note:** Validation of RATE_SET_LAG should always be done in conjunction with RATE_SET_LAG_MULT.

Rate Set Lag Multiplier (RATE_SET_LAG_MULT)

Definition

Used in conjunction with RATE_SET_LAG to define the rate set lag period.

Module Usage

The Oracle ALM cash flow engine references RATE_SET_LAG_MULT when calculating a forecasted interest rate. RATE_SET_LAG_MULT determines the units (Months, Days or Years) of RATE_SET_LAG.

Data Verification Requirements and Suggested Defaults

Valid values are:

- D - Days
- M - Months
- Y - Years

For information on RATE_SET_LAG_MULT validation, refer to the validation discussion for Rate Set Lag (RATE_SET_LAG), page 6-117.

Remaining Number of Payments (REMAIN_NO_PMTS_C)

Definition

The remaining number of principal, interest, or principal and interest payments to be made from the AS_OF_DATE until the MATURITY_DATE of the record.

Module Usage

The Oracle ALM and Funds Transfer Pricing cash flow methodologies reference REMAIN_NO_PMTS_C when calculating and processing payments. The cash flow engine uses REMAIN_NO_PMTS_C to determine the number of payments that remain to be paid until the account matures.

Oracle ALM and Funds Transfer Pricing

The number of remaining payments is used by the cash flow engine for two purposes:

Definition of payment dates on which principal and interest are paid: As each payment is made during the life of the instrument, the cash flow engine reduces the REMAIN_ NO_PMTS_C by **1** and rolls the payment period forward by the PMT_FREQ. If the newly calculated REMAIN_NO_PMTS_C = 1, the cash flow engine no longer rolls the PMT_FREQ and makes the next (and final) payment on the MATURITY_DATE. For additional details, including special considerations for payment-patterned records, see

Amortization Term (AMRT_TERM), page 6-20.

Amortization re-calculation purposes : When a re-calculation occurs (repricing or negative amortization event), the cash flow engine references the REMAIN_NO_PMTS_C for recalculation of amortization. For details on calculation, see Current Payment (CUR_PAYMENT), page 6-37.

You should be aware that there are some exceptions to the use of REMAIN_NO_PMTS_C for balloon records and specific User-Defined Payment Patterns and schedules. Depending on the record characteristics, the cash flow engine may calculate the remaining number of payments itself. For further explanations, see the Amortization Term (AMRT_TERM), page 6-20 section.

Oracle Funds Transfer Pricing

1. The Transfer Pricing - Remaining Term Pricing Basis cash flow methodology for fixed-rate records uses REMAIN_NO_PMTS_C as described earlier.

2. The Transfer Pricing - Standard Pricing Basis cash flow methodology for an adjustable-rate records calculates the remaining number of payments as follows:

   REMAIN_NO_PMTS_C + number of payment periods between the NEXT_PAYMENT_DATE and the LAST_REPRICING_DATE.

Data Verification Requirements and Suggested Defaults

- REMAIN_NO_PMTS_C >= 1.
- If REMAIN_NO_PMTS_C = 0, record is invalid.
- If REMAIN_NO_PMTS_C = 1, NEXT_PAYMENT_DATE = MATURITY_DATE.
- If REMAIN_NO_PMTS_C > 1, NEXT_PAYMENT_DATE < MATURITY_DATE.
- Generally:
  - REMAIN_NO_PMTS_C * PMT_FREQ <= ORG_TERM * ORG_TERM_MULT.
  - MATURITY_DATE <= NEXT_PAYMENT_DATE + (REMAIN_NO_PMTS_ C * PMT_FREQ).
- For Behavior Pattern Instruments, REMAIN_NO_PMT_C should equal the number of payments (tenors) specified in the Behavior Pattern.
- For non-term accounts (Other Assets, Other Liabilities, and so on), REMAIN_NO_PMTS_C = 1.
- The maximum number of payment and repricing events that can be modeled cannot exceed 16000. REMAIN_NO_PMTS_C + (REMAIN_TERM_C (in days)) / (REPRICE_FREQ (in days)) > 16000.

Repricing Frequency (REPRICE_FREQ)

Definition

Used in conjunction with REPRICE_FREQ_MULT to define the frequency of rate change of an account.

Module Usage

Oracle ALM and Funds Transfer Pricing reference REPRICE_FREQ when identifying adjustable-rate instruments and calculating a transfer pricing term or forecasted interest rate.

Oracle ALM

1. In identifying an adjustable record, Oracle ALM uses ADJUSTABLE_ TYPE_CD and REPRICE_FREQ. If the ADJUSTABLE_TYPE_CD > 0 and the REPRICE_FREQ > 0, the record is adjustable.

   Note: Even if REPRICE_FREQ > 0, if ADJUSTABLE_TYPE_CD = 0, the record is treated as a fixed-rate.

2. The cash flow engine uses the REPRICE_FREQ to identify repricing events beyond the NEXT_ REPRICE_DATE. The cash flow engine rolls forward from NEXT_REPRICE_DATE by the REPRICE_FREQ to define the record's remaining forecasted reprice dates. Rolling by the REPRICE_FREQ continues until MATURITY_DATE.

   For additional information on the repricing process, see Next Repricing Date (NEXT_REPRICE_DATE), page 6-83.

3. In defining the CUR_NET_RATE, CUR_GROSS_RATE, and TRANSFER_RATE on each reprice date, the cash flow engine matches the record's INTEREST_RATE_CD (TP IRC from the selected TP Rule in the case of transfer rates), the reprice date, and the REPRICE_FREQ to the appropriate term point on the forecasted Interest Rate Code (IRC) in the Oracle ALM Forecast Rates assumption rule. To this derived rate, the cash flow engine adds the relevant margin amount (MARGIN, MARGIN_GROSS). For Transfer Rate, if the Transfer Pricing option is selected in the ALM Process, the cash flow engine will refer to the TP Rule selected in the process and will apply the appropriate TP Method and related TP IRC to determine the new TP rate on the reprice date.

   Any rate caps/floors, tease periods, and rounding is then applied.

   Note: ADJUSTABLE_TYPE_CD = 30 and 50 will not reference reprice date.

Oracle Funds Transfer Pricing

1. REPRICE_FREQ is used to identify whether records are adjustable or not. If REPRICE_FREQ > 0, the record is considered adjustable. If REPRICE_FREQ = 0, the record is fixed-rate.

2. For adjustable-rate straight term methodology, Oracle Funds Transfer Pricing matches the REPRICE_FREQ to the same term on the Transfer Pricing yield curve (Interest Rate Code) defined in Rate Management. For additional information, see Last Repricing Date (LAST_REPRICE_DATE), page 6-64.

3. FTP Adjustment Rules refer to REPRICE_FREQ when the Reference Term is set to " Reprice Frequency". If the instrument is fixed rate and, therefore, does not have a reprice frequency, the calculation retrieves the rate associated with the term point equaling the original term on the instrument.

Data Verification Requirements and Suggested Defaults

- If REPRICE_FREQ > 0, MATURITY_DATE >= NEXT_REPRICE_DATE > AS_OF_DATE.

- If REPRICE_FREQ > 0, INTEREST_RATE_CD is in the range of 1 - 99999.

- If REPRICE_FREQ > 0, ADJUSTABLE_TYPE_CD > 0.

- REPRICE_FREQ must be either **0** or positive.

> **Note:** In Oracle ALM, ADJUSTABLE_TYPE_CD is used in conjunction with REPRICE_FREQ to define adjustable-rate records.

Repricing Frequency Multiplier (REPRICE_FREQ_MULT)

Definition

Used in conjunction with REPRICE_FREQ to define the frequency of rate change of an account.

Module Usage

Oracle ALM and Funds Transfer Pricing reference REPRICE_FREQ when identifying adjustable-rate instruments and calculating a transfer pricing term or forecasted interest rate. REPRICE_FREQ_MULT determines the units (Months, Days or Years) of REPRICE_FREQ.

Data Verification Requirements and Suggested Defaults

Valid values are:

- D - Days
- M - Months
- Y - Years

Residual Amount (RESIDUAL_AMOUNT)

Definition

Used for instruments with AMRT_TYPE_CD = 840 (Lease). This column represents the residual value of the lease at the maturity date.

Module Usage

Oracle ALM and Funds Transfer Pricing reference RESIDUAL_AMOUNT when AMRT_TYPE_CD = 840 (lease) to calculate the monthly payment due on the lease contract.

Data Verification Requirements and Suggested Defaults

- If AMRT_TYPE_CD = 840, RESIDUAL_AMOUNT > 0.
- If AMRT_TYPE_CD = 840, CUR_PAYMENT should include P and I payment amount required to amortize the current principal balance to the residual amount by the maturity date.
- If AMRT_TYPE_CD = 840 and ADJUSTABLE_TYPE_CD <> 0, the cash flow engine will re-compute the CUR_PAYMENT on each reprice date so the principal will amortize to the RESIDUAL_AMOUNT at maturity of the instrument.

Teaser-rate End Date (TEASER_END_DATE)

Definition

Date that the tease rate (introductory rate) ends and the normal product rate begins.

Module Usage

The cash flow engine references TEASER_END_DATE when calculating a forecasted interest rate. The TEASER_END_DATE defines the end of a tease period, which is an initial low interest rate period from the origination of a loan. At the TEASER_END_DATE, the low rate is repriced to a value defined by the market index (IRC) plus margin.

Oracle ALM

1. The cash flow engine does not adjust the rate on an adjustable-rate account until the TEASER_END_DATE is less than the current date within the modeling horizon.

2. TEASER_END_DATE takes precedence over NEXT_REPRICE_DATE and REPRICE_FREQ. Even if NEXT_REPRICE_DATE < TEASER_END_DATE, the record does not reprice until the TEASER_END_DATE.

3. TEASER_END_DATE does not apply to fixed-rate accounts.

Oracle Funds Transfer Pricing

1. Because TEASER_END_DATE overrides the other repricing fields, if a record is currently in its tease period, the last repricing date equals the ORIGINATION_DATE and the next repricing date equals the TEASER_END_DATE.

2. If the TEASER_END_DATE is greater than the AS_OF_DATE, the Oracle Funds Transfer Pricing mid-period repricing does not apply. For further information, see *Oracle Financial Services Funds Transfer Pricing User Guide*.

Data Verification Requirements and Suggested Defaults

- TEASER_END_DATE > ORIGINATION_DATE

- TEASER_END_DATE < MATURITY_DATE

- For fixed-rate accounts TEASER_END_DATE = **01-JAN-1900**.

- TEASER_END_DATE >= NEXT_REPRICE_DATE

Transfer Price Average Life (TP_AVERAGE_LIFE)

Definition

In Oracle Funds Transfer Pricing, this is both an output field and an input field. The TP_AVERAGE_LIFE column can be populated by the Standard FTP process (optionally) when the TP Method selection is Cash Flow: Average Life and the related, update instrument option is selected in the Transfer Pricing rule. If selected, the calculated average life is output to TP_AVERAGE_LIFE column of the instrument record. The average life amount in the TP_AVERAGE_LIFE column, is additionally referenced by TP Adjustment Rules when the Reference Term selection is Average Life.

Module Usage

The Average Life method determines the average life of the instrument by calculating the effective term (expressed in Years) required to repay back half of the principal or nominal amount of the instrument. When applied as a Transfer Pricing method, the resulting TP rate is equivalent to the rate on the associated interest rate curve corresponding to the calculated term. Oracle Funds Transfer Pricing derives the Average Life based on the cash flows of an instrument determined by the characteristics specified in the Instrument Table and additionally applying any specified prepayment assumption. The average life formula calculates a single term, that is a point on the yield curve used to transfer price the instrument being analyzed.

$$Average\ Life = \sum_{i=1}^{n} \frac{P_i}{P}\, t_i$$

Where:

P is the principal

P_i is the principal repayment in coupon i, hence,

$$\frac{P_i}{P}$$

is the fraction of the principal repaid in coupon i and t_i is the time from the start of coupon *i*.

Data Verification Requirements and Suggested Defaults

- If the Adjustment Rule Reference Term selection is Average Life, then TP_AVERAGE_LIFE > 0. Note, this field can be populated manually (Expressed in Years) using an external calculation source or by the Transfer Pricing process.

Transfer Price Duration (TP_DURATION)

Definition

In Oracle Funds Transfer Pricing, this is both an output field and an input field. The TP_DURATION column can be populated by the Standard FTP process (optionally) when the TP Method selection is Cash Flow: Duration and the related, update instrument option is selected in the Transfer Pricing rule. If selected, the calculated Duration is output to the TP_DURATION column of the instrument record. The Duration amount in the TP_DURATION column, is additionally referenced by TP Adjustment Rules when the Reference Term selection is Duration.

Module Usage

Oracle Funds Transfer Pricing derives the Macaulay Duration (in Years) based on the cash flows of each instrument as determined by the characteristics specified in the

Instrument Table and using any applicable prepayment rate. The Macaulay Duration formula calculates a single term, that is, a point on the yield curve used to transfer price the instrument.

$$Duration = \frac{\sum_{n=1}^{N}\left[\frac{CF_n}{(1+r)^m}\ x\ t_n\right]}{\sum_{n=1}^{N}\left[\frac{CF_n}{(1+r)^m}\right]}$$

Where:

N = Total number of payments from Start Date until the earlier of repricing or maturity

CF_n = Cash flow (such as regular principal, prepayments, and interest) in period n

r = Periodic rate (current rate/payments per year)

m = Remaining term to cash flow/active payment frequency

t_n = Remaining term to cash flow n, expressed in years

> **Note:** Remaining term to cash flow is the difference between the date of each cash flow and the modeling start date for that instrument.

Within the Macaulay Duration calculation, the discount rate or current rate, r, is defined in the Transfer Pricing rule in one of three ways:

1. Current rate is defined as current net rate if the processing option, Model with Gross Rates is not selected and current gross rate if the option is selected. The current rate is used as a constant discount rate for each cash flow.

2. The user may directly input, while defining the TP rule, a constant rate to use for discounting. If specified, this rate is used as a constant discount rate for each flow.

3. The user can select to discount the cash flows using spot rates from a selected interest rate curve. With this approach, a discount rate is read from the selected interest rate curve corresponding to the term of each cash flow.

Data Verification Requirements and Suggested Defaults

- If the Adjustment Rule Reference Term selection is Duration, then TP_DURATION > 0. Note, this field can be populated manually (expressed in Years) using an external calculation source or by the Transfer Pricing process.

Transfer Price Effective Date (TP_EFFECTIVE_DATE)

Definition

In Oracle Funds Transfer Pricing, TP_EFFECTIVE_DATE is used by the FTP Standard Process as an override date for determining the historical rate from the specified interest rate curve (IRC). TP_EFFECTIVE_DATE is typically used in cases where the customer rate on the instrument record is set before the origination date and users wish to align the TP Rate lookup with the date that the customer rate was set.

Module Usage

The FTP process normally uses the ORIGINATION_DATE for fixed rate instruments or LAST_REPRICE_DATE for adjustable rate instruments to determine historical rate effective date for the transfer rate lookup. This input allows users to specify an alternate effective date.

Data Verification Requirements and Suggested Defaults

- If applicable, set the TP_EFFECTIVE_DATE = ISSUE_DATE or the date on which the customer rate was set.
- TP_EFFECTIVE_DATE should be **NULL** if not applicable. Do not default TP_EFFECTIVE_DATE (for example, 01-JAN-1900) to a valid date as the TP Engine will reference any valid date.
- TP_EFFECTIVE_DATE <> ORIGINATION_DATE
- TP_EFFECTIVE_DATE <> LAST_REPRICE_DATE

Transfer Rate Margin (MARGIN_T_RATE)

Definition

The spread added to the forecasted transfer rate for adjustable-rate accounts.

Module Usage

Oracle Balance Sheet Planning

1. Oracle ALM no longer references MARGIN_T_RATE when forecasting transfer rates in adjustable-rate records. It is still required a column however for the Oracle Balance Sheet Planning application.

Data Verification Requirements and Suggested Defaults

- Default MARGIN_T_RATE to **0**.

Transfer Rate (TRANSFER_RATE)

Definition

The associated transfer rate for the account, using the standard Transfer Pricing basis.

Module Usage

Oracle ALM

The TRANSFER_RATE from the detail record is used to calculate Cost/Credit for funds (financial element 450) and Weighted Average Transfer Rate (financial element 170) until the record's NEXT_REPRICE DATE. At the NEXT_REPRICE DATE, Oracle ALM refers to the Transfer Pricing rule attached to the ALM process and re-calculates the future TP rates based on the applicable TP Method and TP IRC.

Oracle Funds Transfer Pricing

After calculating a record's transfer rate using one of the transfer pricing methodologies and the standard pricing basis, the result is written out to the TRANSFER_ RATE field.

> **Note:** When the remaining term pricing basis is used, the transfer rate is written to the TRAN_RATE_REM_TERM field.

Data Verification Requirements and Suggested Defaults

- If TRANSFER RATES are to be loaded directly from an external source, data must be in % format (10% = 10 not 0.10).

Transfer Rate Alternate (TRANSFER_RATE_ALT)

Definition

The transfer rate alternate column is an optional output field used in FTP to hold the calculated Transfer Rate when an Alternate Rate Output Mapping rule is attached to the Standard TP Process.

Module Usage

Oracle Funds Transfer Pricing

In Oracle Funds Transfer Pricing, the default behavior is to output transfer pricing results to the standard seeded columns. If additional Transfer Rates are required, users can attach an Alternate Rate Output Mapping rule to the standard TP Process allowing them to choose an alternate column target for populating the TP Rate. TRANSFER_RATE_ALT is a seeded **alternate** column that can be selected in the Alternate Rate Output Mapping rule. Users can also add user-defined alternate output columns and additionally select these within Alternate Output Mapping rules.

Data Verification Requirements and Suggested Defaults

None

Transfer Rate Remaining Term (TRAN_RATE_REM_TERM)

Definition

The associated transfer rate for the account, using the remaining term pricing basis.

Module Usage

Oracle ALM

Oracle ALM does not use TRAN_RATE_REM_TERM.

Oracle Funds Transfer Pricing

After calculating a record's transfer rate using one of the transfer pricing methodologies and the remaining term pricing basis, the result is written out to the TRAN_RATE_REM_TERM field.

> **Note:** When the standard pricing basis is used, the transfer rate is written to the TRANSFER_RATE field.

Data Verification Requirements and Suggested Defaults

- If TRAN_RATE_REM_TERM values are to be loaded directly from external systems, data must be in % format (10% = 10 not 0.10).

Transfer Rate Remaining Term Alternate (TRAN_RATE_REM_TERM_ALT)

Definition

The transfer rate remaining term alternate column is an optional output field used in FTP to hold the calculated Remaining Term Transfer Rate when an Alternate Rate Output Mapping rule is attached to the Standard TP Process.

Module Usage

Oracle ALM

Oracle ALM does not use TRAN_RATE_REM_TERM_ALT.

Oracle Funds Transfer Pricing

In Oracle Funds Transfer Pricing, the default behavior is to output transfer pricing results to the standard seeded columns. If additional Transfer Rates are required, users can attach an Alternate Rate Output Mapping rule to the standard TP Process allowing them to choose an alternate column target for populating the TP Rate. TRAN_RATE_REM_TERM_ALT is a seeded **alternate** column that can be selected in the Alternate Rate Output Mapping rule. Users can also add user-defined alternate output columns and additionally select these within Alternate Output Mapping rules.

> **Note:** When the remaining term pricing basis is used, the transfer rate is written to the TRAN_RATE_REM_TERM field.

Data Verification Requirements and Suggested Defaults

- If TRAN_RATE_REM_TERM_ALT values are to be loaded directly from client systems, data must be in % format (10% = 10 not 0.10).

Derivative Only Columns

Binary Rate (BINARY_RATE)

Definition

The Binary Rate input is used for modeling interest rate caps and floors. No cash flows are exchanged if the floating rate goes earlier the indicated binary rate.

Module Usage

Oracle ALM

In Oracle ALM, when modeling Interest Rate Caps or Floors, the binary rate is an optional input. If specified, binary rate is read by the cash flow engine and will limit the maximum exposure on the floating leg of the contract.

Data Verification Requirements and Suggested Defaults

If applicable, set binary rate > 0, otherwise set binary rate = NULL

Exchange of Principal flag (EXCHG_OF_PRINCIPAL)

Definition

The exchange of principal flag is used for indicating whether, at maturity, the principal flows on the contract are exchanged or not.

Module Usage

Oracle ALM

In Oracle ALM, the exchange of principal flag indicates whether or not principal flows will be exchanged at maturity of the contract.

Exchange of Principal values:

0 = No Exchange of Principal

1 = Exchange of Principal

Data Verification Requirements and Suggested Defaults

For most derivative instruments, Exchange of Principal = 0 (default)

Leg Type (LEG_TYPE)

Definition

The Leg Type column is used for modeling derivative instruments and indicates if the record is a Payable Leg or a Receivable leg. Derivative instruments should be mapped to an Account Type attribute of **Off Balance Sheet**.

Module Usage

Oracle ALM

In Oracle ALM the values for LEG_TYPE are as follows:

0 – Indicates the instrument not a derivative (appears in ALM result tables only)

1 – Indicates the instrument is the payable leg of a derivative contract. Both Caps and Floors will also contain leg type = 1. For Collar contracts, Caps = 1 and Floors = 2.

2 – Indicates the instrument is the receivable leg of a derivative contract.

> **Note:** The leg type indicator will appear in derivative instrument tables and will be read by the cash flow engine to determine if the record is a payable or receivable leg. The leg type is also written out with the results to the ALM Result Master and Result Detail tables.

Data Verification Requirements and Suggested Defaults

For Derivative Instruments – Payable Leg, LEG_TYPE = 1

For Derivative Instruments – Receivable Leg, LEG_TYPE = 2

Minimum Balance Amortizing Swap (MIN_BAL_AMORT)

Definition

The Minimum Balance Amortizing input indicates the minimum balance that a swap will amortize down to. Amortization will stop when it reaches this specified amount.

Module Usage

Oracle ALM

In Oracle ALM, the Minimum Balance Amortizing input applies to Amortizing Swaps and allows users to specify the minimum amortization balance. The cash flow engine will stop amortizing the contract when this amount is reached.

> **Note:** The account type must be set to **Off Balance Sheet** for this field to become active.

Data Verification Requirements and Suggested Defaults

For Amortizing Swap with a minimum balance, MIN_BAL_AMORT > 0 otherwise default to 0.

Purchase Sale Logic (PURCHASE_SALE_LOGIC)

Definition

The Purchase Sale Logic column indicates whether the contract has been purchased or sold by the Bank.

Module Usage

Oracle ALM

In Oracle ALM, the following values should be used for PURCHASE_SALE_LOGIC.

1 – indicates the contract was sold (short).

2 – indicates the contract was purchased (long).

Data Verification Requirements and Suggested Defaults

For Derivative Instruments which are purchased (long), PURCHASE_SALE_LOGIC = 2

For Derivative Instruments which are sold (short), PURCHASE_SALE_LOGICE = 1

Swap Classification Code (SWAP_CLASS_CD)

Definition

The Swap Classification Code is used to identify the type of swap contract.

Module Usage

Oracle ALM, the Swap Classification code column is used to identify the type of swap contract.

Possible Values:

0 – Swap

1 – Swap with embedded option (reserved for future use)

2 – Swaption (reserved for future use)

Data Verification Requirements and Suggested Defaults

For swap contracts, SWAP_TYPE_CD = 0 (Default)

7

Distributed Originations

This chapter describes how new business instruments are generated as part of an Oracle Asset Liability Management (ALM) forecast to replace balances that have run off for a particular product.

This chapter covers the following topics:

- Introduction to Distributed Originations
- New Business from Rollover
- New Originations
- Rollover Accounts
- Current Target Average Approach
- Rolling between Accounts
- Procedures for Implementation

Introduction to Distributed Originations

New business instruments must be generated as part of an Oracle ALM forecast, to replace balances that have run off for a particular product. New business impacts the total amount of income and market value in future periods. Without new business, both values will be understated in future periods because, over time, existing balances run off. The income and market value associated with new business is a function of the balance booked, the rate booked, and the timing of the booking. The balance and rate are determined by the assumptions input by the user. The timing of the bookings is a function of the modeling methodology.

The booking of new business can be viewed as coming from three sources:

- Internal roll-over of existing accounts (hereafter referred to as **roll-over**)
- Rolling of existing accounts into other accounts (hereafter referred to as **roll-into**)

- New money unrelated to existing business (hereafter referred to as **new originations**)

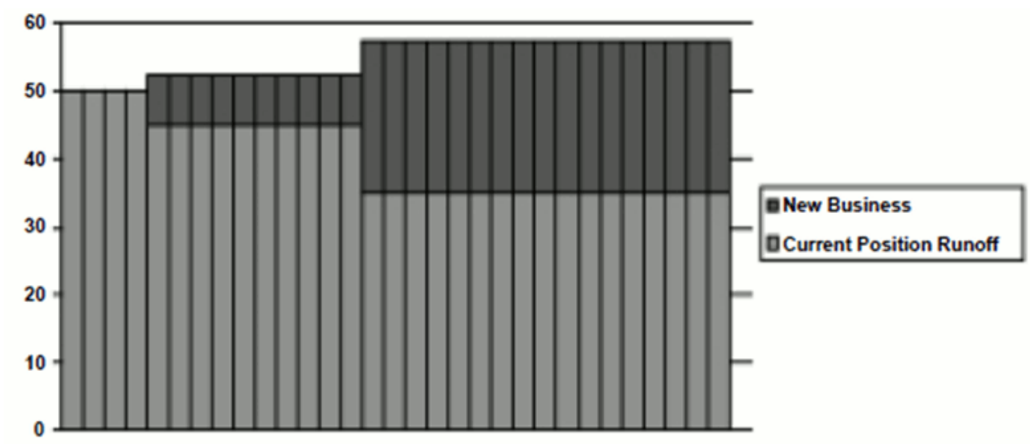
In modeling the timing of new business, the modeling goal is to come as close as possible in reflecting reality while adding minimal processing time.

New bookings generated from run-off (roll-over and roll-into) would realistically occur on the date the run-off occurred. On run-off date, the source of funds becomes available. For a practical example, consider the rollover of a Term Deposit. When a customer opts to rollover a Term Deposit, the funds are rolled into a new Term Deposit on the maturity date of the previous Term Deposit.

New Business from Rollover

To model new business generated from rollover, the new business should be booked on the date the principal run-off occurs. A new record would be created on each date principal run-off occurs during the modeling period, as shown in Figure Roll-over (150% of Principal Runoff), page 7-2.

Roll-over (150% of Principal Runoff)



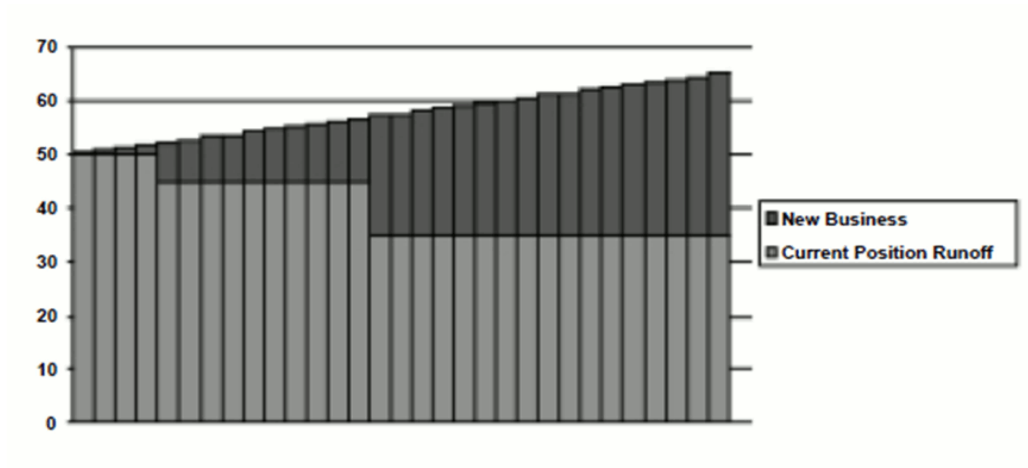

```
Ending Balance = $57.50
Average Balance= ($50*4 days + $52.50*10 days + $57.50*17 days)/31 days
      = $54.9193
```

New Originations

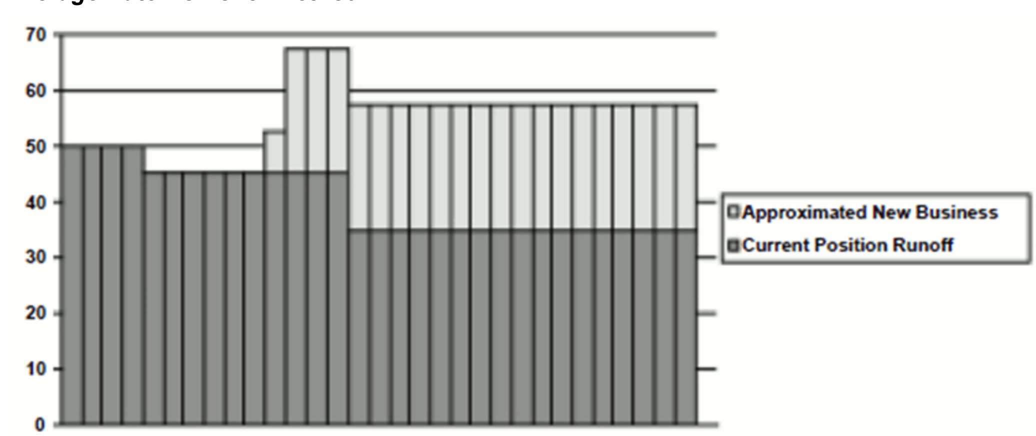
For new originations, users could make several assumptions about the timing. For instance, they could assume that it occurs at a single date within the bucket, that it occurs evenly over the bucket, or that it accelerates over the bucket. For our purposes, we are assuming that new originations are booked evenly over the modeling bucket; the balance will grow gradually over the modeling period.

For new originations, the booking of new accounts are done in a manner that best approximates the user's assumptions about the timing of new business over the bucket. In the case of even bookings over the bucket, the balance in the account appears graphically as a straight, positively sloped line. To match this assumption in the model, a new business record must be generated for each day in the bucket, as shown in Figure- New Originations (30% Growth), page 7-3.

New Originations (30% Growth)



Average Date Roll-over Method



Ending Balance = $65

Average Balance = ($50 + $65)/ 2 = $57.50

$$or = \left(\sum_{n=1}^{31} 50 + \frac{(65-50)}{31} * n^2\right) \div 31$$

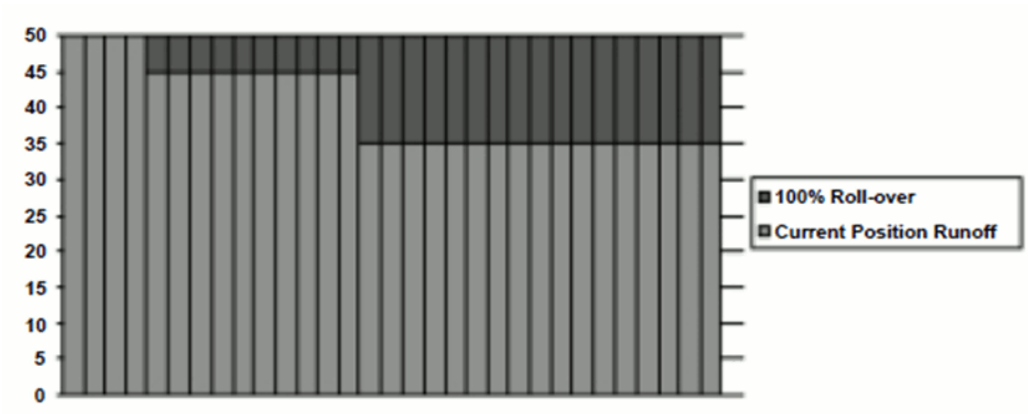
The problem with both of these approaches is performance. The more new business instruments generated per bucket, the longer the processing time will be. For processing efficiency, it is necessary to minimize the number of dates during the bucket when new business is added.

To create the most accurate results, the new business should come as close as possible to matching the area between the new business curve and the current position curve. By matching the area under the curve, we generate the same ending balance and average balance as would be generated if new bookings were made every day.

Rollover Accounts

For rollover accounts, the area under the curve is a function of the timing of runoff and the amount of principal runoff. By calculating an average day of run-off, the area under the new business curve can be matched by generating new bookings on the average date, as shown in Figure- Flat Balance Sheet, page 7-4.

Flat Balance Sheet




```
Average day of runoff  = ($5 * 5 days + $10 * 15 days)/$15 = 11 2/3 days
Balance Plug:          1/3 * $15* 1.5 = $7.50 on the 11th
                        2/3 * $15* 1.5 = $15 on the 12th

Ending Balance        = $57.50

Average Balance=($50 * 4 days + $45 * 6 days + $52.50 * 1 day + $67.50 *
3 days + $57.50 * 17 days)/ 31 = 54.9193
```

When comparing the average and ending balance with the numbers generated from Figure - Flat Balance Sheet, page 7-4, note that the values are exactly the same. This is because we have changed the shape of the new business curve, but matched the area under the curve.

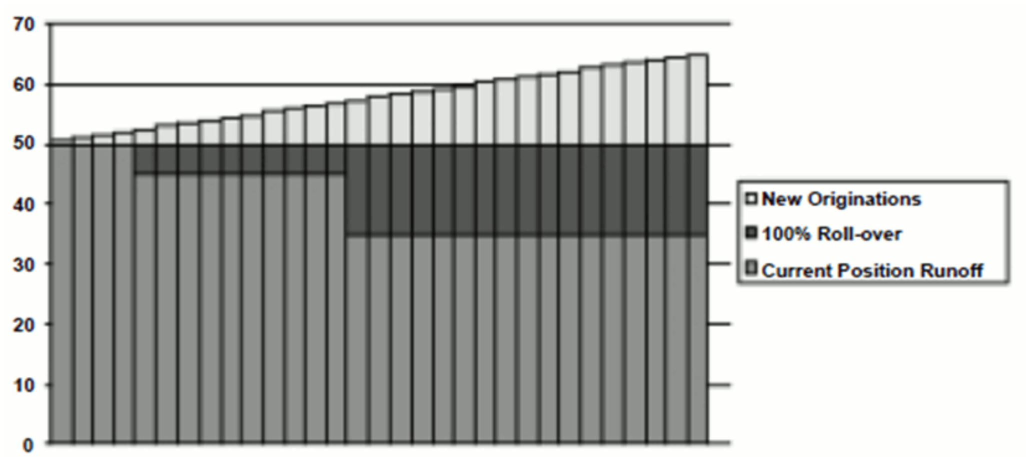
For new originations, the area under the curve is a function of the targeted change in the balance over the bucket, the timing of the runoff of the current position at the start of the bucket, and the amount of principal runoff.

This can broken into two components:

- A component of new business that assumes a flat balance sheet. This is equal to 100% roll-over, as shown in Figure- Flat Balance Sheet, page 7-4.

- The new origination component, which can be viewed as a gradually increasing balance over the flat balance sheet, as shown in Figure- New Originations Separated by Components, page 7-5.

New Originations Separated by Components



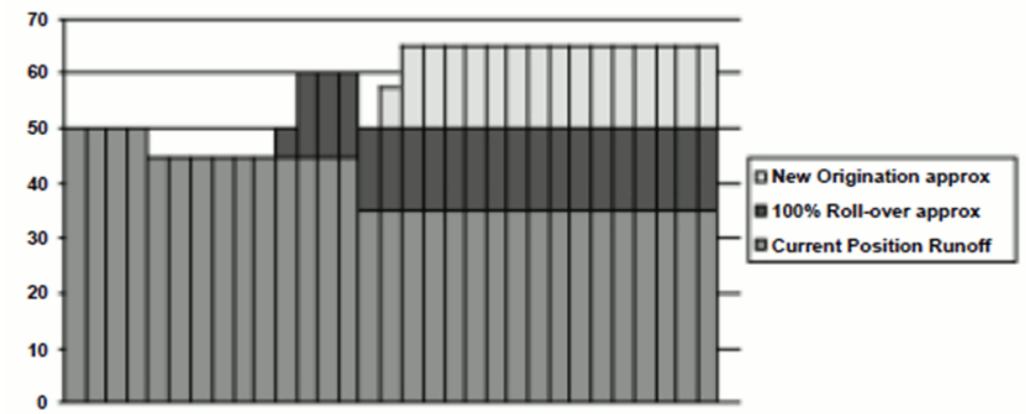
New Business is approximated by splitting the new business into the two components: the average day roll-over method displayed in example 3 to account for the 100% roll-over component of new business plus an additional method to account for the new originations.

That is because, the shape of the new originations curve is a triangle, the area of this shape can be calculated as

*1/2 days in bucket * new origination balance*.

The same area can be achieved by booking the entire balance at the mid-point of the bucket. Because, in this case, the mid-point of the bucket falls between two days, the balance should be spread evenly over those two days: $7.50 on the 15th and $7.50 on the 16th, as shown in Figure- New Origination Average Date Method, page 7-6.

New Origination Average Date Method



Ending Balance = $65

*Average Balance = $50 * 4 days + $45 * 6 days + $50 * 1 day + $60 * 3 days + $50.00 *1 day + 57.50 * 1 day + 65 * 15 days = $57.50*

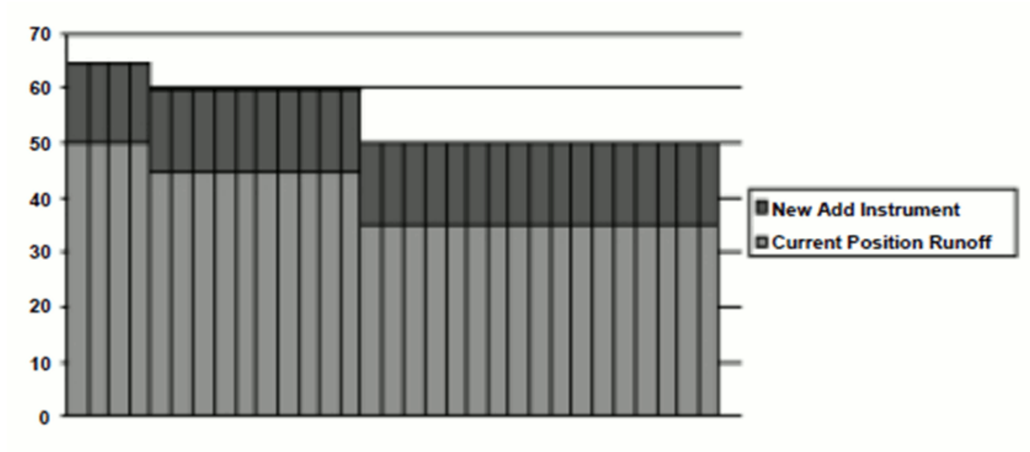
These approaches assume that, after new business has been added, no payments will not occur within the same bucket. To avoid this issue, any payments that occur within the bucket must trigger a secondary set of new originations.

Current Target Average Approach

In the current target average approach, a new business instrument is booked on the first day of the bucket so that the average of this instrument plus the average of the current position will equal the targeted average balance.

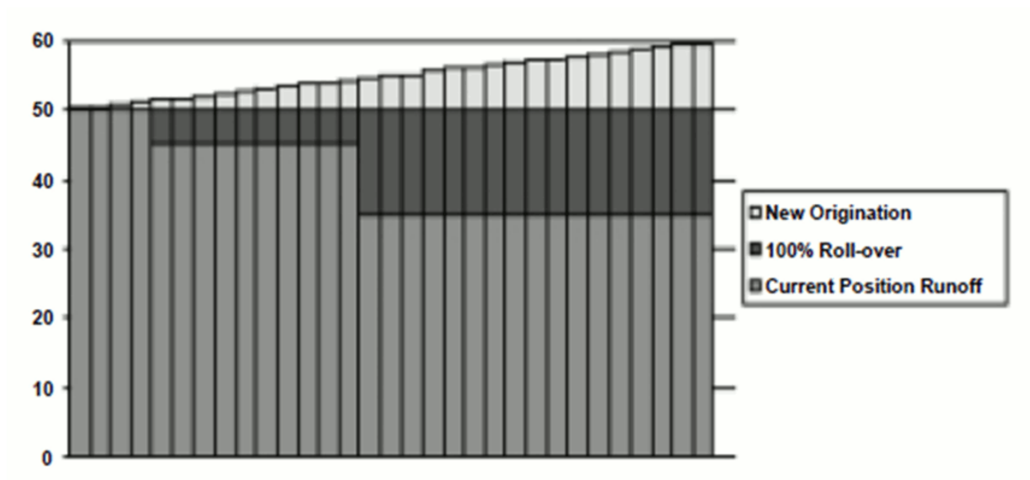
The problem with this approach is that the results are not always logical to an end-user because they do not follow the business logic of new business generated throughout the bucket. In the case represented in Figure Current Average Balance Method, page 7-7, the user has targeted an average balance of $55 from a starting balance of $50. The logical assumption would be that the balance grows over the bucket from $50 to a final value greater than the average. However, in this case, the ending balance is $49.84, lower than the beginning balance. Although the average balance is correct, the ending balance makes no sense.

Current Average Balance Method



A more appropriate method would be to assume a gradual booking over the month to generate the average balance. In this case, an approach similar to the target end method could be used, where both average date and mid-bucket methods are used as shown in Figure Targeting an Average Balance, page 7-7.

Targeting an Average Balance



This example is the same as the example in Figure New Originations Separated by Components, page 7-5 with slightly different ending balance values. The method used to solve this case can be the same one that is demonstrated in Figure New Origination Average Date Method, page 7-6, using a combination of the average date method and the mid-bucket method.

Rolling between Accounts

Inter-account rolling can be achieved with no difficulty if the new instruments do not pay during the modeling bucket in which they are originated. In this simple case, the current position runoff that occurs in bucket 1 will define the new business generated in bucket 1. The new business run-off generated in bucket 2 plus the current position run-off in bucket 2 will impact the business generated in bucket 2, and so on.

If the instruments do pay during the bucket in which they are originated, then, ideally, a new business instrument should be created on the day of run-off for the new instrument for the account that it is rolling into.

Procedures for Implementation

All of the methods described earlier employ one of two techniques: Average Day of Runoff and Mid-Bucket. The implementation of these techniques is described per method as follows:

Forecast Balance Method	Previous Approach	New Approach
Target End	New instrument added at end of bucket equal to target balance minus the current position ending balance.	Average day of runoff with 100% roll-over plus mid-bucket on new add balance.
Target Average	New instrument added at beginning of bucket so that the current position average balance plus the average balance of the new instrument equals the target average balance.	Average day of runoff with 100% roll-over plus mid-bucket on new add balance.
New Add End	New instrument is added at the end of the bucket equal to the new add balance.	Mid-bucket on new add balance.

Forecast Balance Method	Previous Approach	New Approach
Roll-over	On non-amortizing accounts, new business produced from maturing balances is booked on day of run-off. All other run-off from non-amortizing instruments (prepayments, payments, total) is booked at end of bucket. New business generated from all run-off on amortizing accounts is booked at end of bucket.	Average day of runoff with roll-over percentages specified by user.
Roll-into	New business is generated at end of bucket	Average day of runoff with roll-over specified.
Target Growth	Did not exist.	Average day of runoff with 100% roll-over plus mid-bucket on new add balance.

In each case where mid-bucket logic is used, the new add balance will be calculated in a different manner. Following are the different methods for new add balance calculation:

Target End - Target End Balance - Beginning Balance of bucket

Target Average - (Target Average Balance - Beginning Balance of bucket)* 2/(days in bucket - 1)

New Add - New Add Balance

Target Growth - Beginning Balance of bucket * growth percent

Mid-Bucket

1. Determine the mid-point of the bucket by taking the total number of days in the bucket and dividing by two.

2. If the mid-point is not a fractional date, add all new add balance on the mid-point date.

3. If the mid-point is a fractional date, add half of the new add balance on each day.

Average Day of Runoff

1. Calculate and output the average day of runoff for each type of runoff: maturity, prepayments, and payment. Calculate as the balance times the difference between

the date of runoff minus the beginning bucket date. Independently store values for positive and negative runoff of each runoff category.

2. For each bucket and each type of runoff, calculate the amount of balance to roll-over by multiplying the run-off amount by the percent roll-over specified by the user.

3. Add new business on the average date of run-off for each instrument using the proper balance roll-over.

4. If the new records make payments during the bucket, roll the instruments over at the time of payment.

Further Optimizations

When combined to produce a balance forecast for a particular product member, it could result in a large number of new originations on different dates, because of all these methods. The logic finds the earliest date and the latest date of all the different possible originations and adds only new business on these two dates within the bucket.

The first and last plug dates are determined by finding the minimum and maximum date over all average date values. For example, given an average runoff date for payments of 13.4 and an average plug date for maturing balances of 15.6, four plug dates will be derived: 13, 14, 15, and 16. The first plug date will be the minimum date, the 13th. The second plug date will be the maximum plug date, the 16th.

When determining how much of the run-off balance to plug on each date, the calculation will be done as follows:

$\textit{Plug Balance}_{(\textit{first plug})} = \textit{Run-off} * (\textit{first plug date} - \textit{average run-off date}) / (\textit{second plug date} - \textit{first plug date})$

$\textit{Plug Balance}_{(\textit{second plug})} = \textit{Run-off} * (\textit{second plug date} - \textit{average run-off date}) / (\textit{second plug date} - \textit{first plug date})$

A Note About Dynamic Market Value and Gap Results

The dynamic market value and dynamic gap values will be sensitive to the booking date if the dynamic valuation date is in the middle of a modeling bucket. To prevent this type of sensitivity, you should limit dynamic valuation dates to the start date for a bucket.

A Note About Beginning Versus End of Date Averages

Oracle ALM assumes that all events occur at the end of the day on which they occur.

8

Transfer Pricing Option Costs

This chapter describes the option cost calculations implemented in Oracle Funds Transfer Pricing, including theoretical overview, the mathematical details of the calculations, and examples.

This chapter covers the following topics:

- Overview of Transfer Pricing Option Costs
- Architecture of the Calculations
- Theory
- Model Usage Hints

Overview of Transfer Pricing Option Costs

The purpose of option cost calculations is to quantify the cost of optionality, in terms of a spread over the transfer rate, for a single instrument. The cash flows of an instrument with an optionality feature change under different interest rate environments and thus should be priced accordingly.

For example, many mortgages may be prepaid by the borrower at any time without penalty. In effect, the lender has granted the borrower an option to buy back the mortgage at par, even if interest rates have fallen in value. Thus, this option has value.

In another case, an adjustable rate loan may be issued with rate caps (floors) that limit its maximum (minimum) periodic cash flows. These caps and floors constitute options. For the lender, the option cost of a cap is positive and negative for a floor.

Such flexibility given to the borrower raises the bank's cost of funding the loan and affects the underlying profit. The calculated cost of these options may be used in conjunction with the transfer rate to analyze profitability.

Oracle Funds Transfer Pricing uses the Monte Carlo methodology to calculate option cost. This methodology is described in Monte Carlo Analytics, page 11-1 in the context of Oracle Asset Liability Management (ALM). Although Monte Carlo simulation in Oracle ALM generates different types of results than in Transfer Pricing, the underlying calculations are very similar.

In Oracle Funds Transfer Pricing, the option cost is calculated indirectly. Oracle Funds Transfer Pricing calculates and outputs two spreads:

- static spread
- option-adjusted spread (OAS)

The option cost is defined by:

- option cost = static spread – OAS

In the theory section we show that the static spread is equal to margin (credit spread) and the OAS to the risk-adjusted margin of an instrument. Therefore, the option cost quantifies the loss or gain due to risk.

Architecture of the Calculations

For clarity, this chapter assumes the following:

- The instrument pays K cash flows occurring each at the end of the month.
- Each month has the same duration, in number of years, such as 1/12.

Definitions

Neither the static spread nor the OAS can be defined directly as they are solutions of two different equations. We give hereafter a simplified version of the equations that the system solves, using the assumptions described earlier. The static spread is the value *ss* that solves the following equation:

Equation 8–1

$$MV = \sum_{k=1}^{k} \frac{CF(K)}{\prod_{j=0}^{k-1}(1+f(j)+ss)^{\Delta t}}$$

Where:

: MV	market, or book, or par value of the instrument (as selected in the Transfer Pricing rule)
CF(k)	cash flow occurring at the end of month *k* along the forward rate scenario
:f(j)	forward rate for month *j*
Δt	length (in years) of the compounding period; hard-coded to a month, such as 1/12

In the Monte Carlo methodology, the **option adjusted-spread** is the value OAS that solves the following equation:

Equation 8–2

$$MV = \sum_{\omega=1}^{N} \sum_{k=1}^{K} CF(k,\omega) D(k,\omega,OAS)/N$$

Where:

: N	total number of Monte Carlo scenarios
CF(K,ω)	cash flow occurring at the end of month *k* along scenario **ω**
D(k, ω,OAS)	fstochastic discount factor at the end of month *k* along scenario **ω** for a particular OAS (See Architecture of the Rate Generator, page 11-3.

Note that:

- Cash flows are calculated up till maturity even if the instrument is adjustable.

  Note: Otherwise the calculations would not catch the cost of caps or floors.

- In the real calculations, the formula for the stochastic discount factor is simplified.

Example

In this example, the Transfer Pricing curve is the Treasury curve. It is flat at 5%, which means that the forward rate is equal to 1%. We use only two Monte Carlo scenarios:

- up scenario: 1-year rate one year from now equal to 6%.
- down scenario: 1-year rate one year from now equal to 5%.

Observe that the average of these 2 stochastic rates is equal to 5%.

The instrument record is 2 year adjustable, paying yearly, with simple amortization. Its rate is Treasury plus 2%, with a cap at 7.5%. Par value and market value are equal to $1.

For simplicity in this example we assume that the compounding period used for discounting is equal to a year, for example:

Equation 8–3

$$\Delta t = 1$$

The static spread is the solution of the following equation:

Equation 8–4

$$1 = \frac{0.07}{1 + 0.05 + ss} + \frac{1 + 0.07}{(1 + 0.05 + ss)^2}$$

It is intuitively obvious that the static spread should be equal to the margin , for example:

> **Note:** We prove this claim in the **Theory** section of this chapter.

static spread = coupon rate - forward rate= 7%-5%=2%.

Plugging this value in the right side of the earlier equation yields:

Equation 8–5

$$\frac{0.07}{1+0.05+ss}+\frac{1+0.07}{(1+0.05+ss)^2}=$$

$$\frac{0.07}{1+0.07}+\frac{1+0.07}{(1+0.07)^2}=$$

$$\frac{0.07*(1+0.07)+1+0.07}{(1+0.07)^2}=$$

$$\frac{0.07+0.07^2+1+0.07}{(1+0.07)}=1$$

This is equal to par. Our guess was correct.

The OAS is the solution of the following equation:

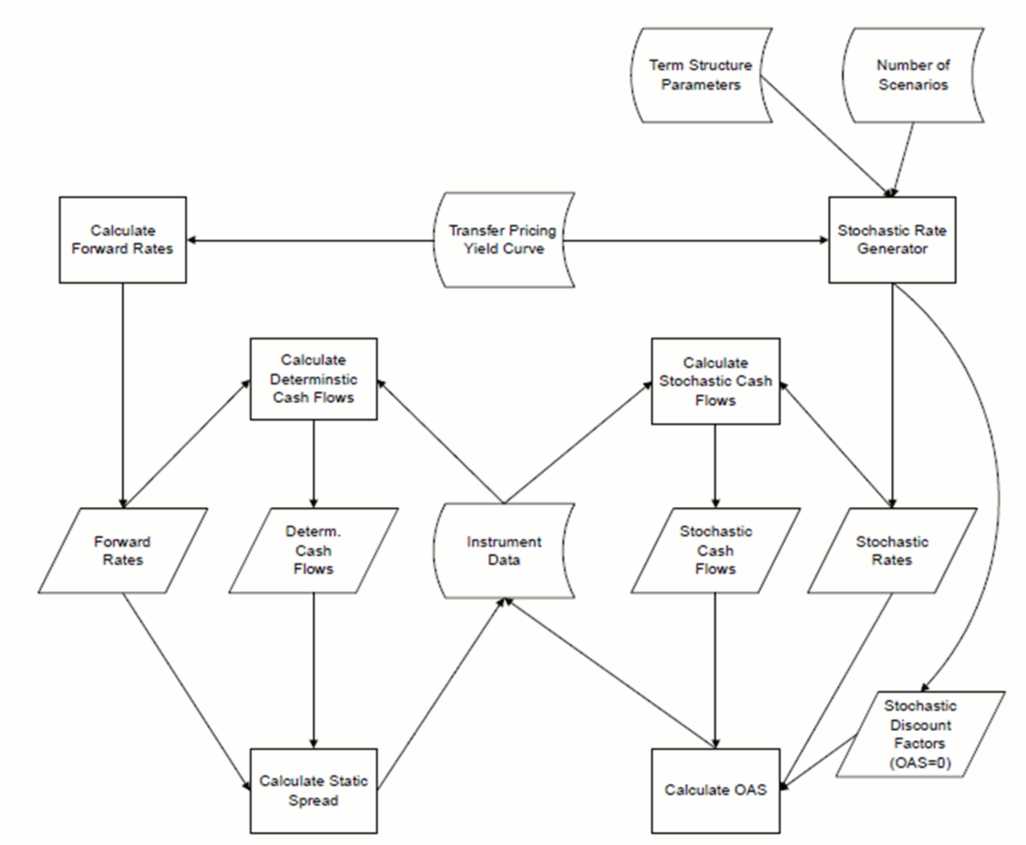
Equation 8–6

$$1=\frac{0.07}{1+0.05+OAS}+\frac{1}{2}\left[\frac{1+0.075}{(1+0.05+OAS)(1+0.06+OAS)}+\frac{1+0.06}{(1+0.05+OAS)(1+0.04+OAS)}\right]$$

By trial and error we find a value of 1.88%. To summarize:

option cost = static spread - OAS=2%-1.88%=12 bp

Process Flow



In this section we describe only the following steps:

- Calculate forward rates
- Calculate static spread
- Calculate OAS

The other steps are described later.

Calculate Forward Rates

The cubic spline interpolation routine first calculates smoothed continuously compounded zero-coupon yields *Y(j)* with maturity equal to the end of month *j*. The formula for the one-month annually compounded forward rate spanning month *j+1* is:

Equation 8–7

$$f_j = \exp[(Y_j(j+1) - Y(j)j] - 1$$

Calculate Static Spread

The static spread is calculated using Newton-Raphson algorithm. In the event that Newton-Raphson algorithm does not converge , we revert to a brute search algorithm, which is much slower.

> **Note:** This can happen if cash flows alternate in sign.

The user can control the convergence speed of the algorithm by adjusting the value of the variable Option Cost Speed Factor. This variable is defined in the Oracle FTP Application Preferences screen.

The default value is equal to one. A lower speed factor results in a better accuracy of the results :

> **Note:** In all our experiments, a speed factor equal to one resulted in a maximum error (on the static spread and OAS) lower than half a basis point.

For convenience we recap hereafter Newton's algorithm. Let *x* be the static spread. At each iteration *m* we define the function *F(m)* by the following equation:

Equation 8–8

$$F(m) = MV - \sum_{k=1}^{k} \frac{CF(K)}{\prod_{j=0}^{k-1}(1 + f(j) + x(m))^{\Delta t}}$$

The algorithm is:

Equation 8-9

$$
\begin{aligned}
&m = 0\\
&\text{INITIALIZE } x(m)\\
&\text{REPEAT}\\
&\quad m = m+1\\
&\quad \text{CALCULATE } F(m), F'(m)\\
&\quad x(m+1) = x(m) - \frac{F(m)}{F'(m)}\\
&\text{UNTIL} \left|\frac{F(m)}{MV}\right| < tol * \text{OptionCostSpeedFactor OR } m > \text{MaxIterations}\\
&\text{IF} \left|\frac{F(m)}{MV}\right| > tol * \text{OptionCostSpeedFactor REVERT TO BRUTE SEARCH}
\end{aligned}
$$

For performance reasons, the code uses a more complicated algorithm albeit similar in spirit than the one described earlier. This is the reason why we did not give specific values for tol (tolerance) and Max Iterations, or details on the brute search.

Calculate OAS

For fixed rate instruments, such as instruments for which the deterministic cash flows are the same as the stochastic cash flows, the OAS is by definition equal to the static spread.

> **Note:** This statement is true in the case of continuous compounding. For discrete compounding, this approximation has a negligible impact on the accuracy of the results.

The OAS is also calculated with an optimized version of Newton-Raphson algorithm. To get a gist of the Newton-Raphson method, refer to the previous section, with the following substitutions:

Equation 8–10

$$OAS = x(m)$$

$$F(m) = MV - \sum_{\omega=1}^{N} \sum_{k=1}^{k} CF(k,\omega) D(k,\omega,x(m)) /N$$

Theory

In this section we show that the static spread is equal to margin and the OAS to the risk-adjusted margin of an instrument, when the user selects the market value of the instrument to equate the discounted stream of cash flows. We assume in this section that the reader has a good knowledge of no-arbitrage theory.

We first need some assumptions and definitions

- To acquire the instrument, the bank pays an initial amount *V(0)*, the current market value.
- The risk-free rate is denoted by *r(t)*.
- The instrument receives a cash flow rate equal to *C(t)*, with $0 \leq t \leq T$. (= maturity)
- The bank reinvests the cash flows in a money market account that, with the instrument, composes the *portfolio.*
- The *total return* on a portfolio is equal to the expected future value divided by the initial value of the investment.
- The *margin p* on a portfolio is the difference between the rate of return (used to calculate the total return) and the risk free rate *r* .
- The *risk-adjusted expected future value* of a portfolio is equal to its expected future value after hedging all diversifiable risks.
- The *total risk-adjusted return* of a portfolio is equal to the risk-adjusted expected future value divided by the initial value of the investment.
- The *risk-adjusted margin m* of a portfolio is the difference between the risk-adjusted rate of return (used to calculate the total risk-adjusted return) and the risk-free rate *r* .

More precisely,

Equation 8–11

$$\text{Total Return} = \mathrm{E}[\mathrm{e}^{\int_t^T r(t)+pdt}]$$

Equation 8–12

$$\text{Total Risk} - \text{Adjusted Return} = \mathrm{E}[\mathrm{e}^{\int_t^T r(t)+mdt}]$$

Equivalence of the OAS and Risk-Adjusted Margin

In a no-arbitrage economy with complete markets, the market value at time *t* of an instrument with cash flow rate *C(t)* is given by:

Equation 8–13

$$V(t) = E_t\left[\int_0^T C(t)e^{-\int_t^t r(s)\,ds}\,dt\right]$$

and expectation is taken with respect to the risk-neutral measure. The expected change in value is given by:

Equation 8–14

$$E_t[dV] = E_t\,[\lim_{\Delta\to 0}\frac{V(t+\Delta)-V(t)}{\Delta t}\,] = E_t\,[\lim_{\Delta\to 0}\frac{E_{t+\Delta}[V(t+\Delta)]-V(t)}{\Delta}\,] =$$

$$E_t[\lim_{\Delta\to 0}\frac{\int_{t+\Delta}^{I} C(u)e^{-\int_{t+\Delta}^{u} x(s)\,ds)}\,du - \int_t^I C(u)e^{-\int_t^u x(s)\,ds}\,du}{\Delta}\,] = E_t[\frac{d}{dt}\int_t^I C(u)e^{-\int_t^u x(s)\,ds}\,]=$$

$$E_t[r(t)V(t) - C(t)]$$

The variation in value is therefore equal to the expected value of the change *dV* plus the change in value of a martingale *M* in the risk-neutral measure:

Equation 8–15

$$dV(t) = E_t\,[dV] + dM = rVdt - Cdt + dM$$

Let *I* be the market value of the money market account in which cash flows are reinvested.

Equation 8–16

$$I(T) = \int_0^T C(t)e^{\int_t^T r(s)\,ds}\,dt$$

Note that unlike *V* this is a process of finite variation. By Ito's lemma, *Equation 8–17*

$$dI = rIdt + Cdt$$

Let *S* be the market value of a portfolio composed of the instrument plus the money market account. We have:

Equation 8–18

$$ds = dV + dI = rSdt + dM$$
$$S(0) = V(0)$$

In other terms, the portfolio (and not the instrument) earns the risk-free rate of return. An alternate representation of this process is:

Equation 8–19

$$\frac{dS}{S} = rdt + dN$$

where *N* is another martingale in the risk-neutral measure. The expected value of the portfolio is then:

Equation 8–20

$$E[S(T)] = V(0)E\left[e^{\int_0^T r(t)dt + N(t) - \frac{1}{2}<N,N>T}\right]$$

Where

<N,N>

is the quadratic variation of *N*. This is equivalent to:

Equation 8–21

$$E[S(T)] = V(0)E\left[e^{\int_0^T r(t)dt + N(t) - \frac{1}{2}<N,N>T}\right]$$

Let us define the martingale:

Equation 8–22

$$Z = e^{N - \frac{1}{2}<N,N>}$$

This represents the relative risk of the portfolio with respect to the standard money market account, that is, the account where only an initial investment of *V(0)* is made. Then,

Equation 8–23

$$E[S(T)] = V(0)E[e^{\int_0^T r(t)\,dt}](1 + \mathrm{corr}\,(e^{\int_0^T r(t)\,dt}, Z(T))$$

In other terms the expected future value of the portfolio is equal to the expected future value of the money market account adjusted by the correlation between the standard money market account and the relative risk. Assuming complete and efficient markets, banks can fully hedge their balance sheet against this relative risk, which should be neglected to calculate the contribution of a particular portfolio to the profitability of the balance sheet. Therefore,

Equation 8–24

$$\text{Risk – adjusted expected future value} = \frac{E[S(T)]}{(1+\mathrm{corr}\,(e^{\int_0^T r(t)\,dt}, Z(T))}$$

Equation 8–25

$$\text{Total Risk – adjusted return} = \frac{E[S(T)]}{V(0)(1+\mathrm{corr}\,(e^{\int_0^T r(t)\,dt}, Z(T))}$$

In this example, the risk-adjusted rate of return of the bank on its portfolio is equal to the risk-free rate of return.

Let us suppose now that another instrument offers cash flows

C'>C

Assuming complete and efficient markets its market value will then be:

Equation 8–26

$$V'(0) = E[\int_0^T C'(t)e^{-\int_0^t r(s)\,ds}\,dt] > V(0)$$

The value of the corresponding portfolio is denoted by

S'>S

By analogy with the previous development, we have:

Equation 8–27

$$\text{Total Risk} - \text{adjusted return} = \frac{E[S'(T)]}{V'(0)(1+\text{corr}\,(e^{\int_0^T r(t)\,dt}, Z'(T))}$$

Again, the risk-adjusted rate of return of the bank on its portfolio is equal to the risk-free rate of return. Suppose now that markets are not complete and inefficient. The bank pays the value *V(0)* and receives cash flows equal to *C′*. We have:

Equation 8–28

$$\text{Total Risk} - \text{adjusted return} = \frac{E[S'(T)]}{V(0)(1+\text{corr}\,(e^{\int_0^T r(t)\,dt}, Z'(T))}$$

By definition of the total risk-adjusted return for Equation 8–12, page 8-9 we have:

Equation 8–29

$$E[S'(T)] = V(0)E[e^{\int_0^T r(t)+mdt}](1 + \text{corr}\,(e^{\int_0^T r(t)+mdt}, Z'))$$

Therefore, by analogy with the previous development,

Equation 8–30

$$dS' = (r + m)S' + dM$$

$$S(0) = V(0)$$

This can be decomposed into

Equation 8–31

$$dV' = (r+m)\,V'dt - C'\,dt + dM'$$

Equation 8–32

$$dI' = rI'dt + C'dt$$
$$dS' = dV' + dI'$$

Equation 8–33

$$V'(0) = V(0)$$

The solution of Equation 8–31, page 8-14 and Equation 8–33, page 8-14 is:

Equation 8–34

$$V'(0) = V(0) = E[\int_0^T C'(t)e^{-\int_0^t f(s)+p ds}\,dt\,]$$

By the law of large numbers, Equation 8–31, page 8-14 and Equation 8–33, page 8-14 result in:

Equation 8–35

$$OAS = m$$

For example, the OAS is equal to the risk-adjusted margin.

Equivalence of the Static Spread and Margin

Static spread calculations are deterministic, therefore they are a special case of the equations in the previous section where, roughly speaking, all processes are equal to their expected value, and the margin p is substituted to the risk-adjusted margin m. The equivalent of Equation 8–29, page 8-13 is then

Equation 8–36

$$S'(T) = V(0)\, e^{\int_0^T f(t)+p\,dt}$$

Where *f* is the instantaneous forward rate. The equivalent of Equation 8–31, page 8-14 and Equation 8–33, page 8-14 is then

Equation 8–37

$$dV' = (r+m)\, V'dt - C'\, dt$$

Equation 8–38

$$dI' = rI'dt + C'dt$$
$$dS' = dV' + dI'$$

Equation 8–39

$$V'(0) = V(0)$$

The solution of Equation 8–37, page 8-15 and Equation 8–39, page 8-15 is:

Equation 8–40

$$V'(0) = V(0) = \int_0^T C'(t)e^{-\int_0^t f(s)+p\,ds}\, dt\,]$$

Comparing Equation 8–40, page 8-15 and Equation 8–8, page 8-7:

Equation 8–41

$$SS=P$$

For example, the static spread is equal to the margin.

Model Usage Hints

Non-Unicity of the Static Spread

In some very rare cases there is more than one value that solves the static spread equation. This section describes one of these cases.

The market value of the instrument is $0.445495. It has 2 cash flows. The following table shows the value of the cash flows and corresponding discount factors (assuming a static spread of zero).

	First event	**Second event**
Time	1	2
Cash flow value	1	-0.505025
Discount factor(static spread=0)	0.9	0.8

The continuously compounded static spread solves the following equation:

Equation 8–42

$$0.9 \text{ Exp}(-ss) - 0.8*0.505025 \text{ Exp}(-2ss) - 0.445494 = 0$$

There are two possible solutions for the static spread:

- static spread =0.19%
- static spread =1.81$

Accuracy

In case you desire a better numerical precision than the default precision, you can take two actions:

- Decrease the speed factor (see the section Calculate Static Spread, page 8-7).
- Increase the number of Monte Carlo scenarios.

Both actions increase calculation time.

9

Rate Conversion

This chapter defines interest rate formats used within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing, describes which format is used in a particular process and what type or types of conversion are used; and outlines the algorithms used for such conversions.

Interest rates come in a variety of formats. Within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing (FTP), interest rates are used for multiple purposes, with each rate requiring a specific format. The system must apply conversion formulas to translate the interest rates from their starting format into the format proper for its use in any given process.

This chapter covers the following topics:

- Definitions
- Rate Format Usage
- Rate Conversion Algorithms

Definitions

The following characteristics define an interest rate code:

- accrual basis
- compound basis
- rate format

The accrual basis can be:

- 30/360
- 30/365
- 30/Actual

- Actual/Actual
- Actual/365
- Actual/360

> **Note:** See Stigum, M., and Robinson *Money Market and Bond Calculations* , Irwin, 1981, for definition

The compound basis can be:

- daily
- monthly
- quarterly
- semi-annual
- annual
- simple

The rate format can be

- zero-coupon yield
- yield-to-maturity
- discount factor

Discount factor is used only internally and cannot be specified as an input rate format in Rate Management Interest Rate setup. For bonds issued at par with payment frequency equal to the compound basis, the yield-to-maturity at origination is equal to the coupon. There are several definitions of yield-to-maturity. The unconventional **true yield** definition of Stigum is not used. Instead, the Street convention is preferred.

> **Note:** See Fabozzi, F. *The Handbook of Fixed Income Securities*. McGraw Hill, 1977.

The yield curve is composed of r par bonds with different terms. Par value is equal to $1.

Symbol	Name	Notes
$AI(T_i)$	Accrued interest for the i-th security.	
$C(T_i)$	Coupon value of the i-th security.	This is the true $value of the cash flow (not annualized)
m	Compounding frequency (per year).	Possible values are: 12 – monthly 2 – semi - annual 1- annual 0- Simple.
n_i	Number of full compounding periods from As of Date up to term T_i	
P(T)	Discount factor with term T_i	Value of a zero-coupon bond with term T_i and par=$1.
r	Total number of securities.	
T_i	Term in Act/Act years of the i-th security.	Sorted in ascending order.
L(i,k)	Time in Act/Act years of the start of the k-th (k=0 …n_i) compounding period for security i.	
w_i	Residual, that is, number of compounding periods between the current date and the next compounding date for i-th security.	

Symbol	Name	Notes
$y_{TM}(T_i)$	Yield-to-maturity of the i-th security	
$y_{zc}(T_i)$	Zero-coupon yield of the i-th security	

The **zero-coupon yield** is the vector of *r* values $y_{zc}(T_i)$ that solve the following set of r equations:

Equation 9–1

$$1 + AI(T_i) = \frac{1+c(T_i)}{\left(1+\frac{y_{zc}(T_i)}{m}\right)^{n_i}}$$

If compounding is simple, or

Equation 9–2

$$1+ AI\ (T_i) = \sum_{k=0}^{n_i} \frac{c(T_i)}{\left(1+\frac{y_{ZC}(L(i,k))}{m}\right)^{k+w_i}} + \frac{1}{\left(1+\frac{y_{ZC}(L(i,n_i))}{m}\right)^{n_i+w_i}}$$

otherwise

The yield-to maturity for the i-th security is the value $y_{TM}(T_i)$ that solves the equation.

Equation 9–3

$$1+ AI(T_i) = \frac{1+ C(T_i)}{1+\frac{w_i}{m} y_{TM}(T_i)}$$

If compounding is simple, or

Equation 9–4

$$1 + AI(T_i) = \sum_{k=0}^{n_i} \frac{c(T_i)}{\left(1+\frac{y_{TM}(T_i)}{m}\right)^{k+w_i}} + \frac{1}{\left(1+\frac{y_{TM}(T_i)}{m}\right)^{n_i+w_i}}$$

otherwise.

Rate Format Usage

Scenario-based Forecast Rates

Scenario-based forecast rates are the interest rates used for repricing events, rate setting of new business, and market rate observations for prepayment assumptions and new business assumptions. The Forecast Rates assumption rule stores the definition of forecasted rate scenarios.

Within the Forecast Rates assumption rule, the user creates deterministic scenario(s) for each IRC, defining the forecasted rates in each modeling bucket (for an **active** time bucket rule) for each IRC and scenario. Most methods use the as-of-date rates, stored in history tables, as the basis for the rates forecast. The following methods are available:

- Flat
- Structured Change
- Direct Input
- Change from Base
- Implied Forward
- Yield Curve Twist

Flat

The flat method assumes rates in effect on the as of date remain constant for the entire term structure over the modeling horizon.

Structured Change

The user inputs a series of rate changes that can vary over the yield curve terms and over the modeling horizon. At runtime and display time, the rate changes are added to the as-of-date rates to create the future scenario. No conversion is applied before the rate is passed to the Cash Flow Engine.

Direct Input

The Direct Input method requires the user to input (or upload) of the forecasted rate scenarios. At runtime and display time, the inputted rates will be pulled directly.

Prior to applying the rates to a cash flow record, the rates must be converted to annual compounded rates. The accrual basis must be converted to either Actual/Actual or 30/360.

No conversion is applied before the rate is passed to the Cash Flow Engine.

Change From Base

For change from base, the user defines a shock to the rate forecast from another scenario, the base scenario. At runtime and display time, the system must first calculate the base scenario rate and then apply the shock to the rate forecast. No conversion is applied before the rate is passed to the Cash Flow Engine.

Implied Forward

The implied forward calculation assumes that the rate is a zero coupon yield. If the IRC rate format is yield-to-maturity the following process occurs:

- Translate input rates from yield-to-maturity to zero coupon yield.
- Apply implied forward calculations on zero coupon yields.
- Translate results of implied forward calculation from zero coupon yields back into yield-to-maturity.

For IRCs that are already in zero-coupon yield format, the implied forward calculations can be applied directly to the historical rates and no conversion between formats is required.

Yield Curve Twist

The user inputs the tenors representing the Short Point, Anchor Point and Long Point. For each of these tenor points, the user additionally adds the required shock amounts for each tenor. At runtime and display time, the rate changes are added to the as-of-date rates to create the future scenario. No conversion is applied before the rate is passed to the Cash Flow Engine.

Derivation of Exchange Rate Forecasts

With the Forecast Rates assumption rule, exchange rates can be forecast as a function of the interest rate scenarios. Each currency has a reference yield curve that drives the parity exchange rate relationships between this currency and other currencies. The two methods that require interest rate forecasts are forward and parity.

If the reference IRC is in a yield-to-maturity format, the system converts the rate to a zero coupon yield format prior to calculation of the exchange rate.

Scenario-based Market Value Calculations

In scenario-based processes, users select, per product and currency, an IRC as their valuation curve. Market value sensitivity processes use shocks to the as-of-date rates to drive the market value changes. Dynamic market value calculations use the forecasted curve as of the dynamic valuation date. The set of valuation curves can be determined from the Discount Methods assumption rule. The system converts all rates to zero-coupon format prior to processing.

Monte Carlo Rate Path Generation

Monte Carlo processes require annually compounded Act/Act zero-coupon yields as the input. If the input IRC format is anything other than that, a conversion process is applied.

Stochastic rates output from the Monte Carlo engine are also annually compounded Act/Act zero-coupon yields.

Rate Index Calculation from Monte Carlo Rate Paths

In the Stochastic Rate Index assumption rule users select an IRC as their valuation curve. For each currency, formulas are applied to the yields forecasted from the valuation curve for the Monte Carlo process. A formula for each additional IRC must exist in the Stochastic Rate Index assumption rule. The formulas are applied during processing when one of the additional IRCs is required for pricing new business or repricing existing business.

Funds Transfer Pricing

There are both cash flow and non-cash flow transfer pricing methods in Oracle FTP.

Non Cash Flow Methods

The non-cash flow transfer pricing methods use historical IRC data pulled directly from the historical rates tables. For these methods, the format of the IRC used as the transfer pricing yield curve is assumed to be a yield-to-maturity.

Cash Flow Methods

There are four cash flow methods:

- Weighted Average Term

- Average Life
- Duration
- Zero Coupon

Weighted Average Term: Weighted Average Term calculates the cash flows over the funding period, treating the next repricing date as a maturity date. The cash flows are discounted by the current net rate. The discounted cash flow at each payment/maturity is used as the weighting factor for the rate from the transfer pricing yield curve. The term from the origination to the cash flow date is used as the term for lookup to the transfer pricing yield curve.

For this method, the transfer pricing yield curve is assumed to be in the proper rate format. No adjustment will be made to the current net rate or the transfer pricing yield curve.

Average Life: The average life method calculates the average life by taking the cash flows over the funding period and calculating average life for the series of cash flows. The computed average life of the cash flows is used as the term for lookup to the transfer pricing yield curve.

For this method, the transfer pricing yield curve is assumed to be in the proper format. No adjustment will be made to the current net rate.

Duration: The duration method calculates the duration by taking the cash flows over the funding period and calculating duration for the series of cash flows. The current net rate is used as the discount rate. The duration of the cash flows is used as the term for lookup to the transfer

The duration of the cash flows is used as the term for lookup to the transfer pricing yield curve. For this method, the transfer pricing yield curve is assumed to be in the proper format. No adjustment will be made to the current net rate.

Zero Discount Factors: The zero discount factor method must calculate discount factors for the transfer pricing yield curve. If the transfer pricing yield curve is stored as yield-to-maturity rates, the rates must first be translated into zero-coupon yields so that the discount factor can be calculated from them. If the transfer pricing yield curve is already in zero-coupon yield format, then discount factors can be calculated directly from the rates.

Rate Conversion Algorithms

Conversion of accrual basis or compounding basis are straightforward. Rate format conversions (zero-coupon and yield-to-maturity) are more difficult.

Conversion From Yield-to-Maturity to Zero-Coupon Yield

Based on Equation 9–3, page 9-4 and Equation 9–4, page 9-5, the system first calculates the values

$c(T_i)$

$AI(T_i)$

Then the system bootstraps the yield curve using the BFGS algorithm to solve Equation 9–1, page 9-4 and Equation 9–2, page 9-4.

> **Note:** See, for instance, Press W., Flannery, Teukolsky, and Vetterling. Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, Second Edition, 1992.

Conversion From Zero-Coupon Yield to Yield-to-Maturity

Based on equations Equation 9–1, page 9-4 and Equation 9–2, page 9-4, the system first calculates the values

$c(T_i)$

$AI(T_i)$

Then, the system solves Equation 9–1, page 9-4 and Equation 9–2, page 9-4 using the Newton-Raphson algorithm.

10

Forecast Rate Calculations

This chapter describes how forecast rate assumptions are created and managed within the Forecast Rate Scenario's user interface.

Forecast assumptions for currency exchange rates and interest rates are defined within the Oracle Asset Liability Management (ALM) Forecast Rates assumption rule. The resulting rates can be calculated and viewed through the user interface. These calculations are also used during Oracle ALM deterministic processing, at which time the resulting rates can be output for auditing or reporting purposes.

This chapter covers the following topics:

- Currency Forecasting
- Interest Rate Forecasting

Currency Forecasting

In order to model the effect of currency fluctuations on income, a process must include a forecast of future exchange rates between currencies. The exchange rates forecast will affect calculation of gains/losses and consolidation to a specified reporting currency.

When a new Forecast Rates assumption rule is created, it is designated with a specific reporting currency. All exchange rates in that assumption rule are defined as exchange rates to one unit of the reporting currency.

The following forecasting options are available:

Method	Description
Flat	Exchange rates throughout the forecast remain equal to the rate in effect on the As-of-Date.

Method	Description
Structured Change	Exchange rates are based on an incremental change from the previous period.
Direct Input	The user manually inputs the exchange rate for each modeling bucket.
Parity	The exchange rate between the selected currency and the reporting currency is based on interest rate forecasts for the Reference IRC associated with each of the two currencies.
No Arbitrage	The exchange rate between the selected currency and the reporting currency depends on the interest rates in effect on the As-of-Date for the Reference IRCs of the two currencies.

This section covers calculations used for the Structured Change, Parity, and No Arbitrage currency exchange rate methods.

Structured Change

For Structured Change input, the user can incrementally increase or decrease exchange rates over specific time periods. Structured rate changes are applied to the exchange rates in effect during the previous period. Rate changes have two components:

- a change amount
- a period over which the change occurs

The minimum period over which a change occurs is one modeling bucket. If a rate change occurs over more than one modeling bucket, the rate change is apportioned across each modeling bucket using a straight-line method based on the amount of time in each bucket.

If the modeling bucket lengths are not even, each modeling bucket's length is converted to units of months. Oracle ALM employs the same method to calculate an equal-month percentage for daily modeling buckets, as described later in this chapter for the Implied Forward calculations.

Once all modeling buckets are expressed in monthly units, the apportionment of rate changes can occur.

1. Add total number of months for the modeling bucket range.
2. Divide total rate change by total number of months.

3. Apply rate change in each bucket by multiplying the monthly rate change by the number of months for that bucket.

For example, assume that the active Time Bucket rule defines each bucket as 1 month, and the following Structured Change is forecast:

Start Bucket	End Bucket	From Date	To Date	Total Rate Change
1	3	1/1/2011	03/31/2011	0.25
4	12	4/1/2011	12/31/2011	1.25

With monthly buckets, the amount of change to apply in this example is as follows:

Bucket(s)	Rate Change per bucket
1 - 3	0.25 / 3 = 0.0833
4 - 12	1.25 / 9 = 0.1389

Applying this change in each bucket results in the following forecast rates:

Bucket #	Bucket Start	Bucket End	Rate
0	N/A	12/31/2010	5.2
1	1/1/2011	1/31/2011	5.2833
2	2/1/2011	2/28/2011	5.3667
3	3/1/2011	3/31/2011	5.45
4	4/1/2011	4/30/2011	5.5889
5	5/1/2011	5/31/2011	5.7278
6	6/1/2011	6/30/2011	5.8667
7	7/1/2011	7/31/2011	6.0056

Bucket #	Bucket Start	Bucket End	Rate
8	8/1/2011	8/31/2011	6.1444
9	9/1/2011	9/30/2011	6.2833
10	10/1/2011	10/31/2011	6.4222
11	11/1/2011	11/30/2011	6.5611
12	12/1/2011	12/31/2011	6.7
13 +	1/1/2012	end of modeling horizon	6.7

Parity

The Parity exchange rate method derives the exchange rate between the selected currency and the reporting currency based on the forecasted reference interest rates for each respective currency. This enables the user to forecast different interest rates associated with the two currencies and maintain a parity relationship in the exchange rate.

The parity method can be used only if both the reporting currency and the selected currency have a Reference IRC (defined through Rate Management).

For parity conditions to hold, an investment made at currency a's interest rate should equal an investment made at currency b's interest rate for the same period of time, taking into account the exchange rate between the two currencies. Interest rates are converted to equal formats of accrual basis and compounding basis. This is achieved by converting the rates to a discount factor. (For complete details on conversion to a discount factor, see Rate Conversion, page 9-1) As a simple example, let's use annual compounding. The Parity formula would be:

$$ba^{n+t} = ba^{n*}[(1 + r_{nb}{}^{t})^{\wedge m} \,/((1 + r_{na}{}^{t})^{\wedge m}]$$

Where

Variable	Definition
ba^{n+1}	Exchange rate from currency *b* (the selected currency) to currency *a* (the reporting currency) at time n+t.
t	Modeling bucket term for modeling bucket n+1.
ba^{n}	Exchange rate from currency *b* to currency *a*, at time n.
$r_{nb}{}^{t}$	Reference interest rate in currency *b* for term t, at time n.
$r_{na}{}^{t}$	Reference interest rate in currency *a* for term t, at time n.
m	Portion of year equivalent to t.

To calculate the exchange rate in each modeling bucket, the process loops through all values of n from zero to the maximum modeling bucket minus 1. The value for t in the calculation for any one exchange rate is determined by the modeling bucket term for modeling bucket n + 1.

For example, consider the following modeling bucket configuration:

Bucket #	Term	Start Date	End Date
1	1 Month	1/1/2011	1/31/2011
2	1 Month	2/1/2011	2/28/2011
3	3 Months	3/1/2011	5/31/2011
4	6 Months	6/1/2011	11/30/2011

The process will loop from n = 0 to n = 3. Assuming the interest rates listed below, the resulting exchange rates are as follows:

n	t	ba^n	r_{nb}^t	r_{na}^t	$ba^{(n+1)}$
0	1 Month (length of Bucket 1)	2.125 (Actual exchange rate from *b* to *a* on As-of-Date 12/31/2010)	4.25 (Actual 1-month interest rate for currency *b* on 12/31/2010)	2.3 (Actual 1-month interest rate for currency *a* on 12/31/2010)	2.12841 (Forecast Exchange Rate for Bucket 1)
1	1 Month (length of Bucket 2)	2.12841 (Calculated exchange rate from *b* to *a* for first modeling bucket)	4.375 (Forecast 1-month interest rate in currency *b* for first modeling bucket)	2.425 (Forecast 1-month interest rate in currency *a* for first modeling bucket)	2.13149 (Forecast Exchange Rate for Bucket 2)
2	3 Months (length of Bucket 3)	2.13149 (Calculated exchange rate from *b* to *a* for second modeling bucket)	4.75 (Forecast 3-month interest rate in currency *b* for second modeling bucket)	2.9 (Forecast 3-month interest rate in currency *a* for second modeling bucket)	2.14109 (Forecast Exchange Rate for Bucket 3)
3	6 Months (length of Bucket 4)	2.14109 (Calculated exchange rate from *b* to *a* for third modeling bucket)	5.25 (Forecast 6-month interest rate in currency *b* for third modeling bucket)	3.275 (Forecast 6-month interest rate in currency *a* for third modeling bucket)	2.16152 (Forecast Exchange Rate for Bucket 4)

No Arbitrage

No Arbitrage (forward) exchange rate forecasting is similar to the Parity method, but it relies only on the interest rates in effect on the As-of-Date for each respective currency. Based on the relative interest rates in each country, the No Arbitrage method tells the user what the forward exchange rates must be to maintain no-arbitrage between the two currencies. Interest rates are converted to equal formats of accrual basis and compounding basis. This is achieved by converting the rates to a discount factor. (For complete details on conversion to a discount factor, see Rate Conversion, page 9-1.) As a simple example, let's use annual compounding; the basic formula for Forward exchange rates would be:

$$ba^t = ba^{0*}[(1 + r_b{}^t)^{\wedge m} / ((1 + r_a{}^t)^{\wedge m}]$$

Variable	Definition
ba^t	Exchange rate from currency *b* (the selected currency) to currency *a* (the reporting currency) at time t
ba^0	Exchange rate from currency *b* to currency *a* at As-of-Date
t	Time from As-of-Date to Start Date of bucket
$r_a{}^t$	Reference interest rate in currency *a* for term t, at As-of-Date
$r_b{}^t$	Reference interest rate in currency *b* for term t, at As-of-Date
m	Portion of year equivalent to t

Calculations then loop through all modeling buckets.

For example, consider the following modeling bucket configuration (and related variables):

Bucket #	Term	Start Date	End Date	t = Start Date - As-of-Date	m = t / 365
1	1 Month	1/1/2011	1/31/2011	1 Day	0.00274
2	1 Month	2/1/2011	2/28/2011	32 Days	0.087671
3	3 Month	3/1/2011	5/31/2011	60 Days	0.164384
4	6 Month	6/1/2011	11/30/2011	152 Days	0.416438

Further, the exchange rate on the As-of-Date (ba^0) is 3.8, and interest rates on that date are as follows:

Term	r_b^t	r_a^t
1 Day	3.625	2.1
32 Days	4.5	2.625
60 Days	4.875	2.75
152 Days	5.9	3.5

Therefore, exchange rates would be:

Bucket #	$ba^t = ba^{0*} [(1 + r_b{}^t)^{\wedge m} / ((1 + r_a{}^t)^{\wedge m}]$
1	3.8 * 1.000041 = 3.800154
2	3.8 * 1.001589 = 3.806037
3	3.8 * 1.003371 = 3.812808
4	3.8 * 1.009592 = 3.836450

Interest Rate Forecasting

The available IRC forecasting methods are as follows:

Method	Description
Flat	Interest rates stay constant throughout the forecast.
Structured Change	The user defines rate changes for each term point over specified periods of time.
Direct Input	The user manually inputs the interest rate for each modeling bucket and term.

Method	Description
Implied Forward	The interest rates will be derived from the term structure of the IRC. This method is available only for yield curves, which are IRCs that consist of multiple terms. If the selected IRC is a single point index, this option is disabled.
Change From Base	The scenario represents a delta from another scenario. This option is available only if the Forecast Rates ID has more than one scenario. Otherwise, this option is disabled.
Yield Curve Twist	The user defines the short point, anchor point and long point and related shock amounts for each.

This section explains calculations for the Structured Change, Implied Forward, Change From Base interest rate and Yield Curve Twist forecast methods.

Structured Change

The calculations for Structured Change of interest rates are similar to the calculations for Structured Change of exchange rates. Structured rate changes for each term point are applied to the interest rates in effect in the previous period. If a rate change occurs over more than one modeling bucket, the rate change is apportioned across each modeling bucket using a straight-line method based on the amount of time in each bucket.

Implied Forward

Implied forward rates are calculated by looking at today's yield curve and inferring the future rate value.

1. Retrieve Yield Curve

   The implied forward calculation starts with the current yield curve. The yield curve is retrieved from the historical rates database.

2. Calculate equal-month terms from yield curve.

   The terms of the yield curve must be translated into equal-month values.

   For daily terms, the system must calculate the portion of a month the daily value represents.

   For monthly terms, the exact unit of the term point is used.

   For yearly terms, the unit of the term point is multiplied by 12.

3. Calculate equal-month time for the modeling buckets.

The modeling buckets are also translated into equal-month time. The same process can be followed as mentioned earlier.

4. Apply formula calling the cubic spline function.

   To determine the implied forward rates, the process must step through each modeling bucket and each term on the yield curve. The process will loop as follows:

   For n = 1 to n *(modeling bucket loop)*

   For m = 1 to m *(term loop)*

   $Yield_c\ (bucket_n,\ term_m) =$

   $[cubicspline\ (bucket_n + term_m) * (bucket_n + term_m) - cubicspline\ (bucket_n) * bucket_n] / term_m$

   Next m

   Next n

   In the earlier formula, cubicspline refers to the cubic spline function. This function, currently used in the Rate Generator for Monte Carlo, takes a term as an argument and returns the smoothed yield for that term.

5. Translate continuous yields to discrete yields.

   The output of the formula mentioned earlier, produces a continuous yield, referred to as $Yield_c$. The required format for yield is a discrete yield. To translate from the continuous yield to the discrete yield ($Yield_d$), the following formula must be applied:

   $$Yield_d = e\text{^}\ Yield_c - 1$$

Change From Base

For Change From Base interest rate forecasting, the base scenario is calculated first and then the incremental change is applied to all rates.

Yield Curve Twist

For the Yield Curve Twist forecasting method, the user must select 3 term points to represent the following:

Short Point: This can be the first tenor from the yield curve, but can also be an intermediate point.

Anchor Point: The anchor point is typically the point on the yield curve where the pivot occurs. It typically does not change in a twist scenario.

Long Point: The long point can be the last tenor in the yield curve or an intermediate point.

If intermediate points are selected for either the short point or the long point, all term points less than the short point and greater than the long point will reflect the same change amount input for the short and long point.

11

Monte Carlo Analytics

This chapter describes the details of stochastic processing used within Oracle Asset Liability Management (ALM) and Funds Transfer Pricing (FTP). This includes the detailed architecture of the Monte Carlo rate generator that is used for stochastic forecasts of interest rates and calculation of market value, Value-at- Risk, and Earnings-at-Risk.

This chapter covers the following topics:

- Overview
- Architecture of the Rate Generator
- Arbitrage Term Structure Model
- No-Arbitrage Term Structure Models
- Yield Curve Smoothing
- Calibration
- Random Number Generation
- Rate Conversion
- Output From Monte Carlo
- Value-at-Risk
- Varying the At-Risk Period
- Earnings-at-Risk
- Recommended Configuration
- Term Structure Models
- Term Structure Parameters Format
- Choosing a Smoothing Method
- Defining a Rate Index Formula
- References

Overview

Monte Carlo is a valuation technique that provides a direct and robust simulation of interest rate paths and provides for market value and Value-at-Risk (VaR) calculations. Monte Carlo becomes a necessary tool in financial markets for solving problems when other methods are unavailable for problems in high dimensions, simulation, and optimization. For Oracle Asset Liability Management (ALM) and Funds Transfer Pricing (FTP) processes, Monte Carlo is a particularly useful tool for valuing instruments with uncertain cash flows. Examples of such instruments include:

- Callable debt
- Capped loans
- Prepayable mortgages

The Monte Carlo rate generator is a calculation engine that forecasts future rate changes within a stochastic process. Central to the understanding of the rate generator is the acknowledgment that a rate forecast will always be imperfect. This means that future rates will not fully match the prognosis given by the model. However, it is possible to quantify the uncertainty of future interest rates or, in other terms, to forecast a probability distribution of interest rates.

Economic theory tells us that there are two types of forecasts:

- forecast of the real interest rates, based on a "subjective" assessment of the economy.
- forecast of the risk-neutral interest rates, based on the original yield curve and the no-arbitrage condition.

The two types of prognosis will not necessarily match. A bank would typically use the first type of rates to model future income, because it wants its income forecast to be as close as possible to the actual future income. A bank would typically use the second type of rates to calculate present and future market value because market value depends not only on the rates but also on the degree of risk-aversion of each agent in the economy. "Risk-neutral rates" are a theoretical construction that enables us to calculate rates as if nobody were risk-averse.

Many types of analysis in interest rate management require computing the expected value of a function of the interest rate. One example is to calculate the probability that portfolio loss is within a certain range. The probability of such an event is nothing but the expected value of the indicator function of this occurrence, which is worth one if the event is true and zero otherwise. Another example is to compute the market value of a derivative instrument.

In mathematical terms, the market value of a security that pays a cash flow at time T is equal to the expected value of the product of the stochastic discount factor at time T and the cash flow, that is:

Market Value = E [Discount Factor * Cash Flow]

where the stochastic discount factor is equal to the present value (along a rate scenario) of one dollar received at time T. It is therefore a function of the rate.

The goal of term structure models is to forecast probability distributions of interest rates under which the expected value is defined.

Most term structure models used in practice, and all term structure models available in the system, are single factor models of the short term interest rate. Short Rate modeling prevails because the problem of correlating multiple factors is greatly simplified. With single factor models the value in the future depends only on the value at the current time, and not on any previous data. This property is referred to as a Markov process.

Monte Carlo is the most popular numerical technique to compute an expected value, in our case market value and Value-at-Risk. The methodology consists of generating rate scenarios using random numbers, computing a function of the rates for each scenario, and then averaging it. Market value is the average across all scenarios of the sum of all cash flows discounted by the (scenario-specific) rate. Value-at-Risk is the maximum loss in value over a specific horizon and confidence level.

Monte Carlo simulation works forward from the beginning to the end of the life of an instrument and can accommodate complex payoffs, for instance, path-dependent cash flows. The other numerical methods, such as lattice and finite difference, cannot handle the valuation of these path-dependent securities. The drawback of Monte Carlo is its slow convergence compared to other methods. We address this problem by implementing better random sequences of random numbers, namely Low-Discrepancy Sequences. Monte Carlo has better performance than other techniques, however, when the dimension of a problem is large.

Architecture of the Rate Generator

The Rate Generator takes as input the information from the Rate Management tables (which include term structure parameters and smoothing technique, and risk-free yield curve yields), Product Characteristic assumptions (OAS), ALM Stochastic Process (risk-free yield curve, number of rate scenarios, risk period at which to compute VaR), Stochastic Rate Index rules (formulas for rate indices), and generates monthly rates and stochastic discount factors for each scenario and monthly bucket.

We take the benchmark securities to be zero-coupon bonds whose yields are stored in Rate Management and identified as the risk-free yield curve.

The Monte Carlo Rate Generator also calculates future rates (for a maturity other than one month) for each scenario and beginning at any month within the modeling horizon. These rates are used by the cash flow engine for repricing events and as market rates within the prepayment function.

The process flow of the Rate Generator depends on the type of a term structure model. However, all processes share the same building blocks.

Computation of Rates and Stochastic Discount Factors

At the beginning of an ALM Process, the Rate Generator calculates and stores monthly rates for 360 months and for all scenarios.

Upon request by the Cash Flow engine, the Rate Generator will calculate the rate, that is, the annually compounded yield at any time T for a discount bond maturing at time T, as follows:

1. Compute the future bond price using a closed form solution (a discretized version of Hull and White solution.)

2. Convert discount bond prices into discrete yields.

We limit the simulation horizon to 30 years from the As-of-Date. After that period, rates are set equal to the 30-year rate. Also, the closed form formula is limited to a 30-year term. Rates with a longer term are set equal to the 30-year term rate.

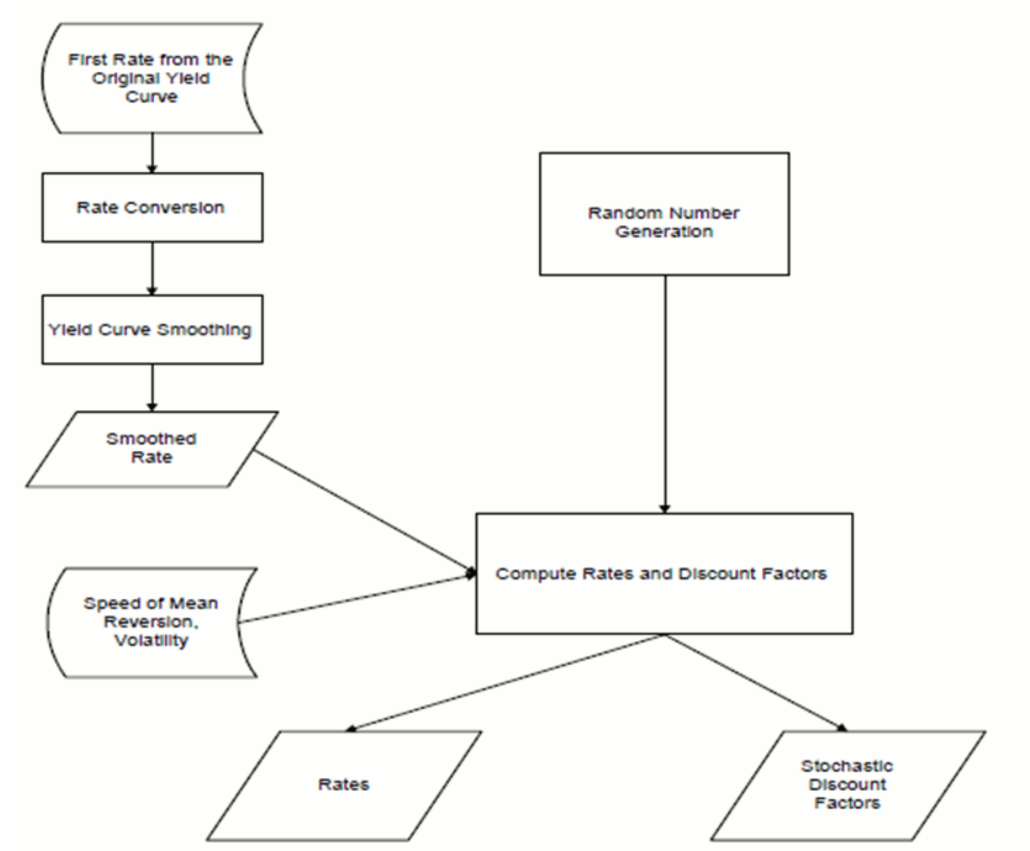
The cash flow engine may need rates from another IRC (Index) than the risk-free IRC. In this case, the value of the Interest Rate Code (IRC) with maturity T will be a function of the value of the risk-free IRC.

The Rate Generator pre-computes the stochastic discount factor for every scenario, for the first day of every monthly bucket. The computation can be selected with or without an Option Adjusted Spread (OAS) added. The OAS depends on the Product Dimension member, and is expressed in terms of discrete annual compounding rate. For internal code optimization reasons, we limit the maximum OAS value to 5%.

Arbitrage Term Structure Model

Arbitrage models (Merton and Vasicek) use the first rate from today's yield curve (smoothed and converted to continuous compounding, as described later in the section). It feeds directly the Rate Generator along with the term structure parameters to produce monthly rates, stochastic discount factors for each scenario and month, and discrete rates for any maturity.

The following diagram shows the process flow for Vasicek (or Merton) models in the Rate Generator:



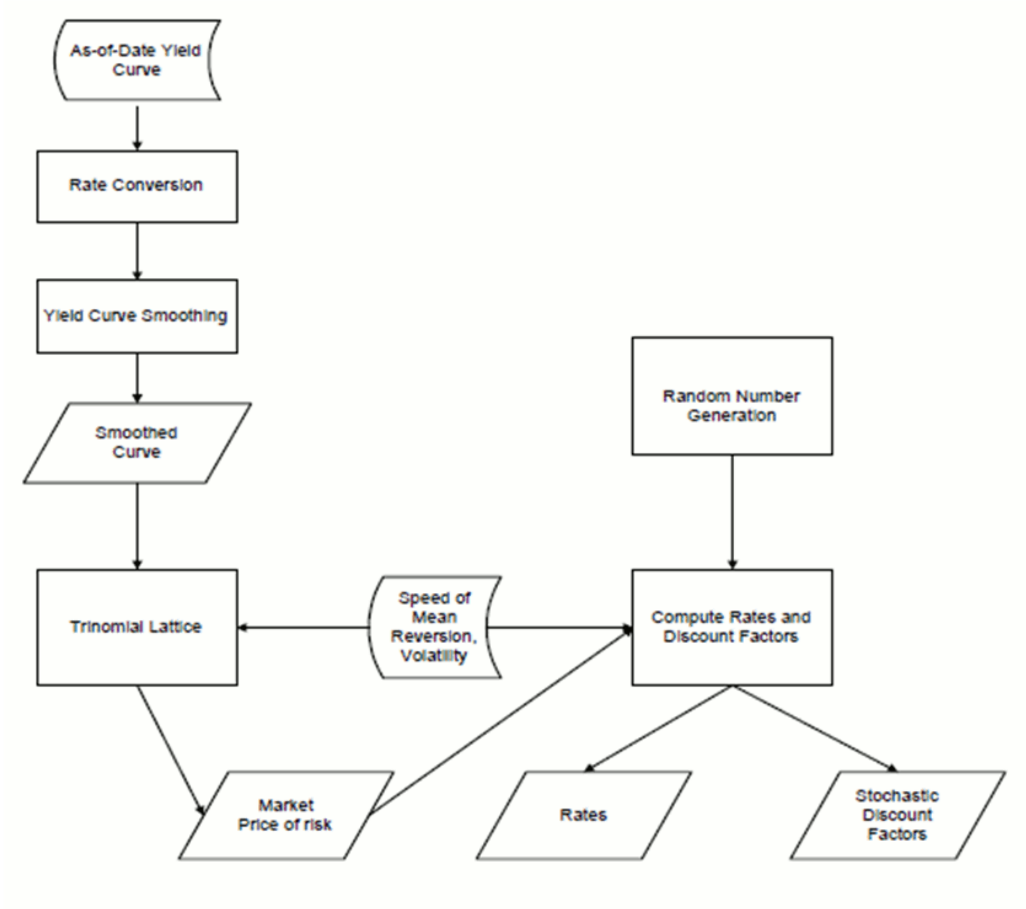
The disadvantage of these models is that they do not automatically fit today's term structure. The parameters can be chosen so that arbitrage models provide a close fit to many of the real yield curves, but the fit is usually not exact and often gives significant errors.

To achieve no-arbitrage conditions, the model itself can be used to calibrate the parameters of the model. Given the prices of benchmark securities, the model finds the rate probabilities such that, when they are used as input to the pricing tool, the output will be as close as possible to the original prices.

No-Arbitrage Term Structure Models

The no-arbitrage models (Ho and Lee and Extended Vasicek) are designed in order to fit the current term structure of interest rates exactly. This is achieved when the price of zero coupon bond computed through the Monte Carlo process exactly matches today's observable market rates, as defined by the risk free curve selection. The market price of risk is a function of time that takes care about this fit.

To adhere to this condition, the process flow for no-arbitrage models is more complicated. A trinomial lattice is used to compute the market price of risk to calibrate the term structure. The full original As-of-Date yield curve is used, smoothed and converted to continuous compounding, to be fed into the lattice.



Yield Curve Smoothing

Today's yield curve from an OFSAA Rate Management IRC might not have enough terms for a term structure model to give the best rates forecast. In particular, the trinomial lattice needs the value of the yield for every month. To do this, two smoothing techniques have been introduced into the Rate Generator: Cubic Spline and Linear Interpolation.

Calibration

Hull and White have shown how trinomial trees (lattice) can be used to value interest rate derivatives. The goal of the trinomial lattice is to compute the market price of risk for all buckets for the Ho and Lee or Extended Vasicek (no-arbitrage) term structure models.

Hull and White lattice is a popular methodology to calibrate a term structure model. The lattice is constructed for up to 360 monthly buckets, from bucket zero to either (the last maturity on the IRC + 15 years) or 30 years, whichever is shorter.

Random Number Generation

To model a discrete-time stochastic process using Monte Carlo method we need to use some random variables. Numbers generated by computers are not truly random. Standard random number generators produce uniformly distributed pseudo-random numbers that are made to have the statistical properties of truly random numbers.

Non-uniform variables are sampled through transformation of uniform variables. There are several methods developed for this purpose, both general (to produce random variables of a given density) and special (to directly generate normal random variables): transformation, acceptance-rejection, Box-Muller, and so on. We use the so-called "polar algorithm" technique.

You have a choice to use Low Discrepancy Sequences (LDS), also called quasi-random numbers, as a substitute to pseudo-random numbers. One more important feature of LDS is to reach the consistency across all computers in generating rate paths (pseudo-random numbers generated by different machines are different).

Low Discrepancy Sequences

Low Discrepancy Sequences, also called quasi-random numbers, are a recent substitute to the traditional random numbers used by all compilers (usually called pseudo-random numbers). They were introduced in the 50's and 60's by mathematicians like Halton, Sobol, and Niederreiter. The theory and implementation of low discrepancy sequences have enjoyed tremendous progress in the past 10 years, and they are now becoming increasingly popular in the area of financial mathematics.

The main objective of low discrepancy sequences is to reduce the error of Monte Carlo methods. Because error in Monte Carlo methods decreases with the number of scenarios, we can also state that Monte Carlo with LDS needs fewer scenarios than traditional Monte Carlo to obtain the same result. Therefore, Monte Carlo with LDS should be faster than traditional Monte Carlo.

Rate Conversion

There are two major procedures applied by the Rate Generator to the original risk-free rates from the Rate Management IRC: timescale conversion and compounding basis conversion.

Timescale Conversion

We introduced two timescales, the normal timescale and the equal-month timescale, in order to satisfy three requirements:

- Monthly buckets have to be an integer number of days in length, because the cash flow engine works on a daily timescale.

- It is better for performance to generate rates with an equal-month timescale.
- We cannot set the bucket length to be 30 days, because buckets will start 5 days earlier each year, and this conflicts with reporting requirements.

The equal-month timescale is used only internally in the Rate Generator. It assumes that each month is constant and is equal to 1/12 of a year. The normal timescale counts the actual number of days, that is, each monthly bucket has a different length. In other words, the normal timescale assumes an Actual/Actual day count basis, whereas the equal-month timescale assumes 30/360 count basis. The convention is the regular Oracle ALM convention for a month: if bucket zero starts on day n then all next buckets start on day n except when this day does not exist (February 30 for instance), in which case it reverts to the last existing day of the month (for example, February 28).

There is a one-to-one relationship between the timescales.

Example

Let us suppose that the As-of-Date is January 15, 2011. By definition, every bucket will then start at the 15th of that corresponding month.

Calendar Time	Time on normal timescale	Bucket Number	Length of the Bucket	Time in equal month timescale	Discrete rate on normal timescale	Converted rate on equal month timescale
1/15/2011	0	0	31/365	0	-	-
1/20/2011	5/365	0	31/365	5/(12*31)	0.05	0.049726
2/14/2011	30/365	0	31/365	30/(12*31)	0.05	0.049726
2/15/2011	31/365	1	28/365	1/12	0.05	0.049726
3/14/2011	58/365	1	28/365	1/12+27/(12*28)	0.05	0.047364

In this example, the discrete yield (quoted Actual/Actual) is constant. However, the 2-month smoothed yield is lower than the 1 month smoothed yield because the timescale transformation overestimates the length of the second month.

Compounding Basis Conversion

The Monte Carlo process requires annually compounded Actual/Actual zero-coupon yields as the input. If the input IRC format is different from that, a rate conversion process is applied (For more information, see Rate Conversion, page 9-1).

In addition, the Rate Generator converts rates from the yield curves (which are discretely compounded, on the normal timescale) into continuously compounded rates on the equal-month timescale. This conversion has to be done for all the points in today's yield curve.

The Rate Generator always uses the latest yield curve from the Rate Management IRC and assumes that the same yield curve prevails at the As-of-Date. Monte Carlo simulation always starts from the As-of-Date.

Output From Monte Carlo

The amount of data to be output in a Monte Carlo processing run depends on the type of processing and the requirements of the users. A stochastic process using Monte Carlo provides the following results:

- Market value at the dimension member level
- Value-at-Risk at the dimension member level
- Value-at-Risk at the portfolio level

The smallest set of the output will result from the market value processing run. Market value is the average across all scenarios of the sum of all cash flows discounted by the (scenario-specific) rate.

The output will be a single value - the market value per dimension member - stored in the FSI_O_STOCH_MKT_VAL table. The table has a simple structure:

LEAF_NODE	MARKET VALUE	CURRENT BALANCE	NUMBER OF RATE PATHS
Fixed Mortgages	55000.00	54000.00	2000
Deposits	10000.00	10100.00	2000

Value-at-Risk (VaR) is the maximum loss in value over a specific horizon (risk period) and confidence level. The horizon is specified by the user. Unlike many other products, Oracle ALM outputs VaR for any confidence level, that is, it outputs the full probability distribution of loss for each Product COA member. The output is written to the FSI_O_STOCH_VAR table:

Probability	Fixed Mortgages VaR	Deposits VaR
10%	-200	-100

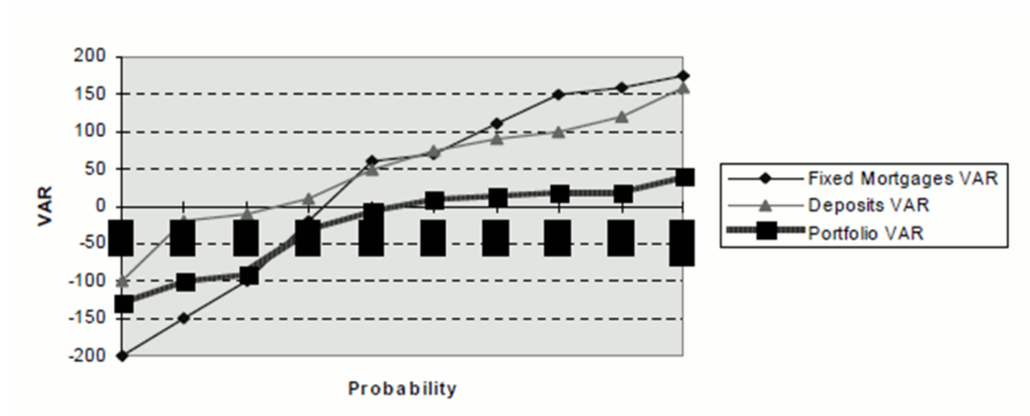
Probability	Fixed Mortgages VaR	Deposits VaR
20%	-150	-20
30%	-100	-10
40%	-20	10
50%	60	50
60%	70	75
70%	110	90
80%	150	100
90%	160	120
100%	175	160

The system will also output VaR at the portfolio level (into the FSI_O_STOCH_TOT_VAR table). In this example we assume that our portfolio has only deposits and fixed mortgages.

Probability	Portfolio
10%	-130
20%	-100
30%	-90
40%	-30
50%	-5
60%	10
70%	15
80%	20

Probability	Portfolio
90%	20
100%	40

As you can see from the following graph, the total VaR does not necessarily equal the sum of the VaR for each portfolio because of the correlation between mortgages and deposits.



We selected Monte Carlo method to implement VaR over other methods such as J.P. Morgan's Riskmetrics approach because of the following advantages:

- Better accuracy in analyzing nonlinear assets such as options.
- More flexibility to model the distribution of economic factors versus normal assumption in Riskmetrics.
- No need to decompose securities into "risk factors"; a complicated process for fixed income.

Oracle ALM also provides auditing options. Monthly rates correspond to yields of a risk-free zero-coupon bond with maturity of one month. These rates are used for all other calculations in the stochastic Rate Generator. They are output into the FSI_INTEREST_RATES_AUDIT table, where the values are displayed for each scenario and every time step, that is, up to 2,000 scenarios times 360 time steps. Because of the amount of data being written, this process can be very time consuming.

Cash flows are also available for auditing purposes, if the Detail Cash Flow option is selected in the Stochastic ALM Process. For more details on this process, see Detail Cash Flow Audit Options, page 4-1.

The stochastic discount factors are also output as an auditing feature. This is extremely valuable for testing purposes. The user can check this output any time the "Detail Cash Flow" output option is on. The table used is called FSI_O_PROCESS_CASH_FLOWS, Financial Element 490.

VaR Output Including Liabilities

This section explains how to interpret the meaning of the Value-at-Risk (VaR) output from Oracle ALM when the balance sheet includes liabilities. We will specifically explain the difference between the output on the product level (as stored in the FSI_O_STOCH_VAR table) from output on the balance sheet level (as stored in the FSI_O_STOCH_TOT_VAR table). For simplicity, let us consider a simple balance sheet with only one liability instrument and five rate scenarios.

An Example: Product Versus Balance Sheet Level Results

Let us assume that for the 1 liability instrument, Oracle ALM calculates a market value of $100,000 and the following accrued dynamic present values for each scenario:

Scenario	Accrued Dynamic Present Value
1	126,408
2	124,773
3	91,192
4	82,043
5	76,851

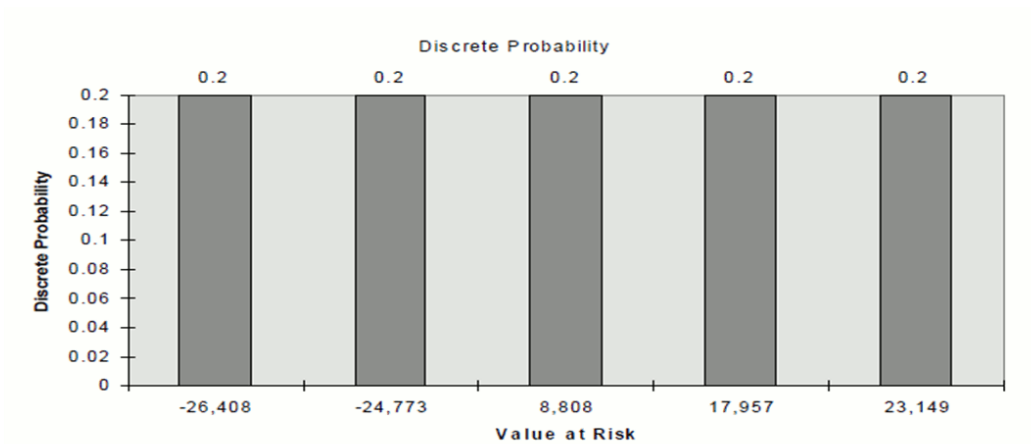
Though the actual value to the bank is -$100,000, Oracle ALM by convention reports a positive market value for all instruments. Therefore, it is important to remember that a positive market value for a liability is in effect a loss. The same convention applies to VaR. Though VaR is defined as the "maximum loss (to the bank)", it must be substituted with "maximum gain (to the bank)" when interpreting the VaR output for an instrument that is a liability.

Assume Oracle ALM outputs the first three columns of the following table to the FSI_O_STOCH_VAR table for the 1 liability product. Recall that FSI_O_STOCH_VAR contains product level output. When interpreting the VaR output for a liability on a product output level, you must remember that VaR is the "maximum gain (to the bank)".

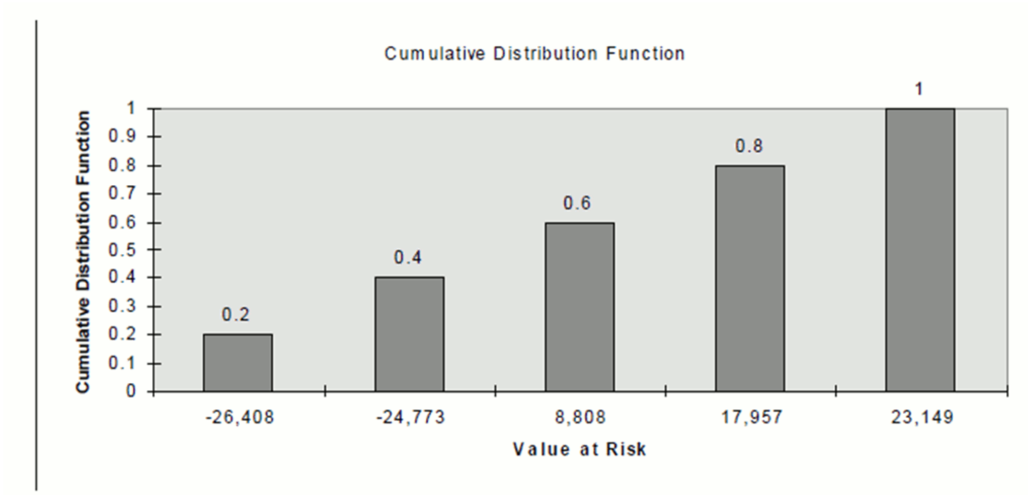
Scenario	Probability	VaR	Interpretation
1	0.2	100,000 -126,408 = -26,408	With a prob. of 20%, gain ≤ -26,408
2	0.4	100,000 -124,773 = -24,773	With a prob. of 40%, gain ≤ -24,773
3	0.6	100,000 -91,192 = 8,808	With a prob. of 60%, gain ≤ 8,808
4	0.8	100,000 -82,043 = 17,957	With a prob. of 80%, gain ≤ 17,957
5	1.0	100,000 -76,851 = 23,149	With a prob. of 100%, gain ≤ 23,149

The discrete probability of a gain of exactly -$26,408, -$24,733, $8,808, $17,957, $23,149 is 20% for a 5 scenarios simulation.

Note: The sum of all probabilities add up to a total of 100% probability.



In Value at Risk interpretation, we need to convert the discrete probability to a cumulative distribution function. For example, we know that there is a chance of a gain of -$26,408 with a probability of 20% and a gain of -$24,773 with a probability of 20%. This means that there is a 40% probability of a gain **less than or equal** to -$24,773. This comes from adding 20% discrete probability from gain of $-26,408 and 20% discrete probability from a gain of -$24,773.



Now let's assume Oracle ALM outputs the first three columns of the following table to FSI_O_STOCH_TOT_VAR table for the 1 liability product. Recall that FSI_O_STOCH_TOT_VAR contains balance sheet level output. At the balance sheet level, we aggregate gain/loss information for different products that can be either assets or liabilities. Oracle ALM by convention reports the aggregated VaR number as a " maximum loss"; therefore, the sign of VaR for all liabilities must be reversed.

Scenario	Probability	VaR	Interpretation
5	0.2	-23,149	With a prob. of 20%, loss ≤ -23,149
4	0.4	-17,957	With a prob. of 40%, loss ≤ -17,957
3	0.6	-8,808	With a prob. of 60%, loss ≤ -8,808
2	0.8	24,773	With a prob. of 80%, loss ≤ 24,773
1	1.0	26,408	With a prob. of 100%, loss ≤ 26,408

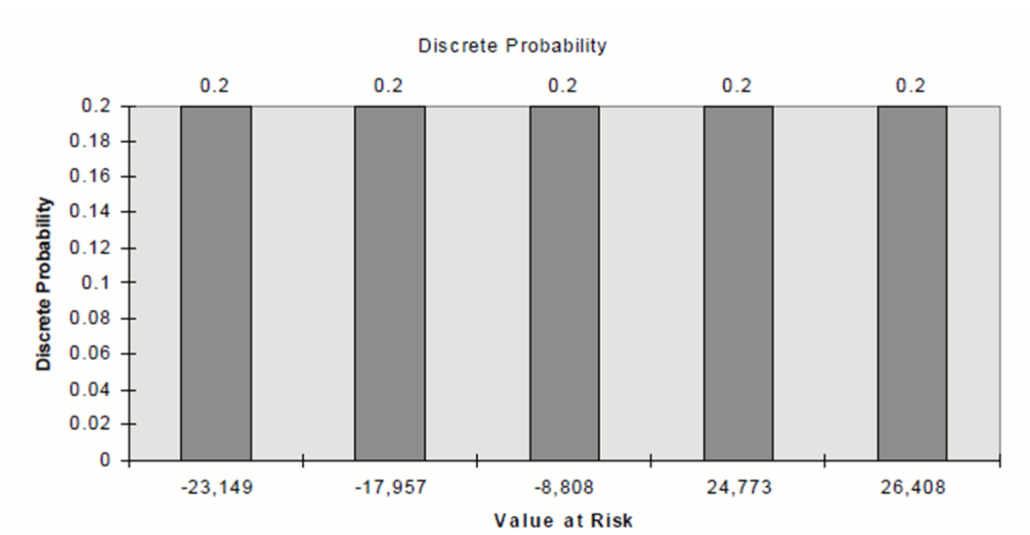
Validating Interpretation

Let us analyze the most beneficial scenario (to the bank) to prove that our interpretation is correct. In scenario 5, the "gain = 23,149" has a probability of 20% due to the fact that it is a discrete event. We also have the following events and probabilities:

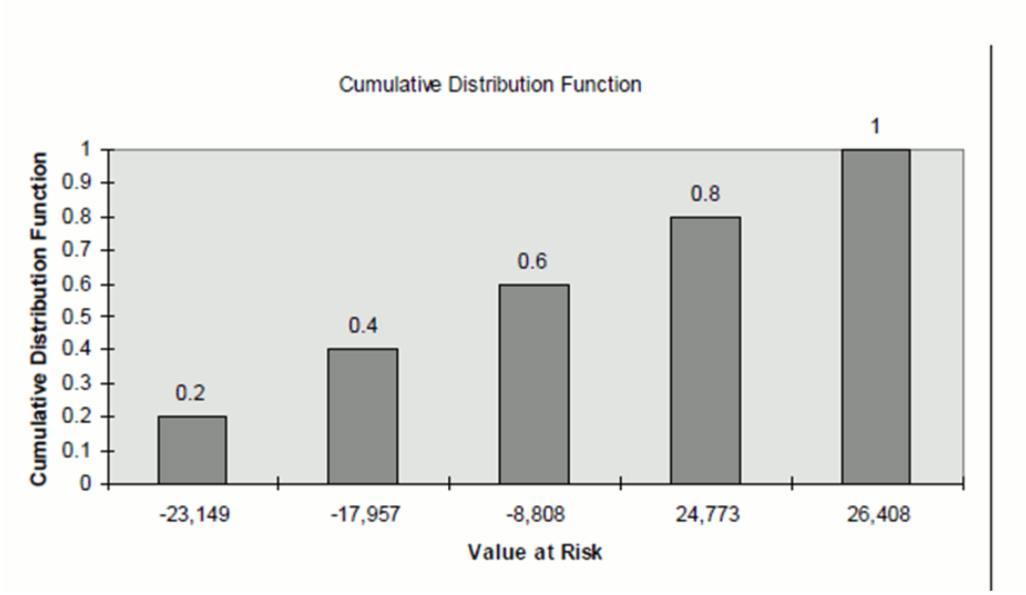
Event	Probability
gain < 23,149	80%
gain = 23,149	20%
gain > 23,149	0

When we apply the signage to liability, the discrete probability is undisturbed for each event.

> **Note:** Each event still has the discrete probability of 20%



We see here that there is a 20% probability of a maximum loss of -$23,149. There is a 40% probability of a maximum loss of -$23,149 (20% discrete probability from -$23,149 plus 20% discrete probability from -$17,957).



The event "gain ≤ 23,149" corresponds to the union of these two events:

"gain < 23,149"

"gain = 23,149"

Because these two events are disjoint - making each event's probability discrete – the probability that "gain ≤ 23,149" is the sum of their respective probabilities. It is equal to 80%+20%=100% as reported in the FSI_O_STOCH_VAR table.

The event "loss ≤ -23,149" on the other hand corresponds to the union of these two events:

"loss = - 23,149", that is, "gain = 23,149"

"loss < - 23,149", that is, "gain > 23,149"

Because these two events are also disjoint - making each event's probability discreet - the probability that "loss ≤ -23,149" is the sum of their respective probabilities. It is equal to 20%+0%=20% as reported in the FSI_O_STOCH_TOT_VAR table.

Value-at-Risk

In our current implementation, Value-at-Risk is the worst loss of present value over an At-Risk period (time horizon), given a confidence level. The maximal At-Risk period is 10 years.

Oracle ALM outputs Value-at-Risk for as many confidence levels as there are scenarios in Monte Carlo simulation. In other terms, it outputs the full present value loss probability distribution and its inverse, which are a much richer statistics than a single VaR number.

The following section describes our current approach to estimate Value-at-Risk.

Present Value-at-Risk

The approach is as follows:

1. For each scenario, compute the accrued dynamic present value by dividing the (scenario-specific) present value by the stochastic discount factor; we do not take new business into account.

2. For each scenario, compute VaR as market value minus accrued dynamic present value.

3. Sort VaR in ascending order and output it along with its normalized ranking (that is, the ranking divided by the total number of scenarios).

The normalized ranking is an unbiased estimator of the probability that loss of value is less than VaR, that is, each couple of values (normalized ranking, VaR) is a point on the loss probability distribution curve.

Illustration of the Approach

We have only 10 scenarios. Today's market value is $80.

Scenario Number	Stochastic Discount Factor	Present Value	Accrued Dynamic Present Value	VaR
1	0.99	81.6	82.4	-2.4
2	0.98	83.1	84.8	-4.8
3	0.97	81.5	84	-4
4	0.965	80.1	83	-3
5	0.95	79.9	84.1	-4.1
6	0.95	79	83.2	-3.2
7	0.949	79.2	83.5	-3.5
8	0.948	78.3	82.6	-2.6
9	0.947	75.1	79.3	0.7
10	0.946	70.1	74.1	5.9

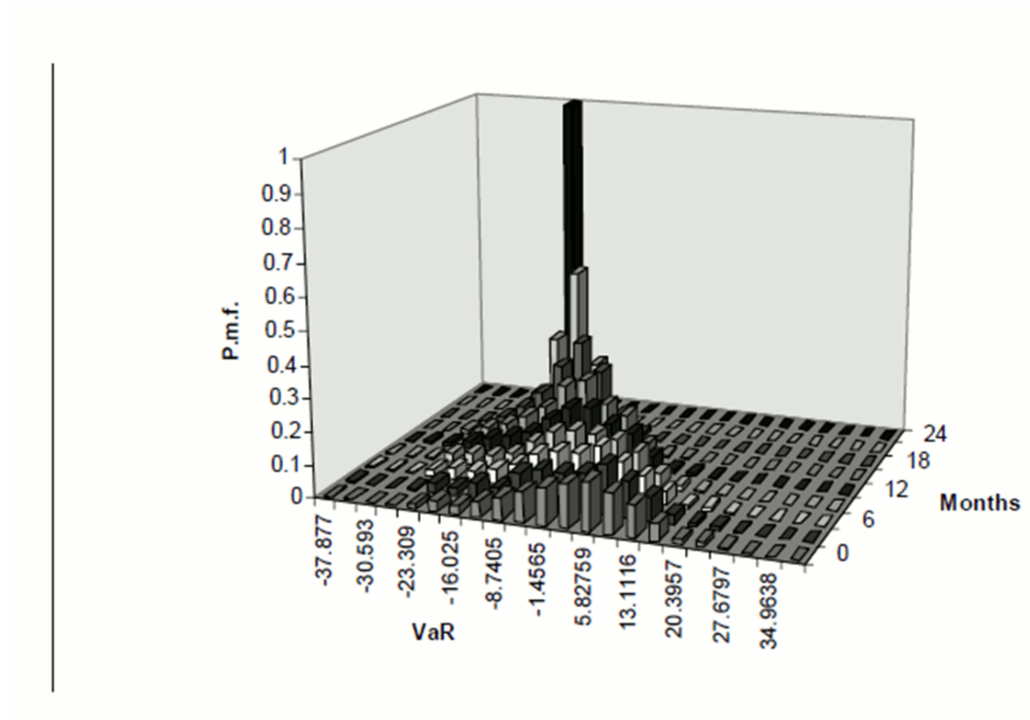
After sorting we have:

Scenario Number	Probability	VaR
2	0.1	-4.8
5	0.2	-4.1
3	0.3	-4
7	0.4	-3.5
6	0.5	-3.2
4	0.6	-3
8	0.7	-2.6
1	0.8	-2.4
9	0.9	0.7
10	1	5.9

Varying the At-Risk Period

This example tries to give an intuition behind the numbers by varying the At-Risk period for a typical case. The instrument we analyze is a two-year discount bond with a principal of $100. Market value is $78.

To ease comprehension of the graph, we display the probability mass function, and not the cumulative probability distribution.



This graph reflects two important features of a fixed rate instrument:

1. On an average, VaR decreases with time because dynamic present value increases with time.

2. The dispersion of VaR goes to zero when approaching maturity.

Earnings-at-Risk

Earnings-at-Risk is a methodology for income sensitivity that combines the Monte Carlo Rate Generator with the cash flow engine to produce statistical information about forecasted income. Earnings-at-Risk functionality uses the stochastic processing methodology to provide users with a probabilistic view of forecasted earnings for individual products and the entire balance sheet. A probability distribution of earnings enables users to view expected income as well as the potential loss of income in the future due to interest rate fluctuations. With this information, users can efficiently determine what is likely to happen as well as identify the scenarios that may provide the greatest risks to the institution.

An Earnings-at-Risk process reads records from the following sources:

- Instrument data

- Transaction Strategy

- Forecast Balance

Cash flows are calculated for each record, for every rate path generated by the Rate Generator. Earnings results may include the following financial data, depending on the account type of the product:

- Net interest accrual
- Deferred runoff
- Non-interest income
- Non-interest expense
- Dividends
- Taxes

The following four sets of output data are available after an Earnings-at-Risk processing run, for each modeling bucket as defined in the active Time Bucket definition:

- Average income over all rate paths for each product
- Average income over all rate paths for the entire bank
- Income in each rate path for each product
- Income in each rate path for the entire bank

From the preceding list, the average income data sets are output for every Earnings-at-Risk process. The other two output sets are optional and may be selected as part of the process definition.

The output is stored in the following tables:

EAR_LEAF_DTL_xxx - (optional) Earnings are aggregated by product and stored for each rate scenario.

> **Note:** Earnings are equivalent to net interest accrual in a scenario-based processing.

EAR_LEAF_AVG_xxx - Average earnings are calculated for each product by taking the simple average across all rate paths for each modeling bucket.

EAR_TOTAL_DTL_xxx- (optional) Both net interest income and net income values are calculated, aggregated across products and stored for every rate scenario per modeling bucket. Net interest income is calculated as interest income less interest expense. Net income is calculated as net interest income and non-interest income combined, less non-interest expense, less taxes, if applicable.

EAR_TOTAL_AVG_xxx - Net interest income and net income values as described earlier are averaged across all rate paths for each modeling bucket.

In addition, the initial balances and rates are output to FSI_O_RESULT_MASTER based on the instrument records being processed. This records information on each product such as the par balance, deferred balance, weighted current rates and weighted average remaining maturity.

As with any stochastic process, users have the option to output audit results for cash flows and rates (For more information on Audit in ALM Processing, see *Oracle Financial Services Asset Liability Management (OFSALM) User Guide.*)

Recommended Configuration

For maximum precision, use the following settings in a Stochastic ALM Process rule:

- Extended Vasicek term structure model
- Cubic Spline smoothing technique
- Low Discrepancy Sequences
- 2,000 rate scenarios

> **Note:** This is the recommended configuration for precision. Because 2000 scenarios may require a significant amount of computing time, the user may prefer to lose some precision by selecting fewer scenarios.

The following section attempts to explain some criteria and issues related to choosing the optimal configuration for stochastic processing.

Term Structure Models

Select the Extended Vasicek model for the computation of market value and Value-at-Risk.

You have a choice of four term structure models:

1. Merton (requires only the parameter volatility to be specified).
2. Vasicek (requires all three parameters to be specified: mean reversion speed, volatility and long run rate).
3. Ho and Lee (requires only the parameter volatility to be specified).
4. Extended Vasicek (requires mean reversion speed and volatility parameters to be

specified).

Merton Model

One simple assumption about interest rates is that they follow a simple random walk with a zero drift. In stochastic process terms, we would write the change in r as:

$$dr = \sigma\, dZ$$

The change in the short rate of interest r equals a constant sigma times a random shock term where Z represents a standard Wiener process with mean zero and standard deviation of 1.

This model has the following virtues and liabilities:

- It is a simple analytical formula.
- Zero coupon bond prices are a quadratic function of time to maturity.
- Yields turn negative (and zero coupon bond prices rise above one) beyond a certain point.
- If interest rate volatility is zero, zero coupon bond yields are constant for all maturities and equal to r.

The Merton model gives us very important insights into the process of deriving a term structure model. It's simple formulas make it a useful expository tool, but the negative yields that result from the formula are a major concern.

Ho and Lee Model

Ho and Lee [7] , page 11-37 extended the Merton model to fit a given initial yield curve perfectly in a discrete time framework. In the Ho and Lee model case, we assume that the short rate of interest is again the single stochastic factor driving movements in the yield curve. Instead of assuming that the short rate r is a random walk, however, we assume that it has a time dependent drift term $\theta(t)$:

$$dr = \theta(t)dt + \sigma\, dZ$$

As before, Z represents a standard Wiener process with mean zero and standard deviation of 1.

By applying the no-arbitrage condition, the function q(t) is chosen such that the theoretical zero coupon yield to maturity and the actual zero coupon yield are exactly the same. The function q(t) is the plug that makes the model fit and corrects for model error, which would otherwise cause the model to give implausible results. Because any functional form for yields can be adapted to fit a yield curve precisely, it is critical, in examining any model for plausibility, to minimize the impact of this extension term. The reason for this is that the extension term itself contains no economic content.

In the case of the Ho and Lee model, the underlying model would otherwise cause interest rates to sink to negative infinity, just as in the Merton model. The extension term's magnitude, therefore, must offset the negative interest zero coupon bond yields that would otherwise be predicted by the model. As maturities get infinitely long, the magnitude of the extension term will become infinite in size. This is a significant cause for concern, even in the extended form of the model.

Vasicek Model

Both the Merton model and its extended counterpart the Ho and Lee model are based on an assumption about random interest rate movements that imply that, for any positive interest rate volatility, zero coupon bond yields will be negative at every single instant in time, for long maturities beyond a critical maturity t. The extended version of the Merton model, the Ho and Lee model, offsets the negative yields with an extension factor that must grow larger and larger as maturities lengthen. Vasicek [19], page 11-38 proposed a model that avoids the certainty of negative yields and eliminates the need for a potentially infinitely large extension factor. Vasicek accomplishes this by assuming that the short rate r has a constant volatility sigma like the models as mentioned earlier, with an important twist: the short rate exhibits mean reversion:

$$dr=\alpha(b-r)dt+\sigma\, dZ$$

where:

r is the instantaneous short rate of interest

α is the speed of mean reversion

b is the long run expected value for rate

σ is the instantaneous standard deviation of r

Z is the standard Wiener process with mean zero and standard deviation of 1. The stochastic process used by Vasicek is known as the Ornstein-Uhlenbeck process. This process enables us to calculate the expected value and variance of the short rate at any time in the future s from the perspective of current time **t.**

Because **r(s)** is normally distributed, there is a positive probability that **r(s)** can be negative. As pointed out by Black [1], page 11-37, this is inconsistent with a no-arbitrage economy in the special sense that consumers hold an option to hold cash instead of investing at negative interest rates. The magnitude of this theoretical problem with the Vasicek model depends on the level of interest rates and the parameters chosen (In general, it should be a minor consideration for most applications). The same objection applies to the Merton and Ho and Lee models and a wide range of other models that assume a constant volatility of interest rates, regardless of the level of short term interest rates. Very low interest rates in Japan in early 1996, with short rates well under 0.5%, did lead to high probabilities of negative rates using both the Vasicek and Extended Vasicek models, when sigma was set to match observable prices of caps and floors. Although the price of a floor with a strike price of zero was positive during this period (indicating that the market perceived a real probability of negative rates), the best fitting values of sigma for all caps and floors prices indicated a probability of negative rates that was unrealistically large.

> **Note:** Lehman Brothers was quoting a floor on six month yen LIBOR with a three year maturity and a strike price of zero at 1 basis point bid, 3 basis points offered during the fall, 1995.

For most economies, the Vasicek and Extended Vasicek models are very robust with wide-ranging benefits from practical use.

Extended Vasicek (Hull and White) Model

Hull and White bridged the gap between the observable yield curve and the theoretical yield curve implied by the Vasicek model by extending or stretching the theoretical yield curve to fit the actual market data. A theoretical yield curve that is identical to observable market data is absolutely essential in practical application. A model that does not fit actual data will propagate errors resulting from this lack of fit into hedge ratio calculations and valuation estimates for more complex securities. No sophisticated user would be willing to place large bets on the valuation of a bond option by a model that cannot fit observable bond prices.

Hull and White apply the identical logic as described in the previous section, but they allow the market price of risk term to drift over time, instead of assuming it is constant as in the Vasicek model. This could be written as:

$$dr = (\theta\ (t) - \alpha r)\ dt + \ \sigma dZ$$

where:

r is the instantaneous short rate of interest

α is the speed of mean reversion

σ is the instantaneous standard deviation of r

$\theta\ (t)$ is the market price of risk for time t

As noted earlier, the Extended Vasicek (Hull and White) model is currently the most popular term structure model. It has a clear economic meaning, and is computationally very robust. Because of its popularity, numerous studies have documented and continue to document what parameters (speed of mean reversion and volatility) should be used. Another advantage of the Extended Vasicek model is that bond prices have an easy closed form formula. This closed form formula leads to very fast computation of rates for any term inside Monte Carlo simulation.

Term Structure Parameters Format

As mentioned in the preceding sections, speed of mean reversion, volatility and long run rate are the parameters the user will have to specify for the chosen model. Speed of Mean Reversion represents the long-run drift factor (1/mean reversion = interest rate cycle). Volatility is the standard deviation of the one-month rate, annually compounded on the equal-month timescale. Long Run Rate (b) represents the equilibrium value of the one-month rate, annually compounded.

Sometimes α and σ are available from external sources, but they are quoted differently. The following equations show how to obtain the "one month" speed of mean reversion and volatility from the values of α and s corresponding to:

1. instantaneous rate, α_i(α_i and σ_i hereafter)

2. rate with term Δt years ($\alpha_{\Delta t}$ and $\sigma_{\Delta t}$ hereafter)

(For clarity we write α_{1m} and σ_{1m} for the one-month parameters)

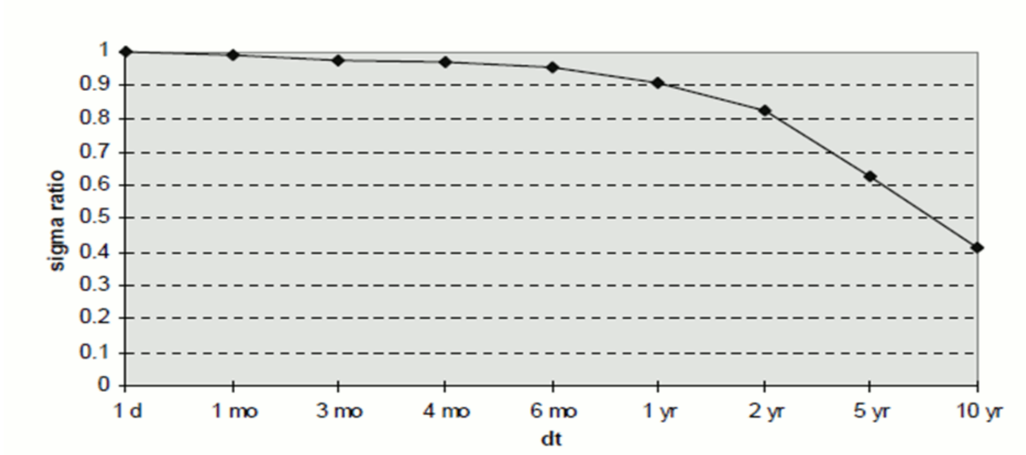
$$\alpha_{1m} = 12* (1- e^{-\alpha_{i/12}})$$

$$\alpha_{\Delta t} = (1- e^{-\alpha_i \Delta t})/\Delta t$$

$$\sigma_{1m} = 12\sqrt{12} * \frac{1-e^{-\alpha_{i/12}}}{\alpha_i} \sigma_i e^{-\alpha_{i/12}} \sqrt{(e^{\alpha_i/6} - 1)/2\alpha_i}$$

$$\sigma_{\Delta t} = \frac{1- e^{-\alpha_i \Delta t}}{\alpha_i \Delta t \sqrt{\Delta t}} \sigma_i e^{-\alpha_i \Delta t} \sqrt{(e^{2\alpha_i \Delta t} - 1/2\alpha_i}$$

The following graph shows the change in the ratio σ_{1m}/σ_i varying Dt (α_i= 0.1):



Though the decision on the best choice for the term structure parameters is left to the user, the software imposes certain restrictions on those as following:

- $0 < \alpha < 1$
- $0.01\% < \sigma < 10\%$
- $0\% < b < 200\%$

One more restriction has been set on the parameter combination for no-arbitrage models:

$(\sigma\text{-}23\alpha) > 0.7$

If this inequality does not hold, the system outputs a warning message and continues processing. Be aware that the resulting rates may not be fully no-arbitrage if the yield curve is very erratic.

Estimating Term Structure Parameters

Many discussions of the use of term structure models in risk management overlook the difficulties of estimating the parameters for such models. Market participants have become used to the idea of estimating parameters from observable market data through the popularity of the Black-Scholes option pricing model and the accepted market practice of estimating volatility for use in the model from market option prices.

"Implied volatility" is the value of volatility in the Black-Scholes model that makes the theoretical price equal to the observable market price.

Treasury Services Corporation (since acquired by Oracle in September of 1997) and Kamakura Corporation made an in-depth analysis of this parameter estimation problem. The analysis was based on the data set that includes 2,320 days of Canadian Government Bond data provided by a major Canadian financial institution. The data spanned the time period January 2, 1987 to March 6, 1996. In the last section, American swaption data provided by one of the leading New York derivatives dealers is used.

Estimating procedures for term structure models have not progressed as rapidly as the theory of the term structure itself, and leading-edge practice is progressing rapidly. There is a hierarchy of approaches of varying quality to determine the appropriate parameters.

Traditional Academic Approach

This approach follows theory precisely to estimate the stochastic process for the short term risk-less rate of interest, in this case the one-month Canadian government bill yield. This approach is generally not satisfactory in any market and was found to be unsatisfactory with Canadian data as well.

For an example of a well done analysis of the parameters of a number of theoretical models, see Chan, Karolyi, Longstaff and Sanders [4], page 11-37. The study, while well done, suffers from the typical outcome of such studies: in no case does the assumed stochastic process explain more than three percent of the variation in the short term rate of interest. We find the same problem when running the regression on 2,320 days of data using the one-month Canadian bill rate as the short rate proxy. It is concluded that this approach is not useful in the Canadian market because of the lack of correlation between the level of interest rates and change in the level of the short rate.

Volatility Curve Approach

This approach matches parameters to the historical relative volatilities of bond yields at different maturities. This approach worked moderately well.

Many market participants use parameter estimates that are consistent with the historical relative degree of volatility of longer term "yields" in relationship to the short rate. Market participants calculate observable variances for bond yields and then select values of s and α that best fit historical "volatility." The findings for the Canadian Government Bond market were based on the use of par bond coupon rates as proxies for zero coupon bond yields. The results of this analysis are summarized in the following table:

Variance in Canadian Government Interest Rates January 2, 1987 to March 6, 1996

	Canadian Treasury Bills				Canadian		Government		Bonds			
	1 mth	**2 mth**	**3 mth**	**6 mth**	**1 yr**	**2 yrs**	**3 yrs**	**4 yrs**	**5 yrs**	**7 yrs**	**10 yrs**	**25 yrs**
Actual Varia nce	7.677	8.346	7.433	6.842	6.529	3.88	3.044	2.623	2.237	1.909	1.471	0.006

	Canadian Treasury Bills				Canadian		Government		Bonds			
	1 mth	**2 mth**	**3 mth**	**6 mth**	**1 yr**	**2 yrs**	**3 yrs**	**4 yrs**	**5 yrs**	**7 yrs**	**10 yrs**	**25 yrs**
Estimated Variance	7.943	7.745	7.552	7.008	6.051	4.564	3.496	2.717	2.143	1.389	0.794	0.14
Error	-0.266	0.602	-0.119	-0.166	0.478	-0.684	-0.452	-0.095	0.094	0.52	0.677	0.856
Squared Error	0.071	0.362	0.014	0.028	0.228	0.468	0.204	0.009	0.009	0.27	0.458	0.732
Best Fitting Parameter Values:												
α	0.305172											
s	2.854318%											
Maturity	0.083	0.167	0.25	0.5	1	2	3	4	5	7	10	25

Note: Interest Rates and Variances are in %.

The best fitting α value was 0.305. This is a fairly large speed of mean reversion and reflects the relatively large variation in short term Canadian rates, relative to long rates, over the sample period. Interest rate volatility was also high at .0285. At the 1996 low levels of Canadian interest rates, this volatility level would clearly be too high. Comparable figures for the United States swap market, based on a fit to 100 swaptions prices, reflected a speed of mean reversion of 0.05 and an interest rate volatility of 0.013.

Advanced Historical Volatility Approach

This approach fits relative yield changes on all 2,320 days based on the theoretical relationship between the short rate and longer term yields by regression analysis, and then fits term structure model parameters to the regression coefficients.

For the Canadian market, a regression of par bond coupon yields (as proxies for zero coupon bond yields) on the one-month Canadian Government bill rate was performed. The results of this regression showed a higher implied mean reversion speed at shorter maturities.

Implied Speed of Mean Reversion by Historical Sensitivity to Movements in the Canadian Treasury Bill Rate (1987 - 1996)

	3 - Year Bond Yield	**10 - Year Bond Yield**	**25 - Year Bond Yield**
Coefficient of Short Rate	0.31430	0.17709	0.13505
Standard Error	0.01960	0.01573	0.01349
t - score	16.03177	11.26116	10.0127
R2	0.09985	0.05189	0.04422
Best Fitting α	1.00921	0.56275	0.29600

The best fitting α at three years was a very high 1.00921. At 25 years, the α at 0.296 is much more consistent with the historical variances. This chart provides a strong indication that a two-factor model would add value in the Canadian market (assuming other problems, like parameter estimation and the valuation of American options that are strong disadvantages of two-factor models). This is true of most markets where recent interest rate fluctuations have been large and where current rate levels are near historical lows. The Australian market has had similar experiences.

Single-Day Yield Curve Fitting

This approach is the yield curve equivalent of "implied volatility" using Black-Scholes. Most market participants feel more comfortable basing analysis on parameter values implied from observable securities prices than on historical data when observable prices are sufficient for this task. For example, if the only observable data is the yield curve itself, we can still attempt to fit the actual data to the theory by maximizing the goodness of fit from the theoretical model.

We arbitrarily set the market price of risk to zero and the long run expected value of the short rate to equal the ten-year bond yield. They then find the best fitting α and s. The result is generally of marginal acceptability. This is a common conclusion, as pointed out by the former head of derivatives research at Merrill Lynch, page 11-37, and one of the reasons why market participants often feel compelled to supplement current yield curve data with historical parameter data.

To illustrate the yield curve fitting approach, we took yield curve data for the beginning, middle and end of the data set and picked the days for which the ten-year Canadian government bond yield reached its highest and lowest points. The following maturities have been used: one-month, six-months, two, three, four, five, seven and ten years. The results of this analysis, using simple spreadsheet software to obtain parameters, were as follows:

- Best Fitting Parameters from Selected Yield
- Canadian Government Bond Market
- Extended Vasicek Model
- Using Common Spreadsheet Non-Linear Equation Solver

Environment	Date Beginning	Highest Rates	Date Mid-Point	Lowest Rates	Date Ending
Date	**January 2,1987**	**April 19, 1980**	**August 1, 1991**	**January 28, 1994**	**March 6, 1996**
Mean Reversion	0.01462	0.25540	0.62661	0.70964	0.58000
Volatility	0.00000	0.05266	0.00000	0.00000	0.00100
Mkt Price of Risk	0.00000	0.00000	0.00000	0.00000	0.00000
Long Run Rate	0.08730	0.11950	0.09885	0.06335	0.07600
Estimate Quality	**Low**	**Medium**	**Low**	**Low**	**Low**

Note: Spreadsheet solver capabilities are limited. Market price of risk and long run rate were arbitrarily set to displayed values with optimization speed of mean reversion and volatility.

The results were consistent with other approaches in generally showing a high degree of mean reversion. The lack of power in spreadsheet non-linear equation solving is reflected in the low or zero values for interest rate volatility and illustrate the need for other data (caps, floors, swaptions, bond options prices, and so on) and more powerful techniques for obtaining these parameters.

The Best Approach

Given these results, we think it is essential to use parameters estimated from observable caps, floors, and swaptions data (or other option-related securities prices) to the extent it is available. To illustrate the power of this approach, consider now the U.S. dollar data on European swaption prices observable in August 1995. At the time the data were obtained, there were 54 observable swaption prices. A swaption gives the holder the right to initiate a swap of a predetermined maturity and fixed rate level on an exercise date in the future. We estimated Extended Vasicek model parameters by choosing the speed of mean reversion (α) and interest rate volatility (s), which minimized the sum of the squared errors in pricing these 54 swaptions. The "price" of the swaption was obtained by converting the Black-Scholes volatility quotation for the swaption price to the percentage of notional principal that the equivalent dollar swaption price represented. The exercise periods on the swaptions were 0.5, one, two, three, four and five years. The underlying swap maturities were 0.5, one, two, three, four, five, six, seven and ten years.

Overall, the Extended Vasicek model's performance was extraordinary. The average model error was literally 0 basis points with a mean absolute error of five basis points of notional principal, even though only two parameters (in addition to the current yield curve) were used to price 54 securities. Compare this to the Black model for commodity futures, which is often used for swaptions and caps and floors pricing. The Black model required 54 different implied volatility values to match actual market prices, even though the model, in theory, assumes that one volatility parameter should correctly price all 54 swaptions. Volatilities in the Black model ranged from 0.13 to 0.226, a very wide range that should indicate to swaption market participants the need for caution.

In summary, the extended version of the Vasicek model, when applied to swaption prices, proved two things:

- Swaptions provide a rich data set with excellent convergence properties that enable market participants to use even common spreadsheet software to obtain high quality term structure parameter estimates.

- The accuracy of the Extended Vasicek model using only two parameters held constant over 54 swaptions, is far superior to that of the Black commodity futures model in predicting actual market prices.

In estimating term structure parameters, the lesson is clear. A rich data set of current prices of securities with significant optionality is necessary to provide an easy-to-locate global optimum for almost any popular term structure model.

Choosing a Smoothing Method

One of the most fundamental steps in fixed-income option valuation is calculating a smooth yield curve. The original yield curve from the Rate Management IRC may not have enough terms for the term structure parameter estimation routine to work properly. In particular, the trinomial lattice needs the value of the yield for every bucket point.

The simple description of a smoothing is to draw a smooth, continuous line through observable market data points. Because an infinite number of smooth, continuous lines pass through a given set of points, some other criterion has to be provided to select among the alternatives. There are many different ways to smooth a yield curve. The best technique is the one that results in the best term structure parameters.

The Rate Generator has two different smoothing techniques:

- Cubic Spline
- Linear Interpolation

Cubic Spline of Yields

One approach to smoothing yield curves is the use of cubic splines. A cubic spline is a series of third degree polynomials that have the form:

$$y = a + bx + cx^2 + dx^3$$

where:

x = years to maturity (independent variable)

y = yield (dependent variable)

These polynomials are used to connect-the-dots formed by observable data. For example, a US Treasury yield curve might consist of interest rates observable at 1, 2, 3, 5, 7, and 10 years. To value a fixed-income option, we need a smooth yield curve that can provide yields for all possible yields to maturity between zero and 10 years. A cubic spline fits a different third degree polynomial to each interval between data points (0 to 1 years, 1 to 2 years, 2 to 3 years, and so on). In the case of a spline fitted to swap yields, the variable x (independent variable) is years to maturity and the variable y (dependent variable) is yield. The polynomials are constrained so they fit together smoothly at each knot point (the observable data point); that is, the slope and the rate of change in the slope with respect to time to maturity have to be equal for each polynomial at the knot point where they join. If this is not true, there will be a kink in the yield curve (that is, continuous but not differentiable).

However, two more constraints are needed to make the cubic spline curve unique. The first constraint restricts the zero-maturity yield to equal the 1-day interest rate (for example, the federal funds rate in the U.S. market). At the long end of the maturity spectrum, a number of alternatives exist. The most common one restricts the yield curve at the longest maturity to be either straight (y''=0) or flat (y'=0). There are other alternatives if the cubic spline is fitted to zero coupon bond prices instead of yields.

Our function will also extrapolate the original yield curve outside its domain of definition. The resulting smoothed yield curve will be constant and equal to:

- the first term yield for T ≤ first term
- the last term yield for T ≥ last term

Linear Interpolation

Cubic splines have historically been the method preferred for yield curve smoothing. Inspite of the popularity of the cubic spline approach, market participants have often relied on linear yield curve smoothing as a technique that is especially easy to implement, but its limitations are well-known:

- Linear yield curves are continuous but not smooth; at each knot point there is a kink in the yield curve.
- Forward rate curves associated with linear yield curves are linear and discontinuous at the knot points. This means that linear yield curve smoothing sometimes cannot be used with the Heath, Jarrow, and Morton term structure model because it usually assumes the existence of a continuous forward rate curve.
- Estimates for the parameters associated with popular term structure models like the Extended Vasicek model are unreliable because the structure of the yield curve is unrealistic. The shape of the yield curve, because of its linearity, is fundamentally incompatible with an academically sound term structure model. Resulting parameter estimates are therefore, often implausible.

> **Note:** As in the case of cubic spline, we extrapolate for the maturities less than the first term yield and greater than the last term yield: in the former, the yield is set to be equal to the first term yield, and for the latter it is set to be the last term yield.

Defining a Rate Index Formula

The purpose of the Stochastic Rate Index assumption rule is to define a forecast of the index rates (such as Libor or Prime) based on the Monte Carlo forecast of the risk-free rate. This forecast is applicable for every time *t* in the future and every scenario."Future" means any time after the As-of-Date.

The rate index primarily should be defined under the following situations:

- The index is a contractual function of the risk-free yield.
- The index is defined through a conjunction of exogenous factors, in which case time series analysis finds the best formula.

In this section we analyze one important special case of a contractual function, namely how to translate simple rates (zero-coupon yields) into par bond coupon rates for a risk-free bond with one payment. We first describe the general formula a user can enter.

General Case

The user can enter the following formula for each index and index term:

$$\text{Index}\,(t, t+T_k, \omega) = a_0 + \sum_{i=1}^{8} a_i \,\text{Max}\,(0, R(t, t+\tau_i, \omega)^{b_i})$$

where:

a_i are coefficients, a_i Š 0

b_i are exponents; they can be integer only

ω refers to the Monte Carlo scenario

T_k is the term of the index for which the formula applies

τ_i is the term of each forecasted risk-free rate

$R(t,t+\tau_i,\omega)$ is the risk-free rate at time t for a term of τ_i

In this formula, we included the scenario ω for notational convenience, although it cannot be specified by the user: the same formula is applied for each scenario - what varies is the risk-free rate. The user can specify only a_i, b_i, T_k, τ_i, and of course of the identity of the Index IRC for which the formula applies.

If an adjustable instrument in the database is linked to an index term for which the user did not define a formula, the engine will linearly interpolate (or extrapolate) along term.

Example

Suppose the user defined formulae only for Libor 1 month and Libor 5 months, but an instrument record is linked to Libor 3 months, the engine will calculate the index rate for Libor 3 months as the average of Libor 1 month and 5 months.

Formula for a Coupon Rate

Suppose that the security is issued at time t and pays T year(s) after it has been issued. A typical example would be a 6-month T-bill, that is, T=0.5. The owner of the security will receive at time $t+T$:

$$Cash\ flow = principal * (1 + cT)$$

where c is the unknown (annual) coupon rate. In a no-arbitrage economy this cash flow should be equal to:

$$Future\ value\ of\ principal = principal * (1 + R(t, t + T, \omega))^T$$

Solving for the coupon rate gives the equation:

$$C = \frac{(1 + R(t, t + T, \omega))^T - 1}{T}$$

Because yields are usually much smaller than 100%, we can expand the numerator of the right hand side in a Taylor series:

$$(1 + R)^{\mathrm{T}} \cong 1 + TR + \frac{T(T-1)}{2} R^2 + \frac{T(T-1)(T-2)}{6} R^3 + \cdots$$

Therefore,

$$\mathrm{C} = R + \frac{(T-1)}{2} R^2 + \frac{(T-1)(T-2)}{6} R^3 + \cdots$$

The user would then input the following formula coefficients:

i	a_i	b_i	τ_i
1	1	1	T
2	$(T\text{-}1)/2$	2	T
3	$(T\text{-}1)(T\text{-}2)/6$	3	T

and all other coefficients equal to zero. The following example shows that we do not need to go very far in the Taylor series to converge to the true value of the coupon rate. This is important to remember because a long formula necessitates more computing time than a slow one.

Example

Term: T=0.5

Yield: R =0.05

Order of approximation	Formula	Coupon
1	R	0.05
2	$R+((T-1)/2)R^2$	0.049375
3	$R+((T-1)/2)R^2+((T-1)(T-2)/6)R^3$	0.049390625
True value	$((1+R)^T-1)/T$	0.049390153

Computation of Effective Duration and Effective Convexity

Oracle ALM does not directly output effective duration and effective convexity. However, a simple manual procedure computes delta and therefore effective duration and convexity. We describe this procedure hereafter.

1. Create a Forecast Rates assumption rule with 3 scenarios:

   Scenario 1: Flat Rates

   Scenario 2: Shock Up 100 bps

   Scenario 3: Shock Down 100 bps

2. Run an Oracle ALM Deterministic Process to compute market value for all 3 scenarios.

   Scenario 1: MV_0

   Scenario 2: MV_+

   Scenario 3: MV_-

3. Effective Duration can be easily computed manually (for example, in ALMBI):

$$\text{Effective Duration} = \frac{MV_- - MV_+}{(MV_0) * (r_+ - r_-)}$$

4. Effective Convexity can also be easily computed manually:

$$\text{Effective Convexity} = \frac{MV_+ + MV_- - 2(MV_0)}{MV_0[.5(r_+ - r_-)]^2}$$

Note: A similar process can be defined using the Monte Carlo engine. With this approach, you must define additional Rate Management, valuation curve IRCs that reflect the immediate parallel rate shocks to the as-of-date rates. You must additionally map each of these IRCs into separate Stochastic Rate Index rules with each rule using one of the three IRCs as the valuation curve, as appropriate. Using these Stochastic Rate Indexing rules, the Monte Carlo engine must be run three times to produce the Market Values for each scenario: (Base, +100, -100). The resulting Option Adjusted Market Value results can be fed into the above formulas to arrive at Option Adjusted Duration and Option Adjusted Convexity.

References

1. Black, F. *Interest Rates as Options.* Journal of Finance, 1995.

2. Bratley, P., and Fox. *Algorithm 659: Implementing Sobol's Quasirandom Sequence generator.* ACM Transactions on Mathematical Software, 1988.

3. Caflisch, R., Morokoff, and A. Owen. *Valuation of Mortgage Backed Securities Using Brownian Bridges to Reduce Effective Dimension.* Caflisch' World Wide Web site, 1997.

4. Chan, K.C., G.A. Karolyi, F.A. Longstaff, and A.B. Sanders. *An Empirical Comparison of Alternative Models of the Short-Term Interest Rate.* Journal of Finance, 1992.

5. Fitton, P. *Hybrid Low Discrepancy Sequences. Effective Path Reduction for Yield Curve Scenario Generation.* To appear in the Journal of Fixed Income.

6. Flesaker, B. *Testing the Heath-Jarrow-Morton/ Ho-Lee Model of Interest Rate Contingent Claims Pricing.* Journal of Financial and Quantitative Analysis, 1993.

7. Ho, T.S.Y., and S.-B Lee. *Term Structure Movements and Pricing Interest Rate Contingence Claims.* Journal of Finance, 1986.

8. Hull, J. *Options, Futures, and other Derivatives.* Prentice Hall, 1993.

9. Hull, J. *Options, Futures, and other Derivatives.* Prentice Hall, 1997.

10. Hull, J., and A. White. *One-factor Interest-Rate Models and the Valuation of Interest-Rate Derivative Securities.* Journal of Financial and Quantitative Analysis, 1993.

11. Joy, C., Boyle, and Tan. *Quasi-Monte Carlo Methods in Numerical Finance.* Management Science, 1996.

12. Lord, G., Paskov, Vanderhoof. *Using Low-Discrepancy Points to Value Complex*

Financial Instruments. Contingencies, 1996.

13. Morokoff, W., and R. Caflisch. *Quasi-Random Sequences and Their Discrepancies.* SIAM Journal of Scientific Computing, 1994.

14. Niederreiter, H. *Random Number Generation and Quasi-Monte Carlo Methods.* Regional Conference Series in Applied Mathematics, SIAM, 1992.

15. Owen, A. *Monte Carlo Variance of Scrambled Equidistribution Quadrature.* SIAM Journal of Numerical Analysis, 1996.

16. Press, W., S. Teukolski, W. Vetterling, B. Flannery. *Numerical Recipes in C, The Art of Scientific Computing.* Cambridge University Press, 1992.

17. Spanier, J. private communication, 1997.

18. Spanier, J., and Li. *Quasi-Monte Carlo Methods for Integral Equations.* Unpublished.

19. Vasicek, O.A. *An Equilibrium Characterization of the Term Structure.* Journal of Financial Economics, 1977.

12

Autobalancing

This chapter describes how to setup the Asset Liability Management (ALM) – Auto-balancing process and demonstrates how the process works.

This chapter covers the following topics:

- Introduction to Autobalancing
- Set-Up Requirements
- Process Types
- Process Steps

Introduction to Autobalancing

The auto-balancing process within Oracle Asset Liability Management simulates the following banking activities:

- Purchase of overnight funds to fill in cash shortfalls or sale of overnight funds to invest excess cash
- Roll net income into retained earnings
- Pay dividends
- Pay federal and local tax
- Accumulation of unrealized currency gain/losses in equity

Set-Up Requirements

The following set-up must be performed Oracle Asset Liability Management (ALM), prior to generation of auto-balancing results:

1. Selection of all auto-balancing accounts in ALM Application Preferences. This

includes:

- Auto-balancing Asset (plug account)
- Auto-balancing Liability (plug account)
- Retained Earnings
- Local Taxes
- Federal Taxes
- Dividends
- Accumulated Translation Account

2. Definition of tax rates and dividend amounts (if applicable) per modeling bucket in the Active ALM Time Bucket rule. This includes:
   - Federal tax percentage
   - State tax percentage
   - Dividend amount <or>
   - Dividend percentage

3. Set-up of Product Characteristic information for the auto-balancing asset plug member and auto-balancing liability plug member.

4. Verify that all auto-balancing product <leaf> members are tied to Common COA Ids with the correct account type designation. The following mappings are required:

Auto-balancing Account	Account Type Attribute (through Common COA)
Asset	Earning Asset
Liability	Interest Bearing Liability
Retained Earnings	Equity
Dividends	Dividends
Federal Taxes	Taxes

Auto-balancing Account	Account Type Attribute (through Common COA)
State Taxes	Taxes
Accumulated Translation Balance	Equity

5. Percentage taxable information is correctly defined in Product Characteristics for each product <leaf> member.

Process Types

Auto-balancing available for the following types of ALM Processing:

- Dynamic Deterministic Product-only output
- Dynamic Deterministic Product Currency output with Consolidated option selected
- Earnings at risk output

Process Steps

Generation of Accumulated Translation Product <Leaf> Member results

The standard cash flow process generates results for the accumulated translation amount financial element for all modeling buckets and for all product members assigned to the **Current Rate** method. The **Current Rate** method is one of three accounting methods available in the application for treatment of currency gain/loss amounts.

The Accumulated Translation Account is a special auto-balancing member, assigned by the user within ALM Application Preferences. The balance for this product <leaf> member holds the aggregated accumulated translation amounts resulting from all product leaf members included in the process. The Accumulated Translation product must have an account type of equity.

The following steps are performed first because other auto-balancing processes rely on these results. The auto-balancing process generates the financial element rows, as follows:

1. Generate an aggregate value for the accumulated translation amount financial element for every modeling bucket and scenario. When aggregating data from product members, include results only from the following account types, with the

sign as defined in the parentheses:

   1. Earning Assets (+)

   2. Other Assets (+)

   3. Interest Bearing Liabilities (-)

   4. Other Liabilities (-)

   5. Equity (-)

2. Select the **Current Balance** for the Accumulated Translation Account from Result Master for the start date index equal to 0. Note that this data may not exist.

3. Transform the accumulated data into beginning, ending, and average financial element data for each modeling bucket and scenario.

   - The aggregated accumulated translation financial element will equal the ending balance for each bucket and scenario.

   - The beginning balance in all buckets beyond the first bucket will equal the ending balance in the previous bucket.

   - The current balance selected from Result Master will serve as the beginning balance in bucket 1 for all scenarios. If no data exists in Result Master, the beginning balance is zero.

   - The average balance will equal the beginning balance from the same bucket.

4. Because the Accumulated Translation Account is an Equity account and Equity accounts do not generate gap results, no data needs to be generated for Gap financial elements.

5. Within Result Master, the CUR_PAR_BAL and CUR_NET_PAR_BAL needs to be generated for all scenarios and start dates. The CUR_PAR_BAL and CUR_NET_PAR_BAL will also be the same for all start date / scenarios. To determine the proper balance amount for a start date, follow the subsequent logic:

   - If the start date is equal to a bucket end date, the Ending Balance for that bucket is used.

   - If the start date is equal to a bucket start date or falls within the modeling bucket, the Beginning Balance is used.

Generation of Auto-balancing Assets and Liabilities

The generation auto-balancing assets and liabilities requires that a cash account is maintained over the modeling horizon. This cash balance results from principal, interest, and non-interest payments. For each modeling bucket, the process simulates overnight investment of cash or purchase (borrowing) of required cash.

Accumulation of Values

The process requires the following accumulated values from the results:

Cash Balance: The cash excess/ shortfall for each modeling bucket from 0 to 240, or maximum modeling bucket.

Net Income: Net income for each modeling bucket from 1 to 240, or maximum modeling bucket.

Net Taxable Income: Net taxable income for each modeling bucket from 1 to 240, or maximum modeling bucket.

The following steps accumulate the cash balances in each modeling bucket.

1. For Cash Balance_0, accumulate the CUR_NET_PAR_BAL + CUR_DEFER_BAL_C from Result Master for each scenario for the rows where start_date_index = 0 (the current position rows). The proper sign for each account type is listed as follows. Account types that are not listed are not be included in this accumulation.
   - Earning Asset (-)
   - Interest-bearing Liability (+)
   - Other Asset (-)
   - Other Liability (+)
   - Equity (+)
2. For Cash Balance_1 to Cash Balance_{240}, accumulate the ending balance financial element and the ending deferred balance financial element in each modeling bucket and each scenario. The proper sign for each account type is the same as shown earlier. Account types that are not listed are not be included in the accumulated values.
3. For Net Income_1 to Net Income_{240}, accumulate net income per scenario and modeling bucket and net taxable income (Net Taxable Income_b) per modeling bucket as follows. Accumulate the following financial elements:
   - Interest accrued net

- Deferred runoff
- Non-interest income
- Non-interest expense
- Realized currency gain/loss on net interest
- Realized currency gain/loss on principal
- Total currency gain/loss
- Taxes
- Dividends

Depending on the account type, either add or subtract these values. The sign for each account type is as follows:

- Earning Asset (+)
- Other Asset (+)
- Off Balance Sheet Receivable (+)
- Interest Income (+)
- Non Interest Income (+)
- Interest Bearing Liability (-)
- Other Liability (-)
- Off Balance Sheet Payable (-)
- Non Interest Expense (-)
- Taxes (-)
- Dividends (-)

For Net Taxable Income$_b$, accumulate the same financial element values given earlier. In addition, multiply each value by the taxable percent for the product <leaf> member. The taxable percent is found each product member, in Product Characteristics.

Processing Cash Balance

The Net Income$_b$plus the Cash Balance$_b$ equals the available cash balance at each bucket end date through the modeling horizon.

Two additional events affect the balance of the cash account.

1. The cash account itself will earn or pay out interest.

2. At the end of each modeling bucket, taxes and dividends will be computed on the net income and net taxable income and paid out of cash.

The calculation of interest earned or paid out from the cash account must first be simulated, because this value also impacts the net income.

The first step is to derive an initial cash amount for every bucket end date, equal to Cash Balance_b plus Net Income_b. Let's call this amount Total Cash_b. Total Cash_0, is equal to the starting cash amount, is equal to just the Cash Balance_0. In addition to this variable, there are four additional variables.

- Total Cash_b is the initial cash amount for every bucket end date, equal to Cash Balance_b plus Net Income_b. Total Cash_0, equal to the starting cash amount, is equal to just the Cash Balance_0. This amount does not include adjustments for interest on the cash account or taxes and dividends paid out of the cash account.

- Daily Cash_b^n is the Daily cash balance for day **n** in each modeling bucket **b**. The variable **n** designates the day into the modeling bucket, for example, n = 1 is the modeling bucket start date.

- Net Interest On Cash_b is the total interest earned over the modeling bucket **b** on the cash account.

- Incremental Cash is the incremental additional cash added due to changes in activity on the balance sheet. Because this value is reset for every modeling bucket, it does not require a **b** subscript.

- To Date Cash is a running total of cash added from processing the cash account in previous modeling buckets. Because this value is a running total for all modeling buckets, it also does not require a **b** subscript.

- Rate_b is the forecasted interest rate for the interest rate code associated with the Auto-balancing Asset or Liability account for the current modeling bucket and scenario. Note that the sign of the current balance will determine whether the interest rate code for the auto-balancing asset or for the auto-balancing liability account should be used.

For each modeling bucket, interest can be calculated as follows:

N = number of days in current modeling bucket

$IncrCash = 0$

$NetIntOnCash_b = 0$

For $n = 1$ to N

$DailyCash_b^n = TotalCash_{b-1} + IncrCash + NetIntOnCash + ToDateCash$

If $DailyCash_b^n >= 0$

Then $Rate_b = AssetRate_b$

Else $Rate_b = LiabilityRate_b$

End If

$DailyCash_b^n = TotalCash_{b-1} + IncrCash + NetIntOnCash + ToDateCash$

$NetIntOnCash_b = (DailyCash_b^n) * Rate_b / 365 + NetIntOnCash_b$

$IncrCash = (TotalCash_b - TotalCash_{b-1}) / N * n$

Next n

After processing of cash account, taxes and dividends must be calculated and deducted from the cash account prior to calculation of the final cash balance.

$StateTaxes_b = StateTaxRate_b * (NetTaxInc_b + NetIntOnCash_b)$

$FedTaxes_b = FedTaxRate_b * (NetTaxInc_b + NetIntOnCash_b)$

$Dividends_b = DivAmount_b + DivRate_b * (NetInc_b + NetIntOnCash_b)$

The **To Date** additional cash must be adjusted for the start of processing of the next modeling bucket:

$ToDateCash = ToDateCash + NetIntOnCash_b - StateTaxes_b - FedTaxes_b - Dividends_b$

The Ending Balance of the current modeling bucket must be adjusted for the cash paid out for taxes and dividends

$DailyCash_b^N = DailyCash_b^N - StateTaxes_b - FedTaxes_b - Dividends_b$

For calculation of the financial elements for each modeling bucket,

Beginning Balance = $DailyCash_b^1$

Ending Balance = $DailyCash_b^N$

Average Balance = For $n = 1$ to N: $(\sum DailyCash_b^n) / N$

Interest Cash Flow = $NetIntOnCash_b$

Interest Accrued = $NetIntOnCash_b$

Beginning Net Rate = $Rate_b * DailyCash_b^1$

Ending Net Rate = $Rate_b * DailyCash_b^N$

Average Net Rate = $\sum Rate_b * DailyCash_b^n$

Note that the calculation of average net rate is slightly more complicated than as shown earlier. The process must track the rate used for each day of the modeling bucket, whether it is an asset or a liability.

Generation of Retained Earnings

Once taxes and dividends have been calculated for the modeling bucket, the Ending Balance of the Retained Earnings account can be calculated as

$RetainedEarnings_b = RetainedEarnings_{b-1} + NetInc_b + NetIntOnCash_b - StateTaxes_b - FedTaxes_b - Dividends_b$

Summary of Output

Once all calculations have been performed, the following financial elements need to be output for the auto-balancing accounts into the Result Detail Tables.

Taxes State

Tax

Taxes Federal

Tax

Dividends

Dividend

Auto-balancing Asset and Auto-balancing Liability

Beginning Balance

Ending Balance

Average Balance

Beginning Net Rate

Ending Net Rate

Average Net Rate

Interest Accrued Net

Interest Cash Flow Net

Retained Earnings

Beginning Balance

Ending Balance

Average Balance

Accumulated Translation Account

Beginning Balance

Ending Balance

Average Balance

If Gap was processed for the run, the Auto-balancing Asset and Liability accounts must include output for each start date, equal to the balance as of that date. The value will appear as financial element Total Runoff and Total Principal Runoff.

In the Result Master table, the Auto-balancing Asset and Liability accounts, the Accumulated Translation Account, and the Retained Earnings Account will display output to the CUR_PAR_BAL, CUR_NET_PAR_BAL columns, equal to the balance as of the start date. Additionally, the Auto-balancing Asset and Liability accounts will include output to the CUR_NET_RATE column, equivalent to the rate as of that date.

13

Forward Starting Instruments

Definition of Forward Starting Instruments

Forward Starting Instruments are those instruments whose Origination Date is greater than the As of Date.

Features and Calculation Process of Forward Starting Instruments:

1. On - balance sheet products with issue date <= as of date < origination date, FE 212 and FE213, are output on the origination date bucket in Result Detail table. They have a negative value indicating the principal cash outflow. FE662 is output if it is a repricing instrument.

2. 2. For Off-balance sheet products with issue date <= as of date < origination date, if principal exchange flag is *on*, then, FE212, FE213, FE660, FE661, FE662, FE663, FE670, FE671, FE672, FE673, FE674, FE675, FE676, FE677, FE680, FE690, **FE1660, FE1661,** FE1663, FE1670, FE1671, FE1672, FE1673, FE1674, FE1675, FE1676, FE1678, FE1679, FE1680, FE1710, FE1715 and FE1716 are output on the origination date bucket in Result Detail table as applicable. The same sets of FEs are output to other buckets as applicable. FE662 is output if it is a repricing instrument. FE661, FE1661, FE1710 and FE1715 represent the principal outflow.

3. For off-balance sheet products with issue date <= as of date < origination date, if principal exchange flag is *off*, then all FE's except for FE 1660 and 1661 are output as described in Step 2 earlier. Since there is no principal exchange, there will be no negative output for the Liquidity related FE's.

4. For the deferred amount (FEs 663, 1663) it is the one time output on origination, not spread across the buckets to maturity.

5. The FE212 is output to Process Cashflows as well.

14

Historical Simulation

Features of Historical Simulation

1. The Cash Flow Engine (CFE) supports up to 2100 rate paths.
2. CFE outputs Market Value, Value at Risk (VaR) and Earnings at Risk (EaR) for all the rate paths the user requires.
3. CFE writes VaR results to the fsi_o_stoch_var and fsi_o_stoch_tot_var tables. The probabilities are arrived at by taking the inverse of the number of rate paths. That is, if the number of rate paths is 2000, then the probability values will start from 0.0005 which is 1/2000 and reach to 1.0 in steps of 0.0005. The At Risk Period is the VaR Term to populate in the table. The leaf level outputs go into fsi_o_stoch_var and the aggregated results into fsi_o_stoch_tot_var table.
4. CFE writes EaR results to the ear_leaf_dtl_xxxxxx, ear_leaf_avg_xxxxxx, ear_total_dtl_xxxxxx and ear_total_avg_xxxxxx tables.
5. CFE writes the market value results to the fsi_o_stoch_mkt_val table. This is the average of the Present Values across the rate paths aggregated by the product.
6. CFE outputs the valuation curve discount rates to FSI_INTEREST_RATES_AUDIT. These will be the forward rates for 360 months for five random rate paths. We are not outputting index IRC rates so as to not load the CFE, but the user can calculate it as needed using the relationship defined.
7. CFE populates the FSI_O_PROCESS_CASH_FLOWS also for auditing purposes, depending upon the user selection.
8. <u>Historical Simulation Algorithm:</u>
    1. Engine uses the historical rates for the valuation curve as the new interest scenarios (rate paths).

2. CFE picks up the latest historical rates (nearest to As of Date) based on the number of rate paths required.

    Note: Processing takes the lesser of the given number -

    number of rate paths given(NUM_OF_RATE_PATHS column in FSI_ALM_STOCHASTIC_PROCESS table) or the (distinct historical rates in fsi_irc_rate_hist table)

3. Rate Index definition gives the relationship between the valuation curve and the Index IRC used.

4. Each of the valuation IRCs is then forecasted based on the Implied Forward rate generation for 360 months. Index IRCs derive the rates of each scenario using the rate index relationship. The forward rates use the cubic spline interpolation. Discounting happens using the forward rates of the valuation IRC for each rate path, continuously compounded.

5. Market Value is the average of the Present Values across all the rate paths.

6. VaR for the holding period is computed for all the rate paths using the calculated market value as follows:

    VaR = Market Value - Dynamic Accrued Present Value

7. Dynamic Accrued Present Value is computed per rate path as:

    Present Value/Stochastic Discount Factor

    The Stochastic Discount Factor is calculated in the following manner:

    1. Take the valuation rate from the valuation curve.

    2. Convert it into a continuously compounded rate using the formula e^{rt}

    3. Multiply the rate from step 2 with the discount rate pertaining to the previous period.

15

Derivatives in Stochastic Processes

Derivatives in Stochastic Processes

1. The following matrix gives the mapping of the user selection to the table populated by the Cash Flow Engine (CFE). The same is used for populating the offbalance sheet products as well.

Table	VaR selection	EaR selection	Market Value selection	Monthly Rates selection
FSI_O_STOCH_TOT_VAR	Yes			
FSI_O_STOCH_VAR	Yes			
FSI_O_STOCH_MKT_VAL			Yes	
EAR_LEAF_AVG_XXXXX		Yes		
EAR_LEAF_DTL_XXXXX		Yes		
EAR_TOTAL_DTL_XXXXX		Yes		
EAR_TOTAL_AVG_XXXXX		Yes		

Table	VaR selection	EaR selection	Market Value selection	Monthly Rates selection
FSI_INTEREST_ RATES_AUDIT				Yes

2. Specifically for all the derivative swaps, the above mentioned tables are populated. A new column leg type outputs the results for the leaf level tables. The list of products are:
    1. Vanilla Swap
    2. Basis Swap
    3. Set in arrears Swap
    4. Amortizing Swap
    5. Asset Swap
    6. Forward Swap
    7. Currency Swap
    8. FRA
    9. Interest Rate Collars
    10. Interest Rate Caps
    11. Interest Rate Floors
3. Leaf level VaR outputs the values at the leg level. The outputs are positive for both payable and receivable while the Total VaR output considers the payable as negative and receivable as positive and hence writes the netted value. One exception is Collars where currently Leg Type 1 is Cap and 2 is Floor. Here, a signage is attached at the cashflow level for the interest income based on the purchase sale logic and the cap/floor indicator. So, leaf VaR outputs as such while the cap/floor values are aggregated for Total VaR. VaR for Cap and Floor instruments also are output alongwith the signage representing the inflow/outflow.
4. Leaf level EaR values outputs at the leg level. The outputs are positive for both payable and receivable legs while the Total EaR output considers the payable as negative and receivable as positive and hence writes the netted value. One exception is Collars where currently Leg Type 1 is Cap and 2 is Floor. Here, a

signage is attached at the cashflow level for the interest income based on the purchase sale logic and the cap/floor indicator. So, leaf EaR is output as such and the cap/floor values are aggregated for Total EaR. EaR for Cap and Floor instruments are output alongwith the signage representing the inflow/outflow.

5. Leaf Average EaR values are output at the level taking the simple average of the Leaf EaR values across the scenarios. Similarly, Total average values are output taking the simple average of the scenarios for all the leaves.

6. Earnings are equivalent to net interest accrued.

7. Net Interest Income is the interest income less the interest expense. CFE considers interest accrued of assets and liabilities to arrive at the net interest income. Net Income is the interest income and non interest income combined less the interest expense and non interest expense. CFE considers interest accrued of all the account type products as explained in the preceding sections.

8. In the case of futures, the Aggregation of the instruments is based on the LEG_TYPE in the instrument record. The buys /sell are related to the leg type in the following manner: Buy = Receive fixed = Leg type 2 and sell = pay fixed = leg type 1

    1. Buy Instrument records would have positive financial measures.

    2. Sell Instrument records would have negative financial measures.

9. For futures in Stochastic Process, the Consolidation Logic is as follows:

    Market Value and CUR_PAR_BAL is aggregated with the LEG_TYPE, so buys and sells net out.

16

Bonds with Embedded Options

Introduction to Options

An option is a contract between a buyer and a seller in which the buyer has the right, but not the obligation, to purchase (in the case of a call option) or sell (in the case of a put option) a specified underlying asset at a specified price during or at the end of a specified period. The option seller, or writer, grants this right in return for the option price, or premium. The option buyer is long the contract; the seller is short.

Bonds with Embedded Options

In finance, a **bond option** is an option to buy or sell a bond at a certain price on or before the option expiry date.

FSI_D_BORROWINGS, FSI_D_LOAN_CONTRACTS and FSI_D_INVESTMENTS are the instrument tables which support the embedded options functionality. The indicator in the instruments tables in the EMBEDDED_OPTION_FLG column. Other relevant tables for embedded options are as follows:

FSI_D_EMBEDDED_OPTIONS

Column Name	Display Name	Description
AS_OF_DATE	As of Date	The date at which the data is current.

Column Name	Display Name	Description
EXERCISE_TYPE_CD	Exercise Type Code	This attribute stores the exercise type code or option style of the embedded option. List of valid values are European, American and Bermudan.
ID_NUMBER	ID Number	Unique record identifier such as account number.
OPTION_START_DATE	Option Start Date	This attribute stores the option start date and is valid for American style of options.
OPTION_TYPE_CD	Option Type Code	This attribute stores the option type of the embedded option. List of valid values are Call and Put.
RATE_LOOKUP_CD	Rate Lookup Code	This attribute stores the rate lookup code. Depending upon the rate lookup, the rate of the appropriate term is compared against the strike rate. For instance if the rate lookup code is Payment Frequency, Strike IRC term point of payment frequency is compared against the strike rate. List of valid values are Payment Frequency, Reprice Frequency, Original term and remaining term.
STRIKE_IRC_CD	Strike Interest Rate Code	This attribute stores the coded value of the interest rate curve which is linked to the strike.

Column Name	Display Name	Description
STRIKE_TYPE_CD	Strike Type Code	This attribute stores the strike type code of the embedded option. A strike type of price assumes that the price of the instrument is used as the strike. A strike type of rate assumes that the interest rate of the instrument is used as the strike. List of valid values are Price and Rate.

FSI_D_EMBEDDED_OPTIONS_SCH

Column Name	Display Name	Description
AS_OF_DATE	As of Date	The date at which the data is current.
ID_NUMBER	ID Number	Stores Identifier Number.
INSTRUMENT_CODE	Instrument Code	Stores the instrument code.
OPTION_EXPIRY_DATE	Option Expiry Date	This attribute stores the expiry date of the option.
STRIKE_VALUE	Strike Value	This attribute stores the strike rate or price depending on the strike type of the option.

FSI_OPTION_DECISION_TYPE_MLS

OPTION_DECISION_TYPE_CD	OPTION_DECISION_TYPE	Description
1	Cashflow to Maturity	Cashflow to Maturity.
2	Cashflow to Exp Date	Cashflow to First Expiry Date.

OPTION_DECISION_TYPE_ CD	OPTION_DECISION_TYPE	Description
3	Rate Path Dependent	Rate Path Dependent
0	None	None

FSI_OPTION_TYPE_MLS

OPTION_TYPE_CD	OPTION_TYPE	Description
1	Call	Call
2	Put	Put
-1	None	None

FSI_OPTION_EXERCISE_MLS

OPTION_EXERCISE_CD	OPTION_EXERCISE	Description
1	American	American
2	Bermudan	Bermudan
3	European	European

FSI_OPTION_STRIKE_TYPE_MLS

OPTION_STRIKE_TYPE_CD	OPTION_STRIKE_TYPE	Description
1	Price	Price
2	Rate	Rate

FSI_OPTION_RATE_LOOKUP_MLS

OPTION_RATE_LOOKUP_CD	OPTION_RATE_LOOKUP_TYPE	Description
1	Remaining Term	Remaining Term
2	Original Term	Original Term
3	Payment Frequency	Payment Frequency
4	Reprice Frequency	Reprice Frequency

Deterministic Approach for Price Based Strike

For deterministic approach, the Cash Flow Engine recognizes the option type, reads the exercise schedule of the Option, checks if the option is exercisable (Callable/Puttable) and outputs the cash flows accordingly.

- Depending on the Call/Put option the corresponding cashflows are bucketed, the financial measures (market value, Duration, Modified Duration and YTM) calculated, and the RES_DTL table financial elements also calculated.
  - The required discount method to use for embedded option valuation is Forecast Remaining Term. The rationale detailed as follows: In the case of fixed rate instruments when we create different scenarios, the difference in NPV would be from the changes in the discounting IRCs. In order for OFSALM to support this, spot input and spot IRC methods cannot be used, as the scenarios created through forecast rates are not applicable. Forecast Remaining Term and Forecast Original Term are the methods which retain the shock scenarios defined as part of the forecast rate scenarios and use them for discounting. However, in case of Forecast Original Term, all cash flows are always discounted with the original term of the instrument, which would have a significant effect on the bond NPV. So, forecast remaining term in the discount rates rules is the viable solution.
- Floating and adjustable rate instruments are not supported for Bonds with embedded options. The embedded options would be ignored and the instrument would be treated like a normal bond.
- If the option is exercised, the RESULT_MASTER outputs such as Duration, Modified Duration and Yield to Maturity are calculated with the cashflows between the as of date and option exercise date and the maturity amount/call amount. YTM

would be the Yield to Call (functional), rather than YTM.

- If the Embedded Option Decisioning is selected as Cashflows to Maturity – none of the FE's pertaining to Bonds with Embedded Options would be outputted in the RES_DTL and process cash flows. The process will ignore the embedded option and calculate cashflows to the maturity of the instrument.

European Expiry

An **European expiry** may be exercised only at the **expiry date** of the option, that is, at a single pre-defined point in time.

Call Option

- Firstly, the underlying bond has to be priced for the given 'As of Date'.

$$P_{DP} = \left\{ \sum_{n=1}^{N} \frac{C}{(1+i_t)^n} \right\} + \frac{M}{(1+i_t)^N}$$

$$P_{CP} = P_{DP} - Accrued\ Interest\ from\ LDP\ and\ AOD$$

Where,

- *i_t= Interest rate at time t*
- *C = F * c = coupon payment*
- *Where, F = Face Value of the bond*
- *c = Coupon rate*
- *N = number of payments*
- *M = maturity value of the bond*
- *P_{CP}= Clean price of bond*
- *P_{DP}= dirty price of bond*
- *LDP = Last Payment Date*
- *AOD = As of Date*

- Calculate the present value of the coupon payments during the life of the option. (Coupon payments between the As of Date and the Option Expiry Date(OED))

$$PV_{Coupons} = \left\{ \sum_{t=AOD}^{OED} \frac{C}{(1+i_t)^n} \right\}$$

- Calculate the Bond Forward Price as of the Option Expiry Date

$$P_{Forward-Dirty} = \left\{ P_{DP} - PV_{Coupons} \right\} * e^{rT}$$

Where,

- *T is the Option Expiry Date – As of Date*
- *r is the rate of interest for Option Expiry Date*

- Calculate the $P_{Forward\text{ - }Clean}$ as equal to

$P_{Forward-Dirty}$ – *Accrued Interest between OED and last coupon date before OED*

- If Strike Price is less than $P_{Forward\text{ - }Clean}$

  then the Call Option can be exercised.

- Market Value when the option on the instrument is exercised is as follows:
  - Market Value – Dirty is calculated as the sum of Present values of
    - Coupon cashflows between As of Date and Option Exercise Date
    - Accrued Interest between Option Exercise Date and next payment date after OED
    - Option – Strike Price
  - Market Value – Clean will be the Market Value –Dirty less the Accrued Interest as of the AOD

- Market Value when the option is not exercised: will be the Market Value of the underlying instrument.

  Note: If Strike Price is greater than $P_{Forward\text{ - }Clean}$ then the Put Option can be exercised. The rest of the procedure for calculating Put Option is the same as that for Call Option.

Bermudan Expiry

A **Bermudan option** is an option where the buyer has the right to exercise at a set (always discretely spaced) number of times. This is intermediate between a European option—which allows exercise at a single time, namely expiry—and an American option, which allows exercise at any time.

The Forward induction method is used for valuing the embedded option in the case of Bermudan expiry. This method would work from the first option expiry date, closest to the as of date, and determine if it is exercisable (in the money) by considering the remaining cash flows to maturity. If not, it would move to the next execution date and again discount the remaining cash flows to maturity. If the option is in the money for an expiration date, the option would be executed and no further decisioning made.

American Expiry

An **American option** can be exercised at **any** time after the option start date and before the option expiry date.

For valuing the embedded option in the case of American expiry, the Bond is treated as a Bermudan Expiry Bond option with Exercise Dates on every bucket start date. Therefore, the more granular the bucket definitions, the more precise the American option decisioning will be.

Deterministic Approach for Rate Based Strike

Rate Based approach is comparing the Strike Rate against the Strike IRC from the FSI_D_EMBEDDED_OPTIONS table. And, the term selection is a user input – Payment Frequency, Reprice Frequency, Remaining Term of the Instrument and Original Term of the Instrument.

Depending on the Rate lookup selection by the user, the term is chosen and compared against the Strike Rate.

European Expiry

- Depending on the Strike IRC defined in the EMBEDDED_OPTIONS table and the forecast rules as per the forecast rules definition, the rates are forecasted for Strike IRC.
- Depending on the Rate lookup selection by the user, the term is chosen and compared against the Strike Rate.
- Bucket Rates in which the option exercise date falls are to be used.

1. Call Option

    1. For Assets, if the Forecasted Rate > Strike Rate ->Then the instrument is callable.

    2. For Liabilities, if the Strike Rate > Forecasted Rate -> Then the instrument is callable.

2. Put Option

    1. For Assets, if the Strike Rate > Forecasted Rate -> Then the instrument is Puttable.

    2. For Liabilities, if the Forecasted Rate > Strike Rate -> Then the instrument is Puttable.

Bermudan and American Expiry

Bermudan and American Expiry follow the same approach as the European expiry. Just that the check against the Strike Rate has to be done at all the exercise dates for each of the scenarios defined from the forecast rules.

17

Financial Elements

This chapter contains the descriptions and calculation details of Oracle Asset Liability Management (ALM) financial elements.

This chapter covers the following topics:

- Accumulation Methods
- Financial Element Listed by Description
- Financial Elements Listed by Number
- Financial Element Type - Cash Flow: Income Statement Accounts
- Financial Element Type - Cash Flow: Simple Accounts
- Financial Element Type – Repricing Gap
- Financial Element Type – Liquidity Gap
- Element Type - Market Value
- Financial Element Output by Account Type
- Currency Translation Methods for Financial Elements

Accumulation Methods

Accumulation methods are applied to the summary financial information calculated at each event in order to generate financial element data for each modeling bucket. There are five different accumulation methods:

- Average Method
- Accrual Method
- Sum Method
- At First Method

- At Last Method

Each of these methods is described in detail in this section.

Average Method

The average method is used to calculate an average value (such as Average Balance, Average Net Rate) over a bucket. The calculation sums up the daily values and divided by the number of days in the bucket.

Daily Average Balance = ∑ (Daily Balance)/days in bucket

All simulated events (originations, payments, prepayments, and repricings) are assumed to happen at the end of the event date. This implies that the balance and rate on the day of an event is counted as the value prior to any changes made by the event. Changes made, impact the value of the next day.

Accrual Method

The accrual method is used to determine how much accrual has occurred over the modeling bucket. The accrual method is determined by the code value in the detail record. Interest in advance instruments calculate interest accruals from the current payment date to the next payment date. Interest in arrears instruments calculate interest accruals from the current payment date to the previous payment date.

The interest cash flow is divided by the number of days between these two dates to determine a daily accrual for each day within the modeling term. Daily interest accruals are summed by modeling bucket.

Daily Interest Accrual = Interest Cash Flow/ number of days in payment.

The following example demonstrates an interest accrual for an arrears record:

Payment Date	Interest Cash Flow	Days in Payment	Daily Accrual
January 15	950	31	30.64
February 15	900	31	29.03
March 15	850	28	30.36

Modeling Start Date = January 1

Bucket End Date	Accrual Calculation	Interest Accrual
January 31	15 days @ 30.64 + 16 days @ 29.03	924.08
February 28	13 days @ 29.03 + 15 days @ 30.36	832.79

Sum Method

Summed financial element values are calculated by adding together all values associated with events occurring during the modeling bucket.

Principal Runoff = ∑ (Principal Runoff)

At First Method

At First accumulation method determines the value from the first event within a modeling bucket (for example, Beginning Balance).

At Last Method

At Last accumulation method determines the value from the last event within a modeling bucket (for example, End Balance). There are two Financial Element listings in this chapter. The first contains all elements by element description and the second by element number.

Financial Element Listed by Description

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Accrued Interest Average Balance	470	Accrual		DCF, I
Accrued Interest Ending Balance	460	Accrual		DCF, I
Accrued Interest (Without Offset)	442	Accrual		DCF, I

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Accrued Interest Net (Without Offset)	443	Accrual		DCF, I
Accumulated Interest CF Gross	448	Sum		DCF, I
Accumulated Interest CF Net	447	Sum		DCF, I
Accumulated Interest CF Transfer Rate	449	Sum		DCF, I
Accumulated Translation Amount	950	At Last		DCF
After Repricing Gross Rate	270	At Last	Repricing Balance At End	DCF
After Repricing Net Rate	290	At Last	Repricing Balance At End	DCF
After Reprice Transfer Rate	310	At Last	Repricing Balance At End	DCF
Annual Prepayment Rate	510	Sum	Balance Before PrePay	DCF
Average Account Balance	141	Daily Average	Average Balance	B, DCF
Average Bal	140	Daily Average		B, DCF
Average Basis Risk Cost Rate	175	Daily Average	Average Balance	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Average Current Option Cost	173	Daily Average	Average Balance	DCF
Average Gross Rate	150	Daily Average	Average Balance	DCF
Average Historic Option Cost	171	Daily Average	Average Balance	DCF
Average Liquidity Adjustment Rate	174	Daily Average	Average Balance	DCF
Average Net Rate	160	Daily Average	Average Balance	DCF
Average Other Adjustment Rate	177	Daily Average	Average Balance	DCF
Average Percentage Active	143	Daily Average	Average Balance	B
Average Pricing Incentive Rate	176	Daily Average	Average Balance	DCF
Average Rem Term Transfer Rate	172	Daily Average	Average Balance	DCF
Average Total Balance	142	Daily Average	Average Balance	B, DCF
Average Transfer Rate	170	Daily Average	Average Balance	DCF
Average Volume Total	144	Daily Average	Average Balance	B, DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Balance Before PrePay	515	Sum		DCF
Basis Risk Cost Charge/Credit	415	Accrual		DCF, I
Before Reprice Transfer Rate	300	At First	Repricing Balance	DCF
Before Repricing Gross Rate	260	At First	Repricing Balance	DCF
Before Repricing Net Rate	280	At First	Repricing Balance	DCF
Beginning Balance	60	At First		B, DCF
Beginning Basis Risk Cost Rate	92	At First	Beginning Balance	DCF
Beginning Gross Rate	70	At First	Beginning Balance	DCF
Beginning Liquidity Adjustment Rate	91	At First	Beginning Balance	DCF
Beginning Net Rate	80	At First	Beginning Balance	DCF
Beginning Other Adjustment Rate	94	At First	Beginning Balance	DCF
Beginning Pricing Incentive Rate	93	At First	Beginning Balance	DCF
Beginning Transfer Rate	90	At First	Beginning Balance	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Charge/Credit Rem Term	452	Accrual		DCF
Convexity	730			DCF
Current Option Cost Charge/Credit	453	Accrual		DCF
Deferred Average Balance	530	Daily Average		DCF
Deferred End Balance	520	At Last		DCF
Deferred Runoff	540	Accrual		DCF
Devolvement Runoff	204	Sum		DCF
Discount Factor	492	Daily Average		DCF
Discount Factor Term (Days)	493	Sum		DCF
Discount Rate FTP	755	At Last		DCF
Discount Rate IS	490	At Last		DCF
Discount Rate LR	1490	At Last		DCF
Dividends	940	Sum		A
Duration	720			DCF
End Balance	100	At Last		B, DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Ending Basis Risk Cost Rate	132	At Last	End Balance	DCF
Ending Gross Rate	110	At Last	End Balance	DCF
Ending Liquidity Adjustment Rate	131	At Last	End Balance	DCF
Ending Net Rate	120	At Last	End Balance	DCF
Ending Other Adjustment Rate	134	At Last	End Balance	DCF
Ending Pricing Incentive Cost Rate	133	At Last	End Balance	DCF
Ending Transfer Rate	130	At Last	End Balance	DCF
Federal Taxes	930	Sum		A
Fully Indexed Gross Rate	320	Sum		DCF
Fully Indexed Net Rate	330	Sum		DCF
Gap Accrued Interest Gross	674	Accrual		DCF, I
Gap Accrued Interest Net	675	Accrual		DCF, I
Gap Accrued Interest Transfer	676	Accrual		DCF, I

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Gap Interest Credited	677	Sum		DCF, I
Gap Deferred Runoff	663	Sum		DCF
Gap Interest Cash Flow Gross	671	Sum		DCF, I
Gap Interest Cash Flow Net	672	Sum		DCF, I
Gap Interest Cash Flow Transfer	673	Sum		DCF, I
Gap Principal Runoff	661	Sum		DCF
Gap Repricing Runoff	662	Sum		DCF
Gap Runoff	660	Sum		DCF
Gap Runoff Gross Rate	680	Sum	Gap Runoff Balance	DCF
Gap Runoff Net Rate	690	Sum	Gap Runoff Balance	DCF
Gap Runoff Term	670	Sum	Gap Runoff Balance	DCF
Gap Runoff Transfer Rate	700	Sum	Gap Runoff Balance	DCF
Historic Option Cost Charge/Credit	451	Accrual		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Interest	420			DCF, I
Interest Accrued	440	Accrual		DCF, I
Interest Accrued Gross	445	Accrual		DCF, I
Interest Accrued Gross (current basis)	446	Accrual		DCF, I
Interest Accrued Gross Hist Fx Basis	681	Accrual		DCF
Interest Accrued Net	441	Accrual		DCF, I
Interest Accrued Net Hist Fx Basis	682	Accrual		DCF
Interest Accrued Transfer Hist Fx Basis	683	Accrual		DCF
Interest Accrued Transfer Rate (Cur Bas)	454	Accrual		DCF, I
Interest Amount Gross	425			DCF, I
Interest Cash Flow	430	Sum		DCF, I
Interest Cash Flow Gross	435	Sum		DCF, I
Interest Cash Flow T-Rate	437	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Interest CF (Without Offset)	438	Sum		DCF, I
Interest Credited	480	Sum		DCF
Life Cap Balance	580	Daily Average		DCF
Life Cap Effect - Amount	600	Accrual		DCF
Life Cap Effect - Rate	590	Daily Average	Life Cap Balance	DCF
Liquidity Adjustment Charge/Credit	414	Accrual		DCF, I
Liquidity Devolvement Runoff	1683			
Liquidity GAP Accrued Interest Gross	1674	Accrual		DCF, I
Liquidity GAP Accrued Interest Net	1675	Accrual		DCF, I
Liquidity GAP Accrued Interest Transfer Rate	1676	Accrual		DCF, I
Liquidity GAP Deferred Runoff	1663	Sum		DCF
Liquidity GAP Interest Cash Flow Gross	1671	Sum		DCF, I

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Liquidity GAP Interest Cash Flow Net	1672	Sum		DCF, I
Liquidity GAP Interest Cash Flow Transfer Rate	1673	Sum		DCF, I
Liquidity GAP Interest Credited	1677	Sum		DCF, I
Liquidity GAP Option Exercise Premium	1664			
Liquidity GAP PV of Interest Cash Flows	1716	Sum		DCF
Liquidity GAP PV of Principal Cash Flows	1715	Sum		DCF
Liquidity GAP PV of TP Int Cash Flows	1717	Sum		DCF
Liquidity GAP Runoff (1661 + 1663)	1660	Sum		DCF
Liquidity GAP Runoff Gross Rate	1678	Daily Average	Liquidity GAP Runoff	DCF, I
Liquidity GAP Runoff Net Rate	1679	Daily Average	Liquidity GAP Runoff	DCF, I
Liquidity GAP Runoff Term	1670	Daily Average	Liquidity GAP Runoff	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Liquidity GAP Runoff Transfer Rate	1680	Daily Average	Liquidity GAP Runoff	DCF, I
Liquidity GAP Total Runoff	1661	Sum		DCF
Liquidity Maturity Runoff	1668			
Liquidity Non Maturity – Core Runoff	1681			
Liquidity Non Maturity – Volatile Runoff	1682			
Liquidity Non Performing Asset Runoff	1685			
Liquidity Payment Runoff	1667			
Liquidity prepayment Runoff	1666			
Liquidity Recovery Runoff	1684			
Local Taxes	935	Sum		A
Market Value	710	Sum		DCF
Market Value of TP Cash Flow (P+I)	495	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Market Value of TP Interest Cash Flow	497	Sum		DCF
Maturity Runoff - Positive	195	Sum		DCF
Maturity Runoff - Negative	197	Sum		DCF
MOA Prepay Runoff	184	Sum		DCF
Net New Business	342			DCF
New Add Balance	340	Sum		DCF
New Add Basis Risk Cost Rate	372	Daily Average	New Add Balance	DCF
New Add Gross Rate	350	Sum	New Add Balance	DCF
New Add Liquidity Adjustment Rate	371	Daily Average	New Add Balance	DCF
New Add Net Rate	360	Sum	New Add Balance	DCF
New Add Other Adjustment Rate	374	Daily Average	New Add Balance	DCF
New Add Pricing Incentive Rate	373	Daily Average	New Add Balance	DCF
New Add Spread	375		New Add Balance	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
New Add Transfer Rate	370	Sum	New Add Balance	DCF
Neg-Am Balance	640	Daily Average		DCF
Neg-Am Interest	650	Accrual		DCF
Non Maturity - Core Runoff	200	Sum		DCF
Non Maturity - Volatile Runoff	202	Sum		DCF
Non Performing Asset Runoff	208	Sum		DCF
Non Interest Expense	457	Sum		N
Non Interest Income	455	Sum		N
Option Exercise Gain/Loss	478			
Option Exercise Market Value	476			
Option Exercise Rate	477			
Other Adjustment Charge/Credit	417	Accrual		DCF, I
Payment Runoff - Negative	192	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Payment Runoff - Positive	190	Sum		DCF
Period Cap Balance	550	Daily Average		DCF
Period Cap Effect - Amount	570	Accrual		DCF
Period Cap Effect - Rate	560	Daily Average	Period Cap Balance	DCF
Annual Prepayment Rate	510	Sum	Balance Before PrePay	DCF
Prepay Runoff - Negative	182	Sum		DCF
Prepay Runoff - Positive	180	Sum		DCF
Prepay Runoff Transfer Rate	245	Sum	Prepay Runoff	DCF
Present Value of Cashflows - Scenariowise	1710	Sum		DCF
Present Value of Interest Cash Flows	716	Sum		DCF
Present Value of Principal Cash Flows	715	Sum		DCF
Present Value of TP Interest Cash Flows	717	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Pricing Incentive Charge/Credit	416	Accrual		DCF, I
Rate Lookup Term (Days)	494	Sum		DCF
Realized Currency Gain/Loss (Interest)	485	Sum		DCF
Realized Currency Gain/Loss (Interest-Gross)	486	Sum		DCF
Realized Currency Gain/Loss (Interest-T-Rate)	487	Sum		DCF
Realized Currency Gain/Loss (Principal)	475	Sum		DCF
Recovery Runoff	206	Sum		DCF
Repricing Balance	250	At First		DCF
Repricing Balance At End	255	At Last		DCF
Roll Add Balance	380	Sum		DCF
Roll Add Basis Risk Cost Rate	412	Daily Average	Roll Add Balance	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Roll Add Gross Rate	390	Sum		DCF
Roll Add Liquidity Adjustment Rate	411	Daily Average	Roll Add Balance	DCF
Roll Add Net Rate	400	Sum		DCF
Roll Add Other Adjustment Rate	418	Daily Average	Roll Add Balance	DCF
Roll Add Pricing Incentive Rate	413	Daily Average	Roll Add Balance	DCF
Roll Add Transfer Rate	410	Sum		DCF
Rollover Percentage	341			DCF
Strike Rate - Rate Based Approach	218			
Tease Balance	610	Daily Average		DCF
Tease Effect - Amount	630	Accrual		DCF
Tease Effect - Rate	620	Daily Average	Tease Balance	DCF
Timing of Call Runoff Amount	215			
Timing of Cash Flow	1491	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Timing of Cash Flow (in days)	491	Sum		DCF
Timing of Devolvement Runoff	205	Sum	Devolvement Runoff	DCF
Timing of Maturity Runoff - Positive	196	Sum	Maturity Runoff - Positive	DCF
Timing of Maturity Runoff - Negative	198	Sum	Maturity Runoff - Negative	DCF
Timing of MOA Prepay Runoff	185	Sum	MOA Prepay Runoff	DCF
Timing of Non Maturity - Core Runoff	201	Sum	Non Maturity Runoff (Core)	DCF
Timing of Non Maturity - Volatile Runoff	203	Sum	Non Maturity Runoff (Volatile)	DCF
Timing of Non Performing Asset Runoff	209	Sum	NPA Runoff	DCF
Timing of Payment Runoff - Negative	193	Sum	Payment Runoff (negative)	DCF
Timing of Payment Runoff - Positive	191	Sum	Payment Runoff (positive)	DCF
Timing of Prepay Runoff - Negative	183	Sum	Prepay Runoff (negative)	DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Timing of Prepay Runoff - Positive	181	Sum	Prepay Runoff (positive)	DCF
Timing of Put Runoff Amount	217			
Timing of Recovery Runoff	207	Sum	Recovery Runoff	DCF
Timing of Total Runoff - Negative	213	Sum	Total Runoff (negative)	DCF
Timing of Total Runoff - Positive	211	Sum	Total Runoff (positive)	DCF
Timing of Writeoff Negative	189	Sum	Writeoff (Negative)	DCF
Timing of Writeoff Positive	187	Sum	Writeoff (Positive)	DCF
Total Call Runoff Amount	214			
Total Currency Gain/Loss (Principal)	465	Sum		
Total Put Runoff Amount	216			
Total Runoff - Negative	212	Sum		DCF
Total Runoff - Positive	210	Sum		DCF

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Account Type Processing*
Total Runoff Gross Rate	220	Sum	Total Runoff (positive)	DCF
Total Runoff Net Rate	230	Sum	Total Runoff (positive)	DCF
Total Runoff Transfer Rate	240	Sum	Total Runoff (positive)	DCF
Transfer Rate Charge/Credit	450	Accrual		DCF, I
WARM	500	Sum	End Balance	DCF
Writeoff Negative	188	Sum		DCF
Writeoff Positive	186	Sum		DCF

*This specifies for which account types the financial elements will be processed. The code values are B = Balance only; I = Interest only; DCF = Detail Cash Flow; A = Autobalancing; N = Non Interest.

Financial Elements Listed by Number

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Beginning Balance	60	At First		Standard	If first bucket, then it is considered as (Cur Par Balance * Percent sold) from the instrument record. If any other bucket, it is considered as Ending Balance of the previous bucket
Beginning Gross Rate	70	At First	Beginning Balance	Standard	If first bucket, then calculated as (Beginning Balance * (Current Gross Rate + Gross Margin) from instrument record). If any other bucket, then it is considered as Ending Net Rate of the previous bucket. Forecasted Rate will be adjusted depending on the reprice characteristics.

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Beginning Net Rate	80	At First	Beginning Balance	Standard	If first bucket, it is calculated as (Beginning Balance * (Current Net Rate + Margin) from instrument record). If any other bucket, it is considered as Ending Net Rate of the previous bucket. Forecasted Rate will be adjusted depending on the reprice characteristics.
Beginning Transfer Rate	90	At First	Beginning Balance	Standard	If first bucket, it is calculated as (Beginning Balance * (Current Transfer Rate + Margin) from instrument record). If any other bucket, it is considered as Ending Net Rate of the previous bucket. Forecasted Rate will be adjusted depending on the reprice characteristics.

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Beginning Liquidity Adjustment Rate	91	At First	Beginning Balance	Standard	Calculated as: Current Liquidity Adjustments Rate * Beginning Balance
Beginning Basis Risk Cost Rate	92	At First	Beginning Balance	Standard	Calculated as: Current Basis Risk Cost Rate * Beginning Balance
Beginning Pricing Incentive Rate	93	At First	Beginning Balance	Standard	Calculated as: Current Pricing Incentive Rate * Beginning Balance
Beginning Other Adjustment Rate	94	At First	Beginning Balance	Standard	Calculated as: Current Other Adjustments Rate * Beginning Balance
End Balance	100	At Last		Standard	Calculated as: Beginning Balance - Total Runoff + New Add Balance + New Roll Add Balance - Writeoff Negative

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Ending Gross Rate	110	At Last	Ending Balance	Standard	Calculated as: End Balance * (Current Gross Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Ending Net Rate	120	At Last	Ending Balance	Standard	Calculated as: End Balance * (Current Net Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Ending Transfer Rate	130	At Last	Ending Balance	Standard	End Balance * (Current Transfer Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Ending Liquidity Adjustment Rate	131	At Last	Ending Balance	Standard	Current Liquidity Adjustments Rate * Ending Balance
Ending Basis Risk Cost Rate	132	At Last	Ending Balance	Standard	Current Basis Risk Cost Rate * Ending Balance

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Ending Pricing Incentive Cost Rate	133	At Last	Ending Balance	Standard	Current Pricing Incentive Rate * Ending Balance
Ending Other Adjustment Rate	134	At Last	Ending Balance	Standard	Current Other Adjustments Rate * Ending Balance
Average Bal	140	Daily Average		Standard	Sum of (Balance * Number of days) **Note:** It includes all balances within the bucket and the corresponding number of days.
Average Account Balance	141	Daily Average	Average Balance		
Average Total Balance	142	Daily Average	Average Balance		
Average Percentage Active	143	Daily Average	Average Balance		
Average Volume Total	144	Daily Average	Average Balance		

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Average Gross Rate	150	Daily Average	Average Balance	Standard	Sum of all (Beginning Balance * (Current Gross Rate + Margin))/ Size of the bucket Forecasted Rate will be adjusted depending on the reprice characteristics.
Average Net Rate	160	Daily Average	Average Balance	Standard	Sum of all (Beginning Balance * (Current Net Rate + Margin))/ Size of the bucket Forecasted Rate will be adjusted depending on the reprice characteristics.
Average Transfer Rate	170	Daily Average	Average Balance	Standard	Sum of all (Beginning Balance * (Current Transfer Rate + Margin))/ Size of the bucket Forecasted Rate will be adjusted depending on the reprice characteristics.
Average Historic Option Cost	171	Daily Average	Average Balance		

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Average Rem Term Transfer Rate	172	Daily Average	Average Balance		
Average Current Option Cost	173	Daily Average	Average Balance		
Average Liquidity Adjustment Rate	174	Daily Average	Average Balance	Standard	Sum of (Current Liquidity Adjustments Rate * Balance) **Note:** It includes all balances within the bucket and the corresponding number of days.
Average Basis Risk Cost Rate	175	Daily Average	Average Balance	Standard	Sum of (Current Basis Risk Cost Rate * Balance) **Note:** It includes all balances within the bucket and the corresponding number of days.

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Average Pricing Incentive Rate	176	Daily Average	Average Balance	Standard	Sum of (Current Pricing Incentive Rate * Balance) **Note:** It includes all balances within the bucket and the corresponding number of days.
Average Other Adjustment Rate	177	Daily Average	Average Balance	Standard	Sum of (Current Other Adjustments Rate * Balance) **Note:** It includes all balances within the bucket and the corresponding number of days.
Prepay Runoff - Positive	180	Sum		Standard	Adjusted Prepayment Factor * Balance before Prepayment

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Timing of Prepay Runoff - Positive	181	Sum	Prepay Runoff (positive)	Prepay Detail	Prepayment Runoff Positive * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket This formula holds good if single cashflow occurs in a bucket. If multiple cashflows occur, then all will need to be aggregated.
Prepay Runoff - Negative	182	Sum		Prepay Detail	Adjusted Prepyment Factor * Balance before Prepayment It is applicable for negative volumes.
Timing of Prepay Runoff - Negative	183	Sum	Prepay Runoff (negative)	Prepay Detail	Prepayment Runoff Negative * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
MOA Prepay Runoff	184	Sum		Prepay Detail	Interest Cashflow without offset - Interest Cashflow with offset

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Timing of MOA Prepay Runoff	185	Sum	MOA Prepay Runoff	Prepay Detail	MOA Prepay Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Writeoff Positive	186	Sum		Runoff Detail	
Timing of Writeoff Positive	187	Sum	Writeoff (Positive)	Runoff Detail	Writeoff Positive * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Writeoff Negative	188	Sum		Runoff Detail	Cur Par Balance on the Maturity Date
Timing of Writeoff Negative	189	Sum	Writeoff (Negative)	Runoff Detail	Writeoff Negative * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Payment Runoff - Positive	190	Sum		Runoff Detail	Current Payment - Interest Cashflow, if positive
Timing of Payment Runoff - Positive	191	Sum	Payment Runoff (positive)	Runoff Detail	Payment Runoff Positive * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Payment Runoff - Negative	192	Sum		Runoff Detail	Current Payment - Interest Cashflow, if negative
Timing of Payment Runoff - Negative	193	Sum	Payment Runoff (negative)	Runoff Detail	Payment Runoff Negative * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Maturity Runoff - Positive	195	Sum		Runoff Detail	Cur Par Balance - Payment Runoff on the maturity date, if positive
Timing of Maturity Runoff - Positive	196	Sum	Maturity Runoff - Positive	Runoff Detail	Maturity Runoff Positive * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Maturity Runoff - Negative	197	Sum		Runoff Detail	Cur Par Balance - Payment Runoff on the maturity date, if negative
Timing of Maturity Runoff - Negative	198	Sum	Maturity Runoff - Negative	Runoff Detail	Maturity Runoff Negative * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Non Maturity - Core Runoff	200	Sum		Runoff Detail	Cur Par Balance * Non Maturity Core Percentage (from Behavior Pattern)
Timing of Non Maturity - Core Runoff	201	Sum	Non Maturity Runoff (Core)	Runoff Detail	Core Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Non Maturity - Volatile Runoff	202	Sum		Runoff Detail	Cur Par Balance * Non Maturity Volatile Percentage (from Behavior Pattern)
Timing of Non Maturity - Volatile Runoff	203	Sum	Non Maturity Runoff (Volatile)	Runoff Detail	Volatile Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Devolvement Runoff	204	Sum		Runoff Detail	Devolvement Percentage * Original Par Balance
Timing of Devolvement Runoff	205	Sum	Devolvement Runoff	Runoff Detail	Devolvement Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Recovery Runoff	206	Sum		Runoff Detail	Recovery Percentage * Cur Par Balance
Timing of Recovery Runoff	207	Sum	Recovery Runoff	Runoff Detail	Recovery Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Non Performing Asset Runoff	208	Sum		Runoff Detail	Non Performing Asset Percentage * Cur Par Balance
Timing of Non Performing Asset Runoff	209	Sum	NPA Runoff	Runoff Detail	Non Performing Asset Runoff * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Total Runoff - Positive	210	Sum		Standard	Payment Runoff Positive + Prepayment Runoff Positive + Maturity Runoff Positive + Non Performing Asset Runoff + Devolvement Runoff + Core Runoff + Volatile Runoff

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Timing of Total Runoff - Positive	211	Sum	Total Runoff (positive)	Standard	Total Runoff Positive * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Total Runoff - Negative	212	Sum		Standard	Payment Runoff Negative + Prepayment Runoff Negative + Maturity Runoff Negative
Timing of Total Runoff - Negative	213	Sum	Total Runoff (negative)	Standard	Total Runoff Negative * (Cashflow Date - Start Date of the bucket) / Total Number of days in the bucket
Total Call Runoff Amount	214				
Timing of Call Runoff Amount	215				
Total Put Runoff Amount	216				
Timing of Put Runoff Amount	217				

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Strike Rate - Rate Based Approach	218				
Total Runoff Gross Rate	220	Sum	Total Runoff (positive)	Standard	Total Runoff * (Current Gross Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Prepay Runoff Gross Rate	225				Prepay Runoff * (Current Gross Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Total Runoff Net Rate	230	Sum	Total Runoff (positive)	Standard	Total Runoff * (Current Net Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Prepay Runoff Net Rate	235				Prepay Runoff * (Current Net Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Total Runoff Transfer Rate	240	Sum	Total Runoff (positive)	Standard	Total Runoff * (Current Transfer Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Prepay Runoff Transfer Rate	245	Sum	Prepay Runoff	Prepay Detail	Prepay Runoff * (Current Transfer Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Repricing Balance	250	At First		Reprice Detail	Cur Par Balance before the reprice event
Repricing Balance At End	255	At Last		Reprice Detail	Cur Par Balance after the reprice event
Before Repricing Gross Rate	260	At First	Repricing Balance	Reprice Detail	Beginning Balance * Gross Rate before Reprice event
After Repricing Gross Rate	270	At Last	Repricing Balance At End	Reprice Detail	Beginning Balance * Gross Rate after Reprice event
Before Repricing Net Rate	280	At First	Repricing Balance	Reprice Detail	Beginning Balance * Net Rate before Reprice event

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
After Repricing Net Rate	290	At Last	Repricing Balance At End	Reprice Detail	Beginning Balance * Net Rate after Reprice event
Before Reprice Transfer Rate	300	At First	Repricing Balance	Reprice Detail	Beginning Balance * Transfer Rate before Reprice event
After Reprice Transfer Rate	310	At Last	Repricing Balance At End	Reprice Detail	Beginning Balance * Transfer Rate after Reprice event
Fully Indexed Gross Rate	320	Sum		Reprice Detail	Current Balance * Current Gross Rate after applying margin, minimum rate change
Fully Indexed Net Rate	330	Sum		Reprice Detail	Current Balance * Current Net Rate after applying margin, minimum rate change

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
New Add Balance	340	Sum		New Origination	For Forward Start Instruments, this is the cur par balance For New Add, this is the new add amount given by the user - TS Originations - FS Originations For Target Growth, if Bucket End, Beginning Balance * Growth Percentage - TS Originations - FS Originations; if Mid Bucket, (Beginning Balance * Growth Percentage) + Runoffs For Target End, if Bucket End, (Ending Balance - Beginning Balance - Transaction Strategies Originations - FS Originations + Runoffs); if Mid Bucket, (Ending Balance - Beginning Balance - Transaction Strategies Originations - FS Originations + Runoffs) For Rollover with Target Growth, if Bucket End, (Beginning

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
					Balance * Growth Percentage - TS Originations - FS Originations - Rollover originations); if Mid Bucket , (Beginning Balance * Growth Percentage - TS Originations - FS Originations - Rollover originations + Runoffs)
Rollover Percentage	341			New Origination	
Net New Business	342			New Origination	
New Add Gross Rate	350	Sum	New Add Balance	New Origination	New Add Balance * (Current Gross Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
New Add Net Rate	360	Sum	New Add Balance	New Origination	New Add Balance * (Current Net Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
New Add Transfer Rate	370	Sum	New Add Balance	New Origination	New Add Balance * (Current Transfer Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
New Add Liquidity Adjustment Rate	371	Daily Average	New Add Balance	New Origination	Current Liquidity Adjustments Rate * New Add Balance
New Add Basis Risk Cost Rate	372	Daily Average	New Add Balance	New Origination	Current Basis Risk Cost Rate * New Add Balance
New Add Pricing Incentive Rate	373	Daily Average	New Add Balance	New Origination	Current Pricing Incentive Rate * New Add Balance
New Add Other Adjustment Rate	374	Daily Average	New Add Balance	New Origination	Current Other Adjustments Rate * New Add Balance
New Add Spread	375		New Add Balance	New Origination	
Roll Add Balance	380	Sum		New Origination	Sum of (Percentage of Runoff * Actual Runoff) This can be Total runoff or any of the runoff types that make up Total Runoff

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Roll Add Gross Rate	390	Sum		New Origination	Roll Add Balance * (Current Gross Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Roll Add Net Rate	400	Sum		New Origination	Roll Add Balance * (Current Net Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Roll Add Transfer Rate	410	Sum		New Origination	Roll Add Balance * (Current Transfer Rate + Margin). Forecasted Rate will be adjusted depending on the reprice characteristics
Roll Add Liquidity Adjustment Rate	411	Daily Average	Roll Add Balance	New Origination	Current Liquidity Adjustments Rate * Roll Add Balance
Roll Add Basis Risk Cost Rate	412	Daily Average	Roll Add Balance	New Origination	Current Basis Risk Cost Rate * Roll Add Balance
Roll Add Pricing Incentive Rate	413	Daily Average	Roll Add Balance	New Origination	Current Pricing Incentive Rate * Roll Add Balance

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Liquidity Adjustment Charge/Credit	414	Accrual		Standard	
Basis Risk Cost Charge/Credit	415	Accrual		Standard	
Pricing Incentive Charge/Credit	416	Accrual		Standard	
Other Adjustment Charge/Credit	417	Accrual		Standard	
Roll Add Other Adjustment Rate	418	Daily Average	Roll Add Balance	New Origination	Current Other Adjustments Rate * Roll Add Balance
Interest	420				
Interest Amount Gross	425				
Interest Cash Flow	430	Sum		Standard	Balance * Compounded Rate per payment. Rate per payment = Accrual Basis * Current Net Rate * Payment Frequency

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Interest Cash Flow Gross	435	Sum		Standard	Balance * Compounded Rate per payment. Rate per payment = Accrual Basis * Current Gross Rate * Payment Frequency
Interest Cash Flow T-Rate	437	Sum		Standard	Balance * Compounded Rate per payment. Rate per payment = Accrual Basis * Current Transfer Rate * Payment Frequency
Interest CF (Without Offset)	438	Sum		Standard	
Interest Accrued	440	Accrual		Standard	Daily Accrual * Number of Days in the bucket. Daily Accrual = Interest cashflow / Number of days in payment period
Interest Accrued Net	441	Accrual			
Accrued Interest (Without Offset)	442	Accrual		Standard	

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Accrued Interest Net (Without Offset)	443	Accrual		Standard	
Interest Accrued Gross	445	Accrual		Standard	
Interest Accrued Gross (current basis)	446	Accrual			
Accumulated Interest CF Net	447	Sum		Standard	
Accumulated Interest CF Gross	448	Sum		Standard	
Accumulated Interest CF Transfer Rate	449	Sum		Standard	
Transfer Rate Charge/Credit	450	Accrual		Standard	
Historic Option Cost Charge/Credit	451	Accrual			
Charge/Credit Rem Term	452	Accrual			
Current Option Cost Charge/Credit	453	Accrual			

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Interest Accrued Transfer Rate (Cur Bas)	454	Accrual			
Non Interest Income	455	Sum		Standard	
Non Interest Expense	457	Sum		Standard	
Accrued Interest Ending Balance	460	Accrual			
Total Currency Gain/Loss (Principal)	465	Sum		Standard	(FE60 Ending Balance) * [(1/ Current Exchange rate) - (1/ Previous Exchange Rate)]
Accrued Interest Average Balance	470	Accrual			
Realized Currency Gain/Loss (Principal)	475	Sum		Standard	(FE210, 212 Total Runoff) * [(1/ Current Exchange rate) - (1/ Original Exchange Rate)]
Option Exercise Market Value	476				
Option Exercise Rate	477				

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Option Exercise Gain/Loss	478				
Interest Credited	480	Sum		Runoff Detail	
Realized Currency Gain/Loss (Interest)	485	Sum		Standard	Interest Cash Flow Net * [(1/Current Exchange Rate) - (1/Original Exchange Rate)]
Realized Currency Gain/Loss (Interest-Gross)	486	Sum		Standard	Interest Cash Flow Gross * [(1/Current Exchange Rate) - (1/Original Exchange Rate)]
Realized Currency Gain/Loss (Interest-T-Rate)	487	Sum		Standard	Interest Cash Flow Transfer * [(1/Current Exchange Rate) - (1/Original Exchange Rate)]
Discount Rate IS	490	At Last		Standard	Discount Rate * Present value of the cashflows
Timing of Cash Flow (in days)	491	Sum		Standard	Days in payment * Discount Rate * Present Value of the cashflows

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Discount Factor	492	Daily Average		Standard	1/(1 + Discount Rate) ^ Discount Factor Term If Spot Rate, Read from Discount Rates. If Spot IRC, read the term point from the IRC Historical Rates. If Forecast Original (or Remaining Term), read from Forecast Rates corresponding to the Original Term of the instrument (or Remaining Term as Cashflow Date - As of Date)
Discount Factor Term (Days)	493	Sum		Standard	Diff between the Cashflow Date and As of Date expressed in years
Rate Lookup Term (Days)	494	Sum		Standard	Cash Flow Date - As of Date expressed in days
Market Value of TP Cash Flow (P+I)	495	Sum		Standard	TP Cashflow * Discount Factor
Market Value of TP Interest Cash Flow	497	Sum		Standard	TP Interest Cashflow * Discount Factor

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
WARM	500	Sum	Ending Balance	Standard	Ending Balance * (Maturity Date - Bucket End Date)/365
Annual Prepayment Rate	510	Sum	Balance Before PrePay	Prepay Detail	CPR Rate
Balance Before PrePay	515	Sum		Prepay Detail	Cur Par Balance - Payment Runoff
Deferred End Balance	520	At Last		Standard	Deferred Beginning Balance - Deferred Runoff
Deferred Average Balance	530	Daily Average		Standard	Sum of all (Beginning Deferred Balance * Number of days for that balance)
Deferred Runoff	540	Accrual		Standard	Daily Deferred Runoff * Bucket Size (for SLA method)
Period Cap Balance	550	Daily Average		Cap/Floor/Tease/NGAM	Cur Par Balance on the cashflow date
Period Cap Effect - Rate	560	Daily Average	Period Cap Balance	Cap/Floor/Tease/NGAM	((Fully Indexed Rate - Capped Rate adjusted with Periodic Cap/Floor)* Beginning Balance)

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Period Cap Effect - Amount	570	Accrual		Cap/Floor/Tease/NGAM	(Interest Cashflow Net without cap/floor adjustment - Interest Cashflow Net with cap/floor adjustment)
Life Cap Balance	580	Daily Average		Cap/Floor/Tease/NGAM	Cur Par Balance on the cashflow date
Life Cap Effect - Rate	590	Daily Average	Life Cap Balance	Cap/Floor/Tease/NGAM	((Fully Indexed Rate - Capped Rate adjusted with Life Cap/Floor)* Beginning Balance)
Life Cap Effect - Amount	600	Accrual		Cap/Floor/Tease/NGAM	(Interest Cashflow Net without cap/floor adjustment - Interest Cashflow Net with cap/floor adjustment)
Tease Balance	610	Daily Average		Cap/Floor/Tease/NGAM	Cur Par Balance on the cashflow date
Tease Effect - Rate	620	Daily Average	Tease Balance	Cap/Floor/Tease/NGAM	((Fully Indexed Rate - Tease Rate adjusted)* Beginning Balance)

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Tease Effect - Amount	630	Accrual		Cap/Floor/Tease/NGAM	(Interest Cashflow Net without tease adjustment - Interest Cashflow Net with tease adjustment)
Neg-Am Balance	640	Daily Average		Cap/Floor/Tease/NGAM	Cur Par Balance on the cashflow date
Neg-Am Interest	650	Accrual		Cap/Floor/Tease/NGAM	
Gap Runoff	660	Sum		Standard	GAP Principal Runoff + GAP Repricing Runoff + GAP Deferred Runoff
Gap Principal Runoff	661	Sum		Runoff Detail	Principal Runoff because of Payment event across IRR Buckets
Gap Repricing Runoff	662	Sum		Repricing	Principal Runoff because of Repricing event across IRR Buckets
Gap Deferred Runoff	663	Sum		Runoff Detail	Deferred Runoff across IRR Buckets
Gap Runoff Term	670	Sum	Gap Runoff Balance	Standard	

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Gap Interest Cash Flow Gross	671	Sum		Runoff Detail	Interest Cashflow Gross across IRR Buckets
Gap Interest Cash Flow Net	672	Sum		Runoff Detail	Interest Cashflow Net across IRR Buckets
Gap Interest Cash Flow Transfer	673	Sum		Runoff Detail	Interest Cashflow Transfer across IRR Buckets
Gap Accrued Interest Gross	674	Accrual		Runoff Detail	Interest Accrued Gross across IRR Buckets
Gap Accrued Interest Net	675	Accrual		Runoff Detail	Interest Accrued Net across IRR Buckets
Gap Accrued Interest Transfer	676	Accrual		Runoff Detail	Interest Across Transfer across IRR Buckets
Gap Interest Credited	677	Sum		Runoff Detail	
Gap Runoff Gross Rate	680	Sum	Gap Runoff Balance	Standard	GAP Runoff * Current Gross Rate across IRR Buckets
Interest Accrued Gross Hist Fx Basis	681				

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Interest Accrued Net Hist Fx Basis	682				
Interest Accrued Transfer Hist Fx Basis	683				
Gap Runoff Net Rate	690	Sum	Gap Runoff Balance	Standard	GAP Runoff * Current Net Rate across IRR Buckets
Gap Runoff Transfer Rate	700	Sum	Gap Runoff Balance	Standard	GAP Runoff * Current Transfer Rate across IRR Buckets
Market Value	710	Sum		Standard	Present Value of Principal Cash Flows + Present Value of Interest Cash Flows
Present Value of Principal Cash Flows	715	Sum		Standard	Discount Factor * Total Runoff
Present Value of Interest Cash Flows	716	Sum		Standard	Discount Factor * Interest Cashflow Net
Present Value of TP Interest Cash Flows	717	Sum		Standard	Discount Factor * Interest Cashflow Transfer Rate
Duration	720				
Convexity	730				

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Discount Rate FTP	755	At Last		Standard	
Federal Taxes	930	Sum		Standard	
Local Taxes	935	Sum		Standard	
Dividends	940	Sum		Standard	
Accumulated Translation Amount	950	At Last		Standard	(FE100 Ending Balance) * [(1/ Current Exchange rate) - (1/ Original Exchange Rate)]
Discount Rate LR	1490	At Last		Standard	(Discount Rate * Present value of the cashflows) output across Liquidity Buckets
Timing of Cash Flow	1491	Sum		Standard	Days to payment * Discount Rate * Present Value of the cashflows output against Liquidity Buckets
Liquidity GAP Runoff (1661 + 1663)	1660	Sum		Standard	Liquidity GAP Total Runoff + Liquidity GAP Deferred Runoff
Liquidity GAP Total Runoff	1661	Sum		Standard	Liquidity Prepayment Runoff + Liquidity Payment Runoff + Liquidity Maturity Runoff

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Liquidity GAP Deferred Runoff	1663	Sum		Standard	Deferred Runoff across Liquidity Buckets
Liquidity GAP Option Exercise Premium	1664				
Liquidity prepayment Runoff	1666				Prepayment Runoff across Liquidity Buckets
Liquidity Payment Runoff	1667				Payment Runoff across Liquidity Buckets
Liquidity Maturity Runoff	1668				Maturity Runoff across Liquidity Buckets
Liquidity GAP Runoff Term	1670	Daily Average	Liquidity GAP Runoff	Standard	
Liquidity GAP Interest Cash Flow Gross	1671	Sum		Standard	Interest Cashflow Gross across Liquidity Buckets
Liquidity GAP Interest Cash Flow Net	1672	Sum		Standard	Interest Cashflow Net across Liquidity Buckets
Liquidity GAP Interest Cash Flow Transfer Rate	1673	Sum		Standard	Interest Cashflow Transfer Rate across Liquidity Buckets

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Liquidity GAP Accrued Interest Gross	1674	Accrual		Standard	Accrued Interest Gross across Liquidity Buckets
Liquidity GAP Accrued Interest Net	1675	Accrual		Standard	Accrued Interest Net across Liquidity Buckets
Liquidity GAP Accrued Interest Transfer Rate	1676	Accrual		Standard	Accrued Interest Transfer Rate across Liquidity Buckets
Liquidity GAP Interest Credited	1677	Sum		Standard	Interest Credited across Liquidity Buckets This is based on "Interest Credited" Flag from Prod Characteristics.
Liquidity GAP Runoff Gross Rate	1678	Daily Average	Liquidity GAP Runoff	Standard	Liquidity GAP Runoff * (Current Gross Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Liquidity GAP Runoff Net Rate	1679	Daily Average	Liquidity GAP Runoff	Standard	Liquidity GAP Runoff * (Current Net Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Liquidity GAP Runoff Transfer Rate	1680	Daily Average	Liquidity GAP Runoff	Standard	Liquidity GAP Runoff * (Current Transfer Rate + Margin) Forecasted Rate will be adjusted depending on the reprice characteristics
Liquidity Non Maturity – Core Runoff	1681				Non Maturity Core Runoff across Liquidity Buckets
Liquidity Non Maturity – Volatile Runoff	1682				Non Maturity Volatile Runoff across Liquidity Buckets
Liquidity Devolvement Runoff	1683				Devolvement Runoff across Liquidity Buckets
Liquidity Recovery Runoff	1684				Recovery Runoff across Liquidity Buckets

Financial Element Description	Financial Element Number	Averaging Type	Weighting Factor	Output Group	Formula/Detailed Description
Liquidity Non Performing Asset Runoff	1685				Non Performing Asset Runoff across Liquidity Buckets
Present Value of Cashflows - Scenariowise	1710	Sum		Standard	Liquidity GAP PV of Principal Cash Flows + Liquidity GAP PV of Interest Cash Flows
Liquidity GAP PV of Principal Cash Flows	1715	Sum		Standard	Discount Factor * Total Liquidity GAP Principal Runoff
Liquidity GAP PV of Interest Cash Flows	1716	Sum		Standard	Discount Factor * Liqudity GAP Interest Cashflow
Liquidity GAP PV of TP Int Cash Flows	1717	Sum		Standard	Discount Factor * Liqudity GAP TP Interest Cashflow

Financial Element Type - Cash Flow: Income Statement Accounts

Financial Element	Income Interest Income	Non Interest Expense	Taxes	Dividends
455 Non Interest Income	X			
457 Non Interest Expense		X		

Financial Element	Income Interest Income	Non Interest Expense	Taxes	Dividends
930 Federal Taxes			X	
935 Local Taxes			X	
940 Dividends				X

Financial Element Type - Cash Flow: Simple Accounts

Financial Element	Simple Acct Type (No TP/Adj Rates)	Simple Acct Type (No Gross Rates)	Simple Acct Type (Net Rates)	Simple Acct Type (Multi Rates)
60 Beginning Balance	X	X	X	X
70 Beginning Gross Rate	X			X
80 Beginning Net Rate	X	X	X	X
90 Beginning Transfer Rate		X		X
91 Beginning Liquidity Adjustment Rate		X		X
92 Beginning Basis Risk Cost Rate		X		X
93 Beginning Pricing Incentive Rate		X		X
94 Beginning Other Adjustment Rate		X		X
100 End Balance	X	X	X	X
110 Ending Gross Rate	X			X
120 Ending Net Rate	X	X	X	X

Financial Element	Simple Acct Type (No TP/Adj Rates)	Simple Acct Type (No Gross Rates)	Simple Acct Type (Net Rates)	Simple Acct Type (Multi Rates)
130 Ending Transfer Rate		X		X
131 Ending Liquidity Adjustment Rate		X		X
132 Ending Basis Risk Cost Rate		X		X
133 Ending Pricing Incentive Cost Rate		X		X
134 Ending Other Adjustment Rate		X		X
140 Average Bal	X	X	X	X
150 Average Gross Rate	X			X
160 Average Net Rate	X	X	X	X
170 Average Transfer Rate		X		X
174 Average Liquidity Adjustment Rate		X		X
175 Average Basis Risk Cost Rate		X		X
176 Average Pricing Incentive Rate		X		X
177 Average Other Adjustment Rate		X		X
210 Total Runoff – Positive	X	X	X	X
211 Timing of Total Runoff – Positive	X	X	X	X
212 Total Runoff – Negative	X	X	X	X

Financial Element	Simple Acct Type (No TP/Adj Rates)	Simple Acct Type (No Gross Rates)	Simple Acct Type (Net Rates)	Simple Acct Type (Multi Rates)
213 Timing of Total Runoff – Negative	X	X	X	X
220 Total Runoff Gross Rate	X			X
230 Total Runoff Net Rate	X	X	X	X
240 Total Runoff Transfer Rate		X		X
340 New Add Balance	X	X	X	X
350 New Add Gross Rate	X			X
360 New Add Net Rate	X	X	X	X
370 New Add Transfer Rate		X		X
371 New Add Liquidity Adjustment Rate		X		X
372 New Add Basis Risk Cost Rate		X		X
373 New Add Pricing Incentive Rate		X		X
374 New Add Other Adjustment Rate		X		X
411 Roll Add Liquidity Adjustment Rate		X		X
412 Roll Add Basis Risk Cost Rate		X		X
413 Roll Add Pricing Incentive Rate		X		X

Financial Element	Simple Acct Type (No TP/Adj Rates)	Simple Acct Type (No Gross Rates)	Simple Acct Type (Net Rates)	Simple Acct Type (Multi Rates)
414 Liquidity Adjustment Charge/Credit		X		X
415 Basis Risk Cost Charge/Credit		X		X
416 Pricing Incentive Charge/Credit		X		X
417 Other Adjustment Charge/Credit		X		X
418 Roll Add Other Adjustment Rate		X		X
440 Interest Accrued	X	X	X	X
445 Interest Accrued Gross	X		X	X
450 Transfer Rate Charge/Credit		X		X

Financial Element Type – Repricing Gap

Element Number	Element Name
660	Gap Runoff
661	Gap Principal Runoff
662	Gap Repricing Runoff
663	Gap Deferred Runoff
670	Gap Runoff Term
671	Gap Interest Cash Flow Gross

Element Number	Element Name
672	Gap Interest Cash Flow Net
673	Gap Interest Cash Flow Transfer
674	Gap Accrued Interest Gross
675	Gap Accrued Interest Net
676	Gap Accrued Interest Transfer
677	Gap Interest Credited
680	Gap Runoff Gross Rate
690	Gap Runoff Net Rate
700	Gap Runoff Transfer Rate

Financial Element Type – Liquidity Gap

Element Number	Element Name
1490	Discount Rate LR
1491	Timing of Cash Flow
1660	Liquidity GAP Runoff (1661 + 1663)
1661	Liquidity GAP Total Runoff
1663	Liquidity GAP Deferred Runoff
1664	Liquidity GAP Option Exercise Premium
1670	Liquidity GAP Runoff Term
1671	Liquidity GAP Interest Cash Flow Gross

Element Number	Element Name
1672	Liquidity GAP Interest Cash Flow Net
1673	Liquidity GAP Interest Cash Flow Transfer Rate
1674	Liquidity GAP Accrued Interest Gross
1675	Liquidity GAP Accrued Interest Net
1676	Liquidity GAP Accrued Interest Transfer Rate
1677	Liquidity GAP Interest Credited
1678	Liquidity GAP Runoff Gross Rate
1679	Liquidity GAP Runoff Net Rate
1680	Liquidity GAP Runoff Transfer Rate
1710	Present Value of Cashflows - Scenariowise
1715	Liquidity GAP PV of Principal Cash Flows
1716	Liquidity GAP PV of Interest Cash Flows
1717	Liquidity GAP PV of TP Int Cash Flows

Element Type - Market Value

Element Number	Element Name
490	Discount Rate - IS
491	Timing of Cash Flow (in days)
492	Discount Factor
493	Discount Factor Term (Days)

Element Number	Element Name
494	Rate Lookup Term (Days)
495	Market Value of TP Cash Flow (P+I)
497	Market Value of TP Interest Cash Flow
710	Market Value
715	Present Value of Principal Cash Flows
716	Present Value of Interest Cash Flows
717	Present Value of TP Interest Cash Flows
755	Discount Rate FTP
1490	Discount Rate LR
1491	Timing of Cash Flow
1710	Present Value of Cashflows - Scenariowise
1715	Liquidity GAP PV of Principal Cash Flows
1716	Liquidity GAP PV of Interest Cash Flows
1717	Liquidity GAP PV of TP Interest Cash Flows

Financial Element Output by Account Type

Account Types are assigned through the Dimension Management Set-Up interface for each Common COA dimension member. Product leaves other than Common COA are assigned the same account type as their associated Common COA. During Oracle ALM processing, the financial elements generated are dependent on the account type associated with each instrument record's product leaf value. The following table defines which financial elements are produced for instruments of a specific account type classification.

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
60	Beginning Balance	X	X		X	X	X				
70	Beginning Gross Rate	X	X		X						
80	Beginning Net Rate	X	X		X						
90	Beginning Transfer Rate	X	X		X						
100	End Balance	X	X		X	X	X				
110	Ending Gross Rate	X	X		X						
120	Ending Net Rate	X	X		X						
130	Ending Transfer Rate	X	X		X						
140	Average Bal	X	X		X	X	X				

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
150	Average Gross Rate	X	X		X						
160	Average Net Rate	X	X		X						
170	Average Transfer Rate	X	X		X						
180	Prepay Runoff - Positive		X		X						
181	Timing of Prepay Runoff-positive		X		X						
182	Prepay Runoff-negative		X		X						
183	Timing of Prepay Runoff - negative		X		X						
190	Payment Runoff - Positive		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
191	Timing of Payment Runoff - Positive		X		X						
192	Payment Runoff - Negative		X		X						
193	Timing of Payment Runoff - Negative		X		X						
195	Maturity Runoff-positive		X		X						
196	Timing of Maturity Runoff - Positive		X		X						
197	Maturity Runoff-negative		X		X						
198	Timing of Maturity Runoff - Negative		X		X						

FE #	Description	Auto bal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
200	Non Maturity - Core Runoff		X		X						
201	Timing of Non Maturity - Core Runoff		X		X						
202	Non Maturity - Volatile Runoff		X		X						
203	Timing of Non Maturity - Volatile Runoff		X		X						
204	Devolvement Runoff		X		X						
205	Timing of Devolvement Runoff		X		X						
206	Recovery Runoff		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
207	Timing of Recovery Runoff		X		X						
208	Non Performing Asset Runoff		X		X						
210	Total Runoff - positive		X		X						
211	Timing of Total Runoff - positive		X		X						
212	Total Runoff negative		X		X						
213	Timing of Total Runoff - negative		X		X						
220	Total Runoff Gross Rate		X		X						
230	Total Runoff Net Rate		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
240	Total Runoff Transfer Rate		X		X						
250	Repricing Balance		X		X						
255	Repricing Balance At End		X		X						
260	Before Repricing Gross Rate		X		X						
270	After Repricing Gross Rate		X		X						
280	Before Repricing Net Rate		X		X						
290	After Repricing Net Rate		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
300	Before Reprice Transfer Rate		X		X						
310	After Reprice Transfer Rate		X		X						
320	Fully Indexed Gross Rate		X		X						
330	Fully Indexed Net Rate		X		X						
340	New Add Balance		X		X		X	X	X		
350	New Add Gross Rate		X		X						
360	New Add Net Rate		X		X						
370	New Add Transfer Rate		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
380	Roll Add Balance		X		X						
390	Roll Add Gross Rate		X		X						
400	Roll Add Net Rate		X		X						
410	Roll Add Transfer Rate		X		X						
430	Interest Cash Flow	X	X	X	X						
435	Interest Cash Flow Gross	X	X	X	X						
437	Interest Cash Flow T-Rate	X	X	X	X						
440	Interest Accrued	X	X	X	X						
441	Interest Accrued Net	X	X	X	X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
445	Interest Accrued Gross	X	X	X	X						
450	Transfer Rate Charge/ Credit	X	X	X	X						
455	Non Interest Income							X			
457	Non Interest Expense								X		
480	Interest Credited		X		X						
500	WARM		X		X						
510	Annual Prepayment Rate		X		X						
515	Balance Before PrePay		X		X						
520	Deferred End Balance		X		X						

FE #	Description	Auto bal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
530	Deferred Average Balance		X		X						
540	Deferred Runoff		X		X						
550	Period Cap Balance		X		X						
560	Period Cap Effect - Rate		X		X						
570	Period Cap Effect - Amount		X		X						
580	Life Cap Balance		X		X						
590	Life Cap Effect - Rate		X		X						
600	Life Cap Effect - Amount		X		X						
610	Tease Balance		X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
620	Tease Effect - Rate		X		X						
630	Tease effect - Amount		X		X						
640	Neg-Am Balance		X		X						
650	Neg-Am Interest		X		X						
660	Gap Runoff	X	X		X	X					
661	Gap Principal Runoff	X	X		X	X					
662	Gap Repricing Runoff	X	X		X						
663	Gap Deferred Runoff	X	X		X						
670	Gap Runoff Term	X	X		X	X					

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
671	Gap Interest Cash Flow Gross	X	X	X	X			X	X		
672	Gap Interest Cash Flow Net	X	X	X	X			X	X		
673	Gap Interest Cash Flow Transfer	X	X	X	X			X	X		
674	Gap Accrued Interest Gross	X	X	X	X			X	X		
675	Gap Accrued Interest Net	X	X	X	X			X	X		
676	Gap Accrued Interest Transfer	X	X	X	X			X	X		
677	Gap Interest Credited	X	X	X							

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
680	Gap Runoff Gross Rate	X	X	X							
690	Gap Runoff Net Rate	X	X	X							
700	Gap Runoff Transfer Rate	X	X	X							
930	Federal Taxes									X	
935	Local Taxes									X	
940	Dividends										X
1660	Liquidity GAP Runoff (1661 + 1663)	X	X		X						
1661	Liquidity GAP Principal Runoff	X	X		X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
1663	Liquidity GAP Deferred Runoff	X	X		X						
1670	Liquidity GAP Runoff Term	X	X		X						
1671	Liquidity GAP Interest Cash Flow Gross	X	X	X	X						
1672	Liquidity GAP Interest Cash Flow Net	X	X	X	X						
1673	Liquidity GAP Interest Cash Flow Transfer Rate	X	X	X	X						
1674	Liquidity GAP Accrued Interest Gross	X	X	X	X						

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
1675	Liquidity GAP Accrued Interest Net	X	X	X	X						
1676	Liquidity GAP Accrued Interest Transfer Rate	X	X	X	X						
1677	Liquidity GAP Interest Credited	X	X	X							
1678	Liquidity GAP Runoff Gross Rate	X	X	X							
1679	Liquidity GAP Runoff Net Rate	X	X	X							
1680	Liquidity GAP Runoff Transfer Rate	X	X	X							

FE #	Description	Autobal Accts	Earning Asset / Interest bearing Liability	Interest Income/ Interest Expense	OBS Rec/ Pay	Equity	Other Assets/ Other Liabs	Non Int Inc	Non Int Exp	Taxes	Dividends
1710	Present Value of Cashflows -Scenario wise	X	X		X						
1715	Liquidity GAP PV of Principal Cash Flows	X	X		X						
1716	Liquidity GAP PV of Interest Cash Flows	X	X		X						
1717	Liquidity GAP PV of TP Int Cash Flows	X	X		X						

Currency Translation Methods for Financial Elements

The following table defines the translation methods used for each financial element:

Code	Financial Elements	Temporal and Current Rate Methods	Historical Basis Method
(most)	All elements not listed in the following rows.	Current bucket exchange rate should be applied to all financial elements, except those listed in the following rows.	Exchange Rate at origination should be applied to all financial elements, except those listed in the following rows.
60	Beginning Balance	Previous bucket exchange rate	Exchange rate at origination
70, 80, 90	Beginning Gross Rate, Net Rate and Transfer Rate	Previous bucket exchange rate	Exchange rate at origination
440	Interest Accrued (historical basis)	Exchange rate at origination	Exchange rate at origination
441	Interest Accrued Net (current basis)	Current bucket exchange rate	N/A
445	Interest Accrued Gross (historical basis)	Exchange rate at origination	Exchange rate at origination
446	Interest Accrued Gross (current basis)	Current bucket exchange rate	N/A
520	Deferred End Balance	Exchange rate at origination	Exchange rate at origination
530	Deferred Average Balance	Exchange rate at origination	Exchange rate at origination
540	Deferred Runoff	Exchange rate at origination	Exchange rate at origination
465	Total Currency Gain/Loss (Principal)	No translation (used only for Temporal)	N/A

Code	Financial Elements	Temporal and Current Rate Methods	Historical Basis Method
475	Realized Currency Gain/Loss (Principal)	No translation (used only for Temporal & Historical methods)	No translation
485, 486, 487	Realized Currency Gain/Loss (Interest)	No translation	No translation
950	Accumulated Translation Amount	No translation (used only for Current Rate method)	N/A
660 – 700 1660-1717	All Gap financial Elements	Calculate values for reporting currencies from translated financial elements	Exchange rate at origination

A

Cash Flow Edit Error Messages

Cash Flow Edit Error Messages

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9100	If ((AMRT_TYPE_CD <> 700 or (AMRT_TYPE_CD = 700 and AMRT_TERM <> 0)) and AMRT_TERM (in days) < ORG_TERM (in days)	Set AMRT_TERM equal to ORG_TERM, AMRT_TERM_MULT = ORG_TERM_MULT	2	Org Term > Amrt Term	Amortization term can only be equal to zero on Non/Amortizing instruments	
9102	CUR_GROSS_RATE < 0	message only: Current Gross Rate < 0	1	Current Gross Rate < 0	Current gross rate must not be negative	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9103	CUR_NET_RATE < 0	message only: Current Net Rate < 0	1	Current Net Rate < 0	Current net rate must not be negative	
9104	ACCRUAL_BASIS_CD <1 or >6	ACCRUAL_BASIS_CD equal to '3'.	2	Invalid Accrual Basis Cd	Accrual basis code must be between 1 and 6 inclusively	
9105	AMRT_TYPE_CD (<= 999 and not equal to 100, 200, 400, 500,600, 700, 710, 800, 801, 802, 820,830,840, 850 or 999) or >99999	Set AMRT_TYPE_CD equal to '700'.	2	Invalid Amortization Type, defaulted to 700	Checks that amortization type codes are either in the user defined range or that they are a valid OFSA code.	
9106	AMRT_TERM_MULT not equal to D, M, or Y	AMRT_TERM_MULT = 'M'	2	Invalid AMRT_TERM_MULT	Checks for valid amortization term multipliers	
9107	NEG_AMRT_EQ_MULT not equal to D, M, or Y	NEG_AMRT_EQ_MULT = 'M'.	2	Invalid NEG_AMRT_EQ_MULT	Checks for valid Neg.Amortization Eq.Multipliers	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9108	(NEG_AMRT_LIMIT >=200)or(NEG_AMRT_LIMIT < 0) and AMRT_TYPE_CD = 600	Set NEG_AMRT_LIMIT equal to 0.	2	Invalid NgAm Limit	Neg Am limit value does not fall in a valid range.(Neg Am instruments only)	
9109	ORG_TERM_MULT not equal to D, M, or Y	ORG_TERM_MULT = 'M'	2	Invalid ORG_TERM_MULT	Checks for valid Original term multipliers	
9110	If AMRT_TYPE_CD = 600 and PMT_CHG_FREQ < 0	PMT_CHG_FREQ equal to 0	2	Invalid Pmt Chg Frq	frequency cannot be negative (Neg Am isntrument only)	
9111	PMT_CHG_FREQ_MULT not equal to D, M, or Y	PMT_CHG_FREQ_MULT equal to 'M'.	2	Invalid PMT_CHG_FREQ_MULT	Checks for valid payment change frequency multipliers	
9112	PMT_FREQ_MULT not equal to D, M, or Y	PMT_FREQ_MULT = 'M'	2	Invalid PMT_FREQ_MULT	Checks for valid payment frequency multipliers	
9114	RATE_CHG_RND_CD <0 or >4	RATE_CHG_RND_CD equal to '0'.	2	Invalid Rate Chg Rnd Cd	Rate change round code must be between 0 and 4	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9115	RATE_SET_LAG_MULT not equal to D, M, or Y	RATE_SET_LAG_MULT equal to 'M'.	2	Invalid RATE_SET_LAG_MULT	Checks for valid rate set lag multipliers	
9117	REPRICE_FREQ_MULT not equal to D, M, or Y	REPRICE_FREQ_MULT = 'M'.	2	Invalid REPRICE_FREQ_MULT	Checks for valid reprice frequency multipliers	
9118	RATE_CHG_RND_FAC < 0 or RATE_CHG_RND_FAC > 1	Set RATE_CHG_RND_FAC equal to 0.	2	Invalid Rt Chg Rnd Fact	Rate change round factor must be a percentage	
9119	MATURITY_DATE < NEXT_PAYMENT_DATE	MATURITY_DATE = NEXT_PAYMENT_DATE + ((REMAIN_NO_PMTS_C -1)*PMT_FREQ)	2	Mat Dt < Next Pmt Dt	Maturity date can not be before the next payment date	
9120	NEG_AMRT_EQ_FREQ < 0 and AMRT_TYPE_CD = 600	NEG_AMRT_EQ_FREQ equal to 0.	2	NegAmEqFreq < 0	Neg Am Equalization frequency cannot be negative (Neg Am instruments only)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9121	If AMRT_TYPE_CD = 600 and NEG_AMRT_EQ_FREQ > 0 and NEG_AMRT_EQ_DATE <= MAX(ORIGINATION_DATE,AS_OF_DATE)	Set NEG_AMRT_EQ_DATE equal to NEXT_REPRICE_DATE	2	NgAm Eq Dt < Org Dt	Neg Am equalization date is less than origination date (future origination) or less than the as-of-date (past origination) (Neg am instruments only)	
9123	AS_OF_DATE > ORIGINATION_DATE and AS_OF_DATE >= NEXT_PAYMENT_DATE	NEXT_PAYMENT_DATE equal to AS_OF_DATE + 1 day.	2	Next Pmt Dt < As of Dt	Next payment date is less than as-of-date (past origination case) As of Date can not be greater than the origination date and greater than the next payment date.	
9124	ORIGINATION_DATE >= AS_OF_DATE and ORIGINATION_DATE >= NEXT_PAYMENT_DATE	NEXT_PAYMENT_DATE = ORIGINATION_DATE + 1 day	2	Next Pmt Dt < Org Dt	Next payment date is less than origination date (future origination case)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9125	ORIGINATION_DATE <= AS_OF_DATE and NEXT_REPRICE_DATE <= AS_OF_DATE and REPRICE_FREQ > 0	NEXT_REPRICE_DATE equal to AS_OF_DATE + 1 day.	2	Next Repr Dt < As of Dt	Next reprice date is less than as-of-date (past origination case)	
9126	ORIGINATION_DATE > AS_OF_DATE and NEXT_REPRICE_DATE < ORIGINATION_DATE and REPRICE_FREQ > 0	Set NEXT_REPRICE_DATE equal to ORIGINATION_DATE + 1 day	2	Next Repr Dt < Org Dt	Next reprice date is less than the origination date (future origination case)	
9127	ORG_PAYMENT_AMT = 0 and PMT_DECR_LIFE > 0	Set PMT_DECR_LIFE equal to 0.	2	Org Pmt=0, Pmt Dec Lf<>0	Payment decrease life is expressed as a percent of a zero original payment. (Neg Am instruments only)	
9128	ORG_PAYMENT_AMT = 0 and PMT_INCR_LIFE > 0	Set PMT_INCR_LIFE equal to 0.	2	Org Pmt=0, Pmt Incr Lf<>0	Payment increase life is expressed as a percent of a zero original payment.(Neg Am instruments only)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9129	ORG_TERM <> calculated original term within 45 days or ORG_TERM = 0	Set ORG_TRM = calculated original term	2	Org Term<> Mat Dt - Org Dt	Original term should equal the time between the origination date and the maturity date.	
9130	(ORIGINATION_DATE < 01/01/1950) or (ORIGINATION_DATE > 01/01/2099)	Set ORIGINATION_DATE = 01/01/1950.	2	Orig. Dt < 01/01/1950	Origination date must be acceptable	
9131	PMT_FREQ > calculated original term	PMT_FREQ equal to calculated original term	2	Payment Freq > Org Term	Payment frequency cannot be greater than original term.	
9132	(CUR_PAYMENT < 0 and CUR_PAR_BAL > 0) or (CUR_PAYMENT > 0 and CUR_PAR_BAL < 0)	Set CUR_PAYMENT equal to 0.	2	Pmt, bal opposite signs	Current payment and current balance can not have opposite signs	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9133	If AMRT_TYPE_CD = 600 and PMT_CHG_FREQ > 0 and AS_OF_DATE > PMT_ADJUST_DATE	PMT_ADJUST_DATE = NEXT_REPRICE_DATE	2	Pmt Adj Dt < As of Dt	Neg Am equalization date is less than the as-of-date (past origination) (Neg am instruments only	
9134	If AMRT_TYPE_CD = 600 and PMT_CHG_FREQ > 0 and AS_OF_DATE < PMT_ADJUST_DATE and PMT_ADJUST_DATE < ORIGINATION_DATE	PMT_ADJUST_DATE = NEXT_REPRICE_DATE	2	Pmt Adj Dt < Org Dt	Payment adjustment date is less than origination date (future origination) (Neg am instruments only)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9135	PMT_FREQ <= 0 and ORIGINATION_DATE <= AS_OF_DATE and MATURITY_DATE > AS_OF_DATE and ORIGINATION_DATE > AS_OF_DATE and MATURITY_DATE > ORIGINATION_DATE	NEXT_PAYMENT_DATE = MATURITY_DATE and ORG_TERM = calculated original term and PMT_FREQ = calculated original term and REMAIN_NO_PMTS_C = 1	2	Pmt Freq <= 0, Trm Assumed	Payment frequency is less than or equal to zero, and both maturity date and origination date are valid dates and can be used to calculate payment frequency.	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9135	PMT_FREQ <= 0 and ORIGINATION_DATE <= AS_OF_DATE, MATURITY_DATE <= AS_OF_DATE but NEXT_PAYMENT_DATE > AS_OF_DATE or ORIGINATION_DATE > AS_OF_DATE, MATURITY_DATE < ORIGINATION_DATE ,but NEXT_PAYMENT_DATE > ORIGINATION_DATE	MATURITY_DATE = NEXT_PAYMENT_DATE and ORG_TERM = calculated original term and PMT_FREQ = calculated original term and REMAIN_NO_PMTS_C = 1	2	Pmt Freq <= 0 ,Trm Assumed	Payment frequency is less than or equal to zero and maturity date is invalid, but next payment date can be used to calculate a valid payment frequency	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9135	PMT_FREQ <= 0 and ORIGINATION_DATE <= AS_OF_DATE, MATURITY_DATE <= AS_OF_DATE, and NEXT_PAYMENT_DATE <= AS_OF_DATE or ORIGINATION_DATE > AS_OF_DATE, and both MATURITY_DATE and NEXT_PAYMENT_DATE less than ORIGINATION_DATE	MATURITY DATE = AS OF DATE + 1 day and NEXT PAYMENT DATE = AS OF DATE + 1 day and ORG TERM = calculated original term and PMT_FREQ = calculated original term and REMAIN_NO_PMTS_C = 1.	2	Pmt Freq <= 0, Trm Assumed	Payment frequency less than or equal to zero and all dates which can be used to calculate payment frequency are in the past	
9136	PMT_INCR_CYCLE < 0	Set PMT_INCR_CYCLE equal to 0.	2	Pmt Incr Cycle < 0	Payment increase cycle cannot be less than zero (Neg Am instruments only)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9138	RATE_CAP_LIFE < CUR_GROSS_RATE and RATE_CAP_LIFE <> 0 and CUR_GROSS_RATE <> 0 and TEASER_END_DATE < AS_OF_DATE	Set RATE_CAP_LIFE equal to CUR_GROSS_RATE.	1	Rt Cap Lf < Cur Gross Rt	Current gross rate is greater than the rate cap life	
9139	RATE_CAP_LIFE < CUR_NET_RATE and RATE_CAP_LIFE <> 0	message only: Rt Cap Life < Cur Net Rt	1	Rt Cap Life < Cur Net Rt	Current net rate is greater than the rate cap	
9140	RATE_CHG_MIN < 0	Set RATE_CHG_MIN equal to 0.	2	Rate Chg Min < 0	Minimum rate change can not be negative	
9141	RATE_DECR_CYCLE < 0	RATE_DECR_CYCLE equal to 0.	2	Rate Decr Cycle < 0	Rate decrease cycle must not be negative	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9143	RATE_FLOOR_LIFE > CUR_GROSS_RATE and CUR_GROSS_RATE <> 0 and TEASER_END_DATE < AS_OF_DATE	Set RATE_FLOOR_LIFE equal to CUR_GROSS_RATE.	1	Rt Floor Lf > Cur Grss Rt	Current gross rate is less than the rate floor	
9144	RATE_FLOOR_LIFE > CUR_NET_RATE	message only: Rt Floor Lf > Cur Net Rt	1	Rt Floor Lf > Cur Net Rt	Rate floor life must not be greater than the current net rate	
9145	RATE_INCR_CYCLE < 0	RATE_INCR_CYCLE equal to 0.	2	Rate Incr Cycle < 0	Rate increase cycle can not be less than 0	
9147	REMAIN_NO_PMTS_C < 1	REMAIN_NO_PMTS_C = 1	2	Rem No Pmts < 1	There has to be at least 1 payment left	
9148	RATE_SET_LAG < 0	RATE_SET_LAG = 0	2	Set Lag < 0	Rate set lag can not be negative	
9152	ORG_PAR_BAL < CUR_PAR_BAL and AMRT_TYPE_CD =710	message only: OrgParBal < CurParBal	1	Org Par Bal < Cur Par Bal	Original balance on rule of 78's instruments should not be greater than current balance	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9153	REPRICE_FREQ = 0 and (AMRT_TYPE_CD = 500 or AMRT_TYPE_CD = 600)	message only: AdjAmrtType, ReprFrq= 0	1	AdjAmrtType, ReprFrq=0	Reprice frequency denotes fixed on adjustable amortization types (Conventional Adjustable oand Adjustable Neg Am)	
9154	REPRICE_FREQ <> 0 and LAST_REPRICE_DATE > NEXT_REPRICE_DATE	LAST_REPRICE_DATE = NEXT_REPRICE_DATE minus REPRICE_FREQ	2	LastReprDt>NextReprDt	Last reprice date is greater than next reprice date	
9155	IF INTEREST_RATE_CD < 0 or INTEREST_RATE_CD > 99999	INTEREST_RATE_CD = 0	2	Int Rt Code out of range	Interest rate code must be within a valid range	
9157	T_RATE_INT_RATE_CD < 0 or T_RATE_INT_RATE_CD > 99999	T_RATE_INT_RATE_CD = 99999	2	T_ Rt Int Rt Cd out of rng	T rate interest rate code must be within a valid range.	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9158	If CUR_BOOK_BAL <> (CUR_PAR_BAL + DEFERRED_CUR_BAL)	Warning Only	1	CurBkBl <> ParBl + Def Bl	Current book balance should equal the current par balance plus the deferred current balance	
9159	CUR_NET_PAR_BAL_C - CUR_PAR_BAL * (1 - PERCENT_SOLD/100) >.001	Warning	1	NetParBl<>Com ptd NetParBl	Current net par balance should reflect the bank-owned portion of the current gross balance	
9160	If AMRT_TYPE_CD = 600 and PMT_DECR_CY < 0	PMT_DECR_CY = 0	2	Payment Decrease Cycle =0	Payment decrease cycle cannot be less than zero (Neg Am instruments only)	
9161	If AMRT_TYPE_CD = 600 and PMT_DECR_LF < 0	Set PMT_DECR_LIFE equal to 0.	2	PmtDecrLf<0 for Adj NegAm	Payment decrease life cannot be less than zero (Neg Am instruments only)	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9162	If AMRT_TYPE_CD = 600 and PMT_DECR_LF <> 0 and CUR_PAYMENT < ORG_PAYMENT_AMT * (1 - PMT_DECR_LF/100)	Warning Only	1	Cur Pmt < Life Pay Floor	Current payment is less than the minimum payment amount (Neg Am instruments only)	
9163	If AMRT_TYPE_CD = 600 and PMT_INCR_LF < 0	Set PMT_INCR_LIFE equal to 0.	2	PmtIncrLf<0 for Adj NegAm	Payment increase life cannot be less than zero (Neg Am instruments only)	
9164	If AMRT_TYPE_CD = 600 and PMT_INCR_LF <> 0 and CUR_PAYMENT > ORG_PAYMENT_AMT * (1 + PMT_INCR_LF/100)	Warning Only	1	Cur Pmt > Life Pay Cap	Current payment is greater than the maximum payment amount (Neg Aminstruments only)	
9165	If ISSUE_DATE > ORIGINATION_DATE	Set ISSUE_DATE = ORIGINATION_DATE	2	Issue Date > Orig Date	Issue date can not be greater than origination date	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9166	If REPRICE_FREQ < 0	REPRICE_FREQ = 0	2	Reprice_Freq < 0	Reprice frequencies must not be negative	
9167	If AMRT_TYPE_CD = 710 and REPRICE_FREQ < > 0	REPRICE_FREQ = 0	2	Rule of 78s Reprice_Freq<> 0	Rule of 78's instruments are implicitly fixed.	
9168	ORG_PAR_BAL = 0 and REPRICE_FREQ = 0	Warning	1	OrgPar Bal=0! Fixed Rate	For transfer pricing of fixed rate instruments, the original balance should be populated,	
9169	If REPRICE_FREQ <> 0 and TEASER_END_DATE > ORIGINATION_DATE and TEASER_END_DATE > AS_OF_DATE and NEXT_REPRICE_DATE > TEASER_END_DATE	Set NEXT_REPRICE_DATE equal to TEASER_END_DATE	2	Next Repr Dt > Ts End Dt	Next reprice date is greater than tease end date.	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9170	If REMAIN_NO_PMTS_C + (REMAIN_TERM_C (in days) / REPRICE_FREQ (in days)) > 2000	Warning	1	>2000 Events for Record	The maximum number of events that can be modeled has been exceeded.	
9171	LRD_BALANCE=0	LRD_BALANCE = CUR_PAR_BAL	2	LRDBalance=0	The balance as of the last reprice date cannot be equal to 0 **Note:** If Cur_Par_Bal = 0, then it would not be processed by the cash flow engine as error.	
9172	ADJUSTABLE_TYPE_CD <> 0 and LAST_REPRICE_DATE < ISSUE_DATE	Message Only	1	LastRepriceDate < Issue Date	When the last reprice date is less than the issue date, transfer pricing will not occur.	
9174	REPRICE_FREQ > 0 and ADJUSTABLE_TYPE_CD = 0	Message Only	1	AdjType=0 & ReprFreq>0	Reprice frequency and adjustable type code are inconsistent	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9175	REPRICE_FREQ = 0 and ADJUSTABLE_TYPE_CD <> 0	Message Only	1	AdjType<>0 & ReprFreq=0	Reprice frequency and adjustable type code are inconsistent	
9176	AMRT_TYPE_CD = 100 and ADJUSTABLE_TYPE_CD <> 0	Message Only	1	AdjType<>0 & AmrtType=100	Variable adjustable type on Conventional Fixed instrument.	
9177	AMRT_TYPE_CD = 710 and ADJUSTABLE_TYPE_CD <> 0	ADJUSTABLE_TYPE_CD =0	2	AdjType<>0 & AmrtType=710	Rule of 78's instrumnet should only have a fixed adjustable type code.	
9178	(AMRT_TYPE_CD = 500 or AMRT_TYPE_CD = 600) and ADJUSTABLE_TYPE_CD = 0	Message Only	1	AdjType=0 & Amrt(500,600)	Fixed adjustable type code on adjustable amortization codes (Conventional Adjustable and Adjustable Neg Am)	
9179	LAST_PAYMENT_DATE > NEXT_PAYMENT_DATE and PAYMENT_FREQ > 0	LAST_PAYMENT_DATE =NEXT_PAYMENT_DATE -PMT_FREQ	2	LastPayDate>=NextPayDate	Last payment date is greater than next payment date and can be calculated using payment frequency	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9179	LAST_PAYMENT_DATE > NEXT_PAYMENT_DATE and PAYMENT_FREQ <= 0	LAST_PAYMENT_DATE =ORIGINATION_DATE	2	LastPayDate>=NextPayDate	Last payment date is greater than next payment date, but cannot be calculated using payment frequency.	
9180	LAST_PAYMENT_DATE < ORIGINATION_DATE	LAST_PAYMENT_DATE =ORIGINATION_DATE	2	LastPayDate < OrigDate	Last payment date cannot be less than the origination date	
9181	LAST_PAYMENT_DATE > AS_OF_DATE and ORIGINATION_DATE <= AS_OF_DATE	LAST_PAYMENT_DATE =AS_OF_DATE	2	LastPayDate>AsOfDate	Last payment date cannot be greater than the as-of-date if the instrument originated in the past.	
9182	INT_TYPE = 2 and AMRT_TYPE CD = 100, 400, 500, 600 or 800	INT_TYPE = 1	2	IntType<>1,2 /AmrtType	Interest type can only be arrears for conventionally amortizing instruments.	
9182	INT_TYPE <> 1 or 2	INT_TYPE = 1	2	IntType<>1,2 /AmrtType	Interest type must be a valid OFSA code.	

Error Code	Error Condition	ASSIGNMENT	Error Level	Warning	Description/Purpose	Edit Priority
9183	COMPOUND_BASIS_CD <> 110, 120, 130, 140, 150, 160, 170 or 200	COMPOUND_BASIS_CD = 160	2	Invalid CompBasisCode	Compounding basis code must be a valid OFSA code	
9184	(ACCRUAL_BASIS_CD = 1, 4 or 5) and (PMT_FREQ_MULT = D or AMRT_TYPE_CD = 800,801, or 802)	ACCRUAL_BASIS_CD = 3	2	AmrtType/AccrlBasis Error	Accrual basis code cannot have a 30 day month assumption on instruments defined by a payment schedule	

Note: The error level for the Cash Flow Edit rules:

Severity 0 is a warning, Severity 1 is Significant (but a default may be used while processing), and Severity 2 is Critical and the processing of the record may be skipped.

Glossary

Assumption rules

This term refers to a family of business rules you use to create underlying assumptions to run Oracle Asset Liability Management (ALM) and Funds Transfer Pricing processes.

Bulk Processing

OFSAA processing implemented with SQL statements that affect multiple rows is identified as Bulk Processing. Oracle Financial Services Analytical Applications (OFSAA) uses bulk processing in situations where a large number of records are updated using a single SQL statement. A single OFSAA process can use more than one bulk SQL statement to complete the process.

Bulk processing is generally database bound.

Cash Flow Column

A Cash Flow column is an instrument table column in the OFSAA data model used for storing specific characteristics that dictate the cash payments and receipts for an account. Cash Flow columns are required for any OFSAA processing that needs to generate cash payment and receipt events for instrument accounts.

Functional Area

A functional area is a set of tables and processes used to satisfy a business or process requirement. Each of the functional areas identified in the *Oracle Financial Services Cash Flow Engine Reference Guide* is a set of tables used to implement a feature or business rule in OFSA. For example, the tables in the Leaves and Hierarchies functional area are used to identify Leaf Nodes and Tree Rollups.

Instrument Table

Instrument tables in the OFSAA database contain account level financial data.

Cash Flow Engine

The OFSAA Cash Flow Engines provide the means by which additional value-added information is created within the database. Information is retrieved from the database and processed based upon assumptions specified by the user.

Key Leaf (dimension) Column

A Key Leaf column is a special column used for categorizing data within the database. Leaf values are the lowest level of designation within a Chart of Accounts.

For example, the Organizational Unit leaf column is provided with the initial installation of the OFSAA database. This column is used to categorize accounts by certain Organizational designations/locations within an institution. Accounts would be assigned (or would originate from) individual Organizational units. Each Organizational Unit would be represented by a leaf value for that column.

Leaf values can be hierarchically organized using an OFSAA Hierarchy.

The standard leaf columns provided with the initial installation are Financial Element, Organizational Unit, General Ledger Chart of Accounts, Product and Common Chart of Accounts. Additional user-defined Key Leaf columns can be created if needed.

LEDGER_STAT table

The LEDGER_STAT table is the data source for summary financial and statistical data used by OFSAA.

OFSAA

OFSAA is an abbreviation for Oracle Financial Services Analytical Applications. OFSAA refers generically to the system that comprises the products in the Oracle Financial Services suite.

OFSAA Reserved Table

OFSAA Reserved tables are provided with the installation of the database and cannot be customized or altered. In addition, Oracle recommends that the data in these tables not be updated or changed outside of the OFSAA interface. To update or change the data in these tables outside of the OFSAA interface may cause OFSAA to function incorrectly.

Product

A product is one applications of the OFSAA suite. Each of the OFSAA products provides a set of business functions. OFSAA is comprised of the following products:

- Oracle Financial Services Analytical Applications Infrastructure (AAI)
- Oracle Financial Services Asset Liability Management
- Oracle Financial Services Asset Liability Management Analytics
- Oracle Financial Services Balance Sheet Planning
- Oracle Financial Services Funds Transfer Pricing

- Oracle Financial Services Hedge Management and IFRS Valuations
- Oracle Financial Services Pricing Management – Transfer Pricing Component
- Oracle Financial Services Profitability Management
- Oracle Financial Services Profitability Analytics

Portfolio Field

A Portfolio field is a field that is common to all financial instruments.

Portfolio Instrument Table

A Portfolio Instrument is a table defined with all required instrument columns and all cash flow columns. By definition, all portfolio fields must be included on a Portfolio Instrument Table.

Row-by-row Processing

OFSAA processing implemented by evaluating one record at a time is identified as Row by Row Processing. OFSAA uses row by row processing for situations where a number of complex calculations are executed for each record. For example, OFSAA uses row by row processing when generating cash flows for individual accounts during Oracle ALM processing.

Row by row processing is generally OFSAA (memory) bound.

Hierarchy

Hierarchies can be defined based on key leaf columns. For example, an Organizational hierarchy showing the relationships of all Organizational Units and any summary nodes can be created for the Organizational Unit leaf using the Hierarchy Management user interface.

OFSAA allows any number of Hierarchies to be created within the database.