

# **Oracle® Banking Platform Collections**

Interface Specification Guide

Release 2.3.0.0.0

**E56276-01**

July 2014

Copyright © 2011, 2014, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

---

---

# Contents

|  |      |
|--|------|
| <b>Preface</b> .....                           | ix   |
| Audience .....                                 | ix   |
| Documentation Accessibility .....              | ix   |
| Organization of the Guide .....                | ix   |
| Related Documents .....                        | x    |
| Conventions .....                              | x    |
| <br>   |      |
| <b>1 Introduction</b>                          |      |
| <br>   |      |
| <b>2 System Overview</b>                       |      |
| <br>   |      |
| <b>3 Staging Area</b>                          |      |
| 3.1 Feeder Tables .....                        | 3-1  |
| 3.1.1 Account Data .....                       | 3-1  |
| 3.1.1.1 Account Details .....                  | 3-1  |
| 3.1.1.2 Account Arrears Details .....          | 3-7  |
| 3.1.1.3 Account Hardship Details .....         | 3-8  |
| 3.1.1.4 Account Repayment Schedule .....       | 3-9  |
| 3.1.1.5 Account Warning Indicator .....        | 3-10 |
| 3.1.2 Party Data .....                         | 3-10 |
| 3.1.2.1 Party Account Relationship .....       | 3-10 |
| 3.1.2.2 Party Details .....                    | 3-12 |
| 3.1.2.3 Party Address Details .....            | 3-14 |
| 3.1.2.4 Party Employment Details .....         | 3-15 |
| 3.1.2.5 Party Identification Details .....     | 3-17 |
| 3.1.2.6 Party Name Details .....               | 3-18 |
| 3.1.2.7 Party Contact Preference Details ..... | 3-18 |
| 3.1.2.8 Party Warning Indicators .....         | 3-20 |
| 3.1.3 Collateral Data .....                    | 3-20 |
| 3.1.3.1 Collateral Details .....               | 3-20 |
| 3.1.3.2 Collateral Charge Details .....        | 3-22 |
| 3.1.3.3 Collateral Entity Mapping .....        | 3-22 |
| 3.1.3.4 Collateral Guarantor Mapping .....     | 3-23 |
| 3.1.3.5 Collateral Owner Mapping .....         | 3-23 |
| 3.1.4 Insurance Data .....                     | 3-24 |

|         |                                |      |
|---------|--------------------------------|------|
| 3.1.4.1 | Insurance Details .....        | 3-24 |
| 3.1.5   | Payment Data .....             | 3-25 |
| 3.1.5.1 | Online Payment Records .....   | 3-25 |
| 3.2     | OBP Views .....                | 3-26 |
| 3.2.1   | Main Account Views .....       | 3-26 |
| 3.2.2   | Account Updateable Views ..... | 3-27 |
| 3.2.3   | Hardship Views .....           | 3-27 |
| 3.2.4   | Party Views.....               | 3-27 |
| 3.2.5   | LCM / Collateral Views .....   | 3-27 |

## 4 Algorithms

|      |  |      |
|------|--|------|
| 4.1  | Stop Contract: C1-CURENTITY.....                                   | 4-1  |
| 4.2  | Cure Account: C1-FINCOLL.....                                      | 4-1  |
| 4.3  | Queue Allocation: C1-ALLOCQUEU .....                               | 4-2  |
| 4.4  | Update Customer Switch: C1-CUSTSW .....                            | 4-3  |
| 4.5  | Update Legal/Repo Switch: C1-LEREPOCT.....                         | 4-4  |
| 4.6  | User Allocation - Round Robin: C1-USRALCRR .....                   | 4-4  |
| 4.7  | User Allocation - % Based: C1-USRALCPR.....                        | 4-5  |
| 4.8  | Vendor Allocation - Round Robin: C1-VENALCRR .....                 | 4-6  |
| 4.9  | Vendor Allocation - % Based: C1-VENALCPR .....                     | 4-8  |
| 4.10 | Bulk Contact Creation: C1-BLKCNTCRE.....                           | 4-9  |
| 4.11 | Cross Strategy Action Matrix: C1-CSAM.....                         | 4-10 |
| 4.12 | Last Payment for Account: C1-PAYDTAMTU .....                       | 4-11 |
| 4.13 | Association Review Check: C1-ASORVCHK .....                        | 4-11 |
| 4.14 | Validate Expired Default Notice: C1-DEFNOEXP.....                  | 4-12 |
| 4.15 | Associate Related Entity: C1-ASSOENTY .....                        | 4-13 |
| 4.16 | Validate Legal Case Exists: C1-CHKLGL.....                         | 4-14 |
| 4.17 | Assign New LSP: C1-ASGNLSP .....                                   | 4-15 |
| 4.18 | Check Approval Requirement: C1-APPRCHK .....                       | 4-17 |
| 4.19 | Save the Status Before Change LSP: C1-SAVESTATUS.....              | 4-18 |
| 4.20 | Resume Status from Previous LSP: C1-RESSTATUS.....                 | 4-18 |
| 4.21 | Check Submission Date: C1-CHKSUBDT1 .....                          | 4-18 |
| 4.22 | Update LSP (CLOS): C1-LSPSTATUS.....                               | 4-19 |
| 4.23 | Update LSP (CANCEL): C1-LSPSTACAN.....                             | 4-20 |
| 4.24 | Validate Expired Default Notice: C1-DEFNOTEXP .....                | 4-20 |
| 4.25 | Collateral Verification: C1-VRFYCOLS .....                         | 4-22 |
| 4.26 | Account Association for Asset Repossession Case: C1-ARSACCTS.....  | 4-22 |
| 4.27 | Customer Association for Asset Repossession Case: C1-ARSCUSTS..... | 4-23 |
| 4.28 | Update Collateral Property: C1-SETCOLLPR .....                     | 4-23 |
| 4.29 | Close To do's Algorithm: C1-CLSTODOA .....                         | 4-24 |
| 4.30 | Validations for Mandatory Characteristics: C1-CHARVALU .....       | 4-26 |
| 4.31 | Validations for Mandatory Characteristics: C1-CHARVALA .....       | 4-27 |
| 4.32 | Update Collateral Status in the Host: C1-UPCOLLSTZ.....            | 4-29 |
| 4.33 | Initiate Collateral Valuation: C1-COLLVALX .....                   | 4-30 |
| 4.34 | Close To do's Algorithm: C1-CLSTODOV .....                         | 4-32 |
| 4.35 | Validations for Mandatory Characteristics: C1-CHARVALC.....        | 4-34 |
| 4.36 | Update Collateral Status in the Host: C1-UPCOLLSTX.....            | 4-35 |

|      |   |      |
|------|---|------|
| 4.37 | Validation Settlement: C1-VALSET .....                  | 4-36 |
| 4.38 | Initiate LMI Process: C1-INITLMI.....                   | 4-37 |
| 4.39 | Close To do's Algorithm: C1-CLSTODO.....                | 4-39 |
| 4.40 | Update Collateral Property: C1-RESETCOLL .....          | 4-40 |
| 4.41 | Update Collateral Status in the Host: C1-UPCOLLSTY..... | 4-41 |
| 4.42 | PTP Active Algorithm: C1-PTPACTIVE.....                 | 4-42 |
| 4.43 | PTP Active Algorithm: C1-PTPKEPT .....                  | 4-44 |
| 4.44 | PTP Active Algorithm: C1_PTPBRKLS .....                 | 4-46 |
| 4.45 | Rule facts populating algorithm: C1-BRLSR.....          | 4-48 |

## 5 Feeder Services

## List of Tables

|      |  |      |
|------|--|------|
| 3-1  | Account Details .....                                  | 3-1  |
| 3-2  | Account Arrears Details.....                           | 3-7  |
| 3-3  | Account Hardship Details .....                         | 3-8  |
| 3-4  | Account Repayment Schedule .....                       | 3-9  |
| 3-5  | Account Warning Indicator .....                        | 3-10 |
| 3-6  | Account Party Relationship.....                        | 3-11 |
| 3-7  | Party Details.....                                     | 3-12 |
| 3-8  | Party Address Details.....                             | 3-14 |
| 3-9  | Party Employment Details.....                          | 3-16 |
| 3-10 | Party Identification Details .....                     | 3-17 |
| 3-11 | Party Name Details.....                                | 3-18 |
| 3-12 | Party Contact Preference Details .....                 | 3-19 |
| 3-13 | Party Warning Indicators.....                          | 3-20 |
| 3-14 | Collateral Details.....                                | 3-21 |
| 3-15 | Collateral Charges Details .....                       | 3-22 |
| 3-16 | Collateral Entity Mapping .....                        | 3-22 |
| 3-17 | Collateral Guarantor Mapping .....                     | 3-23 |
| 3-18 | Collateral Owner Mapping.....                          | 3-23 |
| 3-19 | Insurance Details.....                                 | 3-24 |
| 3-20 | Online Payment.....                                    | 3-25 |
| 4-1  | Stop Contract: C1-CURENTITY .....                      | 4-1  |
| 4-2  | Cure Account: C1-FINCOLL.....                          | 4-1  |
| 4-3  | Cure Account: Sample Algorithm .....                   | 4-2  |
| 4-4  | Queue Allocation: C1-ALLOCQUEUE .....                  | 4-2  |
| 4-5  | Queue Allocation: Sample Algorithm.....                | 4-3  |
| 4-6  | Update Customer Switch: C1-CUSTSW .....                | 4-3  |
| 4-7  | Update Customer Switch: Sample Algorithm .....         | 4-3  |
| 4-8  | Update Legal/Repo Switch: C1-LEREPOCT .....            | 4-4  |
| 4-9  | Update Legal/Repo Switch: Sample Algorithm .....       | 4-4  |
| 4-10 | User Allocation - Round Robin: C1-USRALCRR .....       | 4-5  |
| 4-11 | User Allocation - % Based: C1-USRALCPR .....           | 4-5  |
| 4-12 | Vendor Allocation - Round Robin: C1-VENALCRR .....     | 4-6  |
| 4-13 | Vendor Allocation - % Based: C1-VENALCRR .....         | 4-8  |
| 4-14 | Bulk Contact Creation: C1-BLKCNTCRE.....               | 4-10 |
| 4-15 | Cross Strategy Action Matrix: C1-CSAM .....            | 4-10 |
| 4-16 | Cross Strategy Action Matrix: Sample Algorithm .....   | 4-11 |
| 4-17 | Last Payment for Account: C1-PAYDTAMTU .....           | 4-11 |
| 4-18 | Association Review Check: C1-ASORVCHK .....            | 4-11 |
| 4-19 | Association Review Check: Sample Algorithm.....        | 4-12 |
| 4-20 | Validate Expired Default Notice: C1-DEFNOEXP.....      | 4-12 |
| 4-21 | Validate Expired Default Notice: Sample Algorithm..... | 4-13 |
| 4-22 | Associate Related Entity: C1-ASSOENTY .....            | 4-13 |
| 4-23 | Associate Related Entity: Sample Algorithm.....        | 4-14 |
| 4-24 | Validate Legal Case Exists: C1-CHKLGL.....             | 4-14 |
| 4-25 | Validate Legal Case Exists: Sample Algorithm .....     | 4-15 |
| 4-26 | Assign New LSP: C1-ASGNLSP .....                       | 4-15 |
| 4-27 | Assign New LSP: Sample Algorithm.....                  | 4-16 |
| 4-28 | Check Approval Requirement: C1-APPRCHK .....           | 4-17 |
| 4-29 | Check Approval Requirement: Sample Algorithm.....      | 4-17 |
| 4-30 | Save the Status Before Change LSP: C1-SAVESTATUS.....  | 4-18 |
| 4-31 | Resume Status from Previous LSP: C1-RESSTATUS.....     | 4-18 |
| 4-32 | Check Submission Date: C1-CHKSUBDT1 .....              | 4-18 |
| 4-33 | Check Submission Date: Sample Algorithm.....           | 4-19 |

|      |  |      |
|------|--|------|
| 4-34 | Update LSP (CLOS): C1-LSPSTATUS.....                               | 4-19 |
| 4-35 | Update LSP (CLOS): Sample Algorithm .....                          | 4-20 |
| 4-36 | Update LSP (CANCEL): C1-LSPSTACAN.....                             | 4-20 |
| 4-37 | Update LSP (CANCEL): Sample Algorithm.....                         | 4-20 |
| 4-38 | Validate Expired Default Notice: C1-DEFNOTEXP .....                | 4-20 |
| 4-39 | Validate Expired Default Notice: Sample Algorithm.....             | 4-21 |
| 4-40 | Collateral Verification: C1-VRFYCOLS .....                         | 4-22 |
| 4-41 | Collateral Verification: Sample Algorithm.....                     | 4-22 |
| 4-42 | Account Association for Asset Repossession Case: C1-ARSACCTS.....  | 4-22 |
| 4-43 | Customer Association for Asset Repossession Case: C1-ARSCUSTS..... | 4-23 |
| 4-44 | Update Collateral Property: C1-SETCOLLPR .....                     | 4-23 |
| 4-45 | Close To do's Algorithm: C1-CLSTODOA.....                          | 4-24 |
| 4-46 | Close To do's Algorithm: Sample Algorithm.....                     | 4-25 |
| 4-47 | Validations for Mandatory Characteristics: C1-CHARVALU .....       | 4-26 |
| 4-48 | Validations for Mandatory Characteristics: Sample Algorithm .....  | 4-27 |
| 4-49 | Validations for Mandatory Characteristics: C1-CHARVALA .....       | 4-27 |
| 4-50 | Validations for Mandatory Characteristics: Sample Algorithm .....  | 4-29 |
| 4-51 | Update Collateral Status in the Host: C1-UPCOLLSTZ.....            | 4-29 |
| 4-52 | Update Collateral Status in the Host: Sample Algorithm.....        | 4-30 |
| 4-53 | Initiate Collateral Valuation: C1-COLLVALX.....                    | 4-31 |
| 4-54 | Initiate Collateral Valuation: Sample Algorithm .....              | 4-32 |
| 4-55 | Close To do's Algorithm: C1-CLSTODOV .....                         | 4-32 |
| 4-56 | Close To do's Algorithm: Sample Algorithm.....                     | 4-33 |
| 4-57 | Validations for Mandatory Characteristics: C1-CHARVALC.....        | 4-34 |
| 4-58 | Validations for Mandatory Characteristics: Sample Algorithm .....  | 4-35 |
| 4-59 | Update Collateral Status in the Host: C1-UPCOLLSTX.....            | 4-35 |
| 4-60 | Update Collateral Status in the Host: Sample Algorithm.....        | 4-36 |
| 4-61 | Validation Settlement: C1-VALSET .....                             | 4-36 |
| 4-62 | Validation Settlement: Sample Algorithm .....                      | 4-37 |
| 4-63 | Initiate LMI Process: C1-INITLMI.....                              | 4-37 |
| 4-64 | Initiate LMI Process: Sample Algorithm .....                       | 4-38 |
| 4-65 | Close To do's Algorithm: C1-CLSTODO.....                           | 4-39 |
| 4-66 | Close To do's Algorithm: Sample Algorithm.....                     | 4-40 |
| 4-67 | Update Collateral Property: C1-RESETCOLL .....                     | 4-40 |
| 4-68 | Update Collateral Property: Sample Algorithm.....                  | 4-41 |
| 4-69 | Update Collateral Status in the Host: C1-UPCOLLSTY.....            | 4-41 |
| 4-70 | Update Collateral Status in the Host: Sample Algorithm.....        | 4-42 |
| 4-71 | PTP Active Algorithm: C1-PTPACTIVE.....                            | 4-42 |
| 4-72 | PTP Active Algorithm: Sample Algorithm .....                       | 4-44 |
| 4-73 | PTP Active Algorithm: C1-PTPKEPT .....                             | 4-44 |
| 4-74 | PTP Active Algorithm: Sample Algorithm .....                       | 4-46 |
| 4-75 | PTP Active Algorithm: C1_PTPBRKLS .....                            | 4-46 |
| 4-76 | PTP Active Algorithm: Sample Algorithm .....                       | 4-48 |
| 4-77 | Rule Facts Populating Algorithm: C1-BRLSR.....                     | 4-49 |
| 4-78 | Sample Algorithm.....  | 4-51 |
| 5-1  | Feeder Services .....  | 5-1  |





---

---

# Preface

This document covers the staging data table structure and the services exposed by the system for host systems to use.

This preface contains the following topics:

- [Audience](#)
- [Documentation Accessibility](#)
- [Organization of the Guide](#)
- [Related Documents](#)
- [Conventions](#)

## Audience

This document is intended for the following audience:

- IT Deployment Team
- Consulting Staff
- Administrators

## 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.

## Organization of the Guide

This document contains:

### Chapter 1, "Introduction"

This chapter presents an overview of staging area and services exposed.

### Chapter 2, "System Overview"

This chapter provides information about the modules or systems interfaced with OBP Collections.

### Chapter 3, "Staging Area"

This chapter provides details of the feeder tables.

### Chapter 4, "Algorithms"

This chapter outlines the pre-shipped algorithm details.

### Chapter 5, "Feeder Services"

This chapter lists the services exposed by collections for data updates.

## Related Documents

For more information, see the following documentation:

- For the complete list of the adapters for integration with Oracle Banking Platform modules and technology stacks such as DMS / Alert / Email systems, see the Oracle Banking Platform Collections Adapter Configuration Guide.

## Conventions

The following text conventions are used in this document:

| Convention      | Meaning  |
|-----------------|--|
| <b>boldface</b> | Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.         |
| <i>italic</i>   | Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.                          |
| monospace       | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |

---

---

## Introduction

In Oracle Banking Platform, Collection system identifies delinquent accounts, fetches the account and party related data and stores it in the staging tables. After validation of these records, entity creation batch processes these records and moves them to Collections tables. For other host systems, it is expected that delinquent account data is pushed into these staging tables.

The feeder services exposed by Collections are invoked when changes in data take place in OBP. These services bring modified data into staging tables before batch processes these and update collections tables.

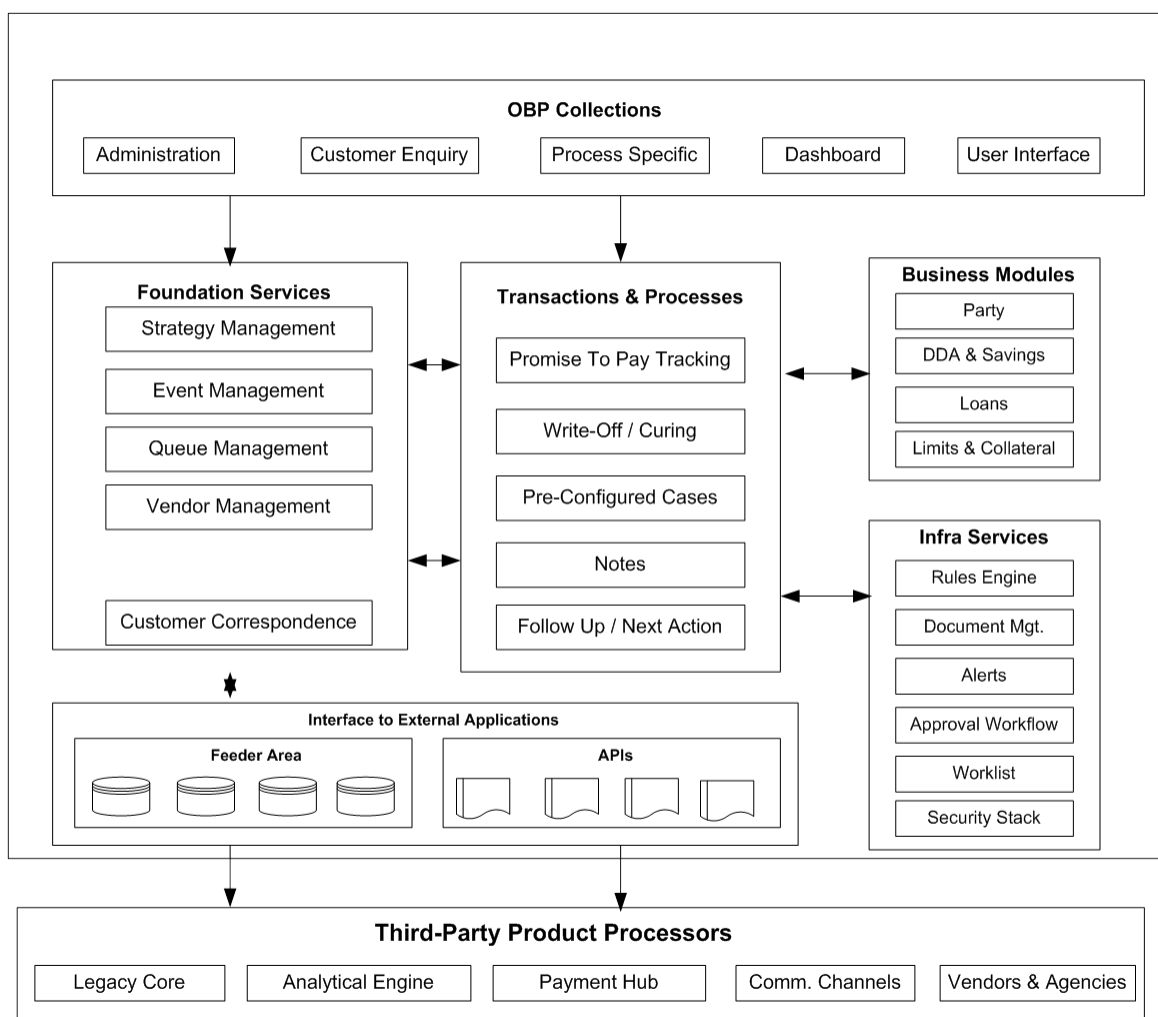


## System Overview

This chapter provides information about the modules or systems interfaced with OBP Collections.

The diagram below shows the interface that Collections has with other modules or systems. It depicts the collections flow and its interface with OBP modules.

**Figure 2-1 System Overview**





## Staging Area

This chapter provides information about the modules or systems interfaced with OBP Collections.

### 3.1 Feeder Tables

The feeder tables listed in this section provide a staging area for the host systems to push data. Offline collection batch process reads this data and creates accounts in Collections.

#### 3.1.1 Account Data

This section provides information on the tables related to accounts.

##### 3.1.1.1 Account Details

**Table Name:** Account Details (CI\_FDR\_ACCT)

**Description:** This table holds account related data from host.

**Table 3–1 Account Details**

| Field Name    | Description  | Value | Data Type    | Length | Required | Column Name       |
|---------------|--|-------|--------------|--------|----------|-------------------|
| Account No    | Account Number as stored in Host   |       | VARCHAR<br>2 | 40     | Y        | HOST_ACCT_<br>NBR |
| Host ID       | Source Host ID for host  |       | VARCHAR<br>2 | 10     | Y        | SRC_HOST_ID       |
| Business Unit | Business Unit of the Account. This field is used only if multi-branding features are to be used.               |       | VARCHAR<br>2 | 40     | N        | BUSINESS_<br>UNIT |
| Market Entity | Market Entity to which account belongs. This field is used only if multi-branding features are to be used.     |       | VARCHAR<br>2 | 40     | N        | MARKET_<br>ENTITY |
| Facility ID   | Facility ID under which account is created. This field is used based on the structure of accounts in the host. |       | VARCHAR<br>2 | 40     | N        | FACILITY_ID       |

**Table 3-1 (Cont.) Account Details**

| Field Name                  | Description  | Value  | Data Type | Length | Required | Column Name            |
|-----------------------------|--|--|-----------|--------|----------|------------------------|
| Liability ID                | Liability ID under which the Facility ID of the account has been created. This field is used based on the structure of accounts in the host. |  | VARCHAR2  | 40     | N        | LIABILITY_ID           |
| Product Class               | Product Class of the account   | Lending, CASA                                      | VARCHAR2  | 10     | Y        | HOST_PROD_CLASS_CD     |
| Product Group               | Product Group associated with the account  | Auto, Loan, and so on                              | VARCHAR2  | 20     | Y        | HOST_PROD_GRP_CD       |
| Product Code                | Code of the banking product offered to the customer  |  | VARCHAR2  | 10     | Y        | HOST_PROD_CD           |
| System Account Status       | As defined in the host   | Regular, Dormant, Closed, Written Off              | VARCHAR2  | 20     | Y        | HOST_SYS_ACCT_STAT_FLG |
| User defined Account Status | As defined in the host   | For example, Debit Block, Credit Block, and so on. | VARCHAR2  | 100    | N        | USR_DEF_ACCT_STAT_FLG  |
| Accrual Status              | This field displays the accrual status for the account.  | Normal, Suspended                                  | CHAR      | 1      | Y        | ACCRL_STAT_FLG         |
| Asset Classification Code   | As identified by the host  |  | VARCHAR2  | 30     | Y        | ASST_CLASS_CD          |
| Repayment Frequency         | Repayment Frequency of the loan  | Monthly, weekly, quarterly                         | VARCHAR2  | 30     | N        | REPAYMNT_FREQ          |
| Un-Cleared Payment Amount   | Sum of all uncleared credits to the account  |  | NUMBER    | 36,18  | N        | UNCLR_PAYMNT_AMT       |
| Loan Maturity Date          | Date when loan matures   |  | DATE      | 10     | Y        | MATURITY_DT            |
| Redraw Count                | Number of times a redraw has been performed  |  | NUMBER    | 3,0    | N        | REDRAW_CNT             |
| Account Write Off Date      | Date when account is fully written off/ abandoned  |  | DATE      | 10     | N        | WRITE_OFF_DT           |
| Account Write Off Amount    | Written off loan amt (abandonment amount). Total of all sums written off will be given.  |  | NUMBER    | 36,18  | N        | WRITE_OFF_AMT          |
| Last Provision Date         | Date on which the provision entry was last accounted   |  | DATE      | 10     | N        | LAST_PROVSN_DT         |



Table 3-1 (Cont.) Account Details

| Field Name  | Description  | Value   | Data Type | Length | Required | Column Name               |
|---|--|---|-----------|--------|----------|---------------------------|
| Provision Balance                                 | Latest balance in Provision GL for the account   |   | NUMBER    | 36,18  | N        | LAST_PROVSN_BAL           |
| Last Principal Write Off date                     | Date on which the principal write off entry was last passed  |   | DATE      | 10     | N        | LAST_PRNCPL_WRITE_OFF_DT  |
| Principal Write Off Balance                       | Latest balance in Principal Write Off GL for the account   |   | NUMBER    | 36,18  | N        | LAST_PRNCPL_WRITE_OFF_BAL |
| Loan Purpose Type                                 | Loan purpose types as applicable to the host   |   | VARCHAR2  | 20     | N        | ACCT_PURPS_TYPE           |
| Loan Purpose Code                                 | List of values as per loan purpose type  |   | VARCHAR2  | 20     | N        | ACCT_PURPS_CD             |
| Date of last loan restructure                     | Date when the loan was last restructured   |   | DATE      | 10     | N        | LAST_ACCT_RESTR_DT        |
| Offer ID  | Offer ID applicable to the customer account  |   | VARCHAR2  | 30     | N        | OFFER_ID                  |
| Offer Name  | Offer Name as per the Offer ID provided  |   | VARCHAR2  | 60     | N        | OFFER_NAME                |
| Account Opening Date or Initial Disbursement Date | Term Loan: First Disbursement Date<br>OD: Date on which OD facility is provided<br>Current Account with TOD facility: TOD utilization Date - Derived |   | DATE      | 10     | Y        | SETUP_DT                  |
| Account Currency Code                             | Currency code of the account   |   | VARCHAR2  | 3      | Y        | ACCT_CURR_CD              |
| Outstanding Amount                                | Outstanding Amount for the account   | <b>OD Accounts:</b><br>OD Limit Utilized +<br>AUF Limit Utilized +<br>Overdue Amount<br><br><b>Term Loans :</b><br>Outstanding Principal -<br>RPA Balance +<br>Overdue Amount | NUMBER    | 36,18  | Y        | OUTSTANDING_AMT           |

**Table 3-1 (Cont.) Account Details**

| Field Name                       | Description  | Value   | Data Type | Length | Required | Column Name           |
|----------------------------------|--|---|-----------|--------|----------|-----------------------|
| Overdue Amount                   | Overdue amount for the account   | <b>OD Accounts:</b><br>TOD utilized + Overline utilized + Temporary Excess utilized<br><br><b>Term Loans :</b><br>All amounts due and still unpaid                          | NUMBER    | 36,18  | Y        | OVERDUE_ AMT          |
| Account Limit                    | Sanctioned Limit offered to the account  | <b>OD Accounts :</b> OD limit + Temporary Excess limit<br><br><b>Term Loans :</b> Sanctioned Amount   | NUMBER    | 36,18  | Y        | OVERLIMIT_ AMT        |
| DPD                              | Longest Days past due value computed by the host                                     |   | NUMBER    | 4,0    | Y        | DAYS_PAST_ DUE        |
| Delinquency Start Date           | Current Delinquency Start Date. To be sent only once with the initial data hand off. |   | DATE      | 10     | N        | DEL_START_ DT         |
| Installment(s) in Arrears        | Total number of installments in arrears.   | Installment amount can at most consist of Principal, Interest and Fees. Even if one of the components is not fully paid; the installment will be construed as 'In Arrears'. | NUMBER    | 4,0    | N        | INSTALLMEN T_IN_ARS   |
| Disbursed Amount                 | Amount disbursed so far in case of a tranche   |   | NUMBER    | 36,18  | N        | DISBRS_ AMT           |
| Available for Disbursement       | Total loan amount available for disbursement   |   | NUMBER    | 36,18  | N        | TOTL_AVL_ DISBRS_ AMT |
| Last Payment Date                | Last Payment Date - Customer initiated credit.                                       |   | DATE      | 10     | N        | LAST_ PAYMENT_DT      |
| Last Payment Amount              | Last Payment Amount - Customer initiated credit.                                     |   | NUMBER    | 36,18  | N        | LAST_ PAYMENT_ AMT    |
| Amount of Debit Interest Accrued | Applicable only to accounts with Debit balance                                       |   | NUMBER    | 36,18  | N        | DR_INT_ ACCRD_ AMT    |
| Interest Rate                    | Rate of interest for current applicable stage.                                       |   | NUMBER    | 5,0    | Y        | INT_RATE              |

Table 3-1 (Cont.) Account Details

| Field Name            | Description   | Value  | Data Type    | Length | Required | Column Name               |
|-----------------------|---|--|--------------|--------|----------|---------------------------|
| Interest Type         | Fixed or Floating   |  | VARCHAR<br>2 | 14     | Y        | INT_TYPE                  |
| Address Type Code     | Overriding address type configured for an account                 |  | VARCHAR<br>2 | 20     | N        | ADDR_TYPE_CD              |
| Employee Account Flag | Indicate if the account belongs to a bank employee                | Y/N  | VARCHAR<br>2 | 1      | Y        | EMPLOYEE_ACCT_FLG         |
| Minor Account Status  | Indicate if the account belongs to a minor                        | Y/N  | VARCHAR<br>2 | 40     | Y        | MINOR_ACCOUNT_STATUS_TYPE |
| Home Branch           | Home Branch of the account  |  | VARCHAR<br>2 | 20     | Y        | BRANCH_CD                 |
| User Defined Field 1  | User Defined Field in case any additional attributes are required | <b>Exposure at Default :</b><br>String value coming from third party interface             | VARCHAR<br>2 | 60     | N        | UDF1                      |
| User Defined Field 2  | User Defined Field in case any additional attributes are required | <b>Loss Given Default :</b><br>String value coming from third party interface              | VARCHAR<br>2 | 60     | N        | UDF2                      |
| User Defined Field 3  | User Defined Field in case any additional attributes are required | <b>Expected Loss :</b> String value coming from third party interface                      | VARCHAR<br>2 | 60     | N        | UDF3                      |
| User Defined Field 4  | User Defined Field in case any additional attributes are required | <b>Risk Weighted Asset Calculation :</b><br>String value coming from third party interface | VARCHAR<br>2 | 60     | N        | UDF4                      |
| User Defined Field 5  | User Defined Field in case any additional attributes are required |  | VARCHAR<br>2 | 60     | N        | UDF5                      |
| User Defined Field 6  | User Defined Field in case any additional attributes are required |  | VARCHAR<br>2 | 60     | N        | UDF6                      |
| User Defined Field 7  | User Defined Field in case any additional attributes are required |  | VARCHAR<br>2 | 60     | N        | UDF7                      |
| User Defined Field 8  | User Defined Field in case any additional attributes are required |  | VARCHAR<br>2 | 60     | N        | UDF8                      |
| User Defined Field 9  | User Defined Field in case any additional attributes are required |  | VARCHAR<br>2 | 60     | N        | UDF9                      |

**Table 3–1 (Cont.) Account Details**

| Field Name             | Description   | Value | Data Type | Length | Required | Column Name                 |
|------------------------|---|-------|-----------|--------|----------|-----------------------------|
| User Defined Field 10  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF10                       |
| User Defined Field 11  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF11                       |
| User Defined Field 12  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF12                       |
| User Defined Field 13  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF13                       |
| User Defined Field 14  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF14                       |
| User Defined Field 15  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF15                       |
| User Defined Field 16  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF16                       |
| User Defined Field 17  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF17                       |
| User Defined Field 18  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF18                       |
| User Defined Field 19  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF19                       |
| User Defined Field 20  | User Defined Field in case any additional attributes are required           |       | VARCHAR2  | 60     | N        | UDF20                       |
| Reason for Delinquency | Reason code for delinquency of the account                                  |       | VARCHAR2  | 40     | N        | HOST_REASON_FOR_DELINQUENCY |
| Redraw Availability    | Facility to redraw loan   | Y/N   | CHAR      | 1      | Y        | FDR_REDRAW_AVL_SW           |
| Joint Applicant        | Indicator if the account has a Joint Applicant                              | Y/N   | VARCHAR2  | 1      | Y        | FDR_JOINT_APPLICANT_SW      |
| Delinquent             | Indicate if the account is delinquent                                       | Y/N   | VARCHAR2  | 1      | Y        | FDR_IS_DELINQUENT_SW        |
| Non Starter            | Indicate if the customer defaults the first installment after disbursement. | Y/N   | VARCHAR2  | 1      | Y        | FDR_NON_STARTER_SW          |

**Table 3-1 (Cont.) Account Details**

| Field Name             | Description   | Value                    | Data Type | Length | Required | Column Name              |
|------------------------|---|--------------------------|-----------|--------|----------|--------------------------|
| Behavior Score         | Current Behavior Score captured at account level  |                          | VARCHAR2  | 10     | N        | FDR_BEHAVIOR_SCORE       |
| Probability of Default | Current Probability of default captured at account level  |                          | VARCHAR2  | 60     | N        | PROBABILITY_OF_DEFLT_VAL |
| Application Score      | Application Score captured at the time of opening of account  |                          | VARCHAR2  | 10     | N        | FDR_APPL_SCR             |
| Loan to Value Ratio    | Loan to Value Ratio (Book/ Bank Value is considered) - Value of External Charge on Collateral is considered while calculating LVR |                          | NUMBER    | 5,2    | N        | FDR_LTV_VAL              |
| Loan to Value Ratio    | Loan to Value Ratio (MTM is considered) - Value of External Charge on Collateral is considered while calculating LVR              |                          | NUMBER    | 5,2    | N        | FDR_LVR_VAL              |
| Joint Nomination flag  | Joint Nomination flag   |                          | VARCHAR2  | 1      | N        | FDR_JOINT_NOMINATION_SW  |
| Record Type            | Signify if the data is created initially or is update for existing data   | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                 |
| Record Creation Date   | Date on which data is fed to Collections  |                          | DATE      | 10     | Y        | CRET_DTTM                |
| FDR_FORCED_SW          | FDR Forced SW   |                          | VARCHAR2  | 1      | Y        | FDR_FORCED_SW            |
| FORCED_REASON_CD       | Forced Reason CD  |                          | VARCHAR2  | 4      | Y        | FORCED_REASON_CD         |
| IOA_BALANCE_AMT        | IOA Balance Amount  |                          | NUMBER    | 36,18  | N        | IOA_BALANCE_AMT          |

**3.1.1.2 Account Arrears Details****Table Name:** Account Arrear Details (CI\_FDR\_ACCT\_ARS)**Description:** This table holds account arrears data from host.**Table 3-2 Account Arrears Details**

| Field Name      | Description                      | Value | Data Type | Length | Required | Column Name   |
|-----------------|----------------------------------|-------|-----------|--------|----------|---------------|
| Account No      | Account Number as stored in Host |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR |
| Host ID         | Source Host ID for host          |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Sequence Number | Sequence Number for arrear type  |       | VARCHAR2  | 50     | Y        | REFERENCE_VAL |

**Table 3–2 (Cont.) Account Arrears Details**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name      |
|----------------------|---|--------------------------|-----------|--------|----------|------------------|
| Arrear Type          | Arrear type like interest, fee, and so on.  |                          | VARCHAR2  | 40     | Y        | ARS_TYPE         |
| Arrear Amount        | Total arrear rose per arrear type. Details of arrear type should be sent only where arrear amount > 0 |                          | NUMBER    | 36,18  | Y        | ARS_ASSESSED_AMT |
| Paid Amount          | Amount paid so far. Zero if no payments are received  |                          | NUMBER    | 36,18  | N        | ARS_PAID_AMT     |
| Arrear Due           | As calculated by Host   |                          | NUMBER    | 36,18  | Y        | ARS_DUE_AMT      |
| Last Payment Date    | Date when last payment was received   |                          | DATE      | 10     | N        | LAST_PAYMENT_DT  |
| Days in Arrear       | Days this arrear is open. Zero is a valid value   |                          | NUMBER    | 4,0    | N        | DAYS_IN_ARS      |
| Installment Number   | Installment Number  |                          | NUMBER    | 5      | N        | INSTALLMENT_NUM  |
| Record Creation Date | Date on which data is fed to Collections  |                          | DATE      | 10     | Y        | CRET_DTTM        |
| Record Type          | Signify if the data is created initially or is update for existing data                               | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE         |

**3.1.1.3 Account Hardship Details**

**Table Name:** Account Hardship Details (CI\_FDR\_ACCT\_HARDSHIP\_DTLS)

**Description:** This table holds account hardship data from host.

**Table 3–3 Account Hardship Details**

| Field Name                | Description                           | Value | Data Type | Length | Required | Column Name             |
|---------------------------|---------------------------------------|-------|-----------|--------|----------|-------------------------|
| Account No                | Account Number as stored in Host      |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR           |
| Host ID                   | Source Host ID for host               |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID             |
| Application ID            | Hardship Application ID               |       | VARCHAR2  | 40     | Y        | HARDSHIP_APPLICATION_ID |
| Relief Effective Date     | Will be unique per Application ID     |       | DATE      | 10     | Y        | RELIEF_EFFECTIVE_DT     |
| Relief Expiry Date        | Will be unique per Application ID     |       | DATE      | 10     | Y        | RELIEF_EXPIRY_DT        |
| Relief Type(s)            | Can be more than 1 per application ID |       | VARCHAR2  | 40     | Y        | RELIEF_TYPE             |
| Number of Payments Waived | Number of Payments Waived             |       | NUMBER    | 4,0    | N        | NO_PAYMENT_WAIVED       |

**Table 3–3 (Cont.) Account Hardship Details**

| Field Name                        | Description  | Value                    | Data Type | Length | Required | Column Name               |
|-----------------------------------|--|--------------------------|-----------|--------|----------|---------------------------|
| User's Discretionary Margin (UDM) | These field details will be received only in case of Change Interest Rate relief type. |                          | VARCHAR2  | 60     | N        | USR_DISCRTN_MRGN          |
| UDM Start Date                    | User's discretionary Margin start date for the relief                                  |                          | DATE      | 10     | N        | USR_DISCRTN_MRGN_START_DT |
| UDM End Date                      | User's discretionary Margin end date for the relief                                    |                          | DATE      | 10     | N        | USR_DISCRTN_MRGN_END_DT   |
| Reason for UDM                    | Reason for User's discretionary Margin   |                          | VARCHAR2  | 60     | N        | USR_DISCRTN_MRGN_RSN      |
| Status                            | Current Status of Hardship Relief if applicable  |                          | CHAR      | 10     | N        | STATUS                    |
| Original Relief Type              | Original Relief Type   |                          | VARCHAR2  | 40     | N        | ORIG_RELIEF_TYPE          |
| Record Creation Date              | Date on which data is fed to Collections   |                          | DATE      | 10     | Y        | CRET_DTTM                 |
| Record Type                       | Signify if the data is created initially or is update for existing data                | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                  |

### 3.1.1.4 Account Repayment Schedule

**Table Name:** Account Repayment Schedule (CI\_FDR\_REPAYMENT\_SCH)

**Description:** This table holds account repayment schedule data from host.

**Table 3–4 Account Repayment Schedule**

| Field Name | Description                                       | Value | Data Type | Length | Required | Column Name       |
|------------|---|-------|-----------|--------|----------|-------------------|
| Account No | Account Number as stored in Host                  |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR     |
| Host ID    | Source Host ID for host                           |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID       |
| Date       | Date when the installments are to be recovered    |       | DATE      | 10     | Y        | INSTALLMENT_DT    |
| Amount     | Installment amount                                |       | NUMBER    | 36,18  | Y        | INSTALLMENT_AMT   |
| Principal  | Principal component                               |       | NUMBER    | 36,18  | Y        | PRINCIPAL_AMT     |
| Interest   | Interest component                                |       | NUMBER    | 36,18  | N        | INTEREST_AMT      |
| Fee        | Fee component, if any                             |       | NUMBER    | 36,18  | N        | FEE_AMT           |
| Balance    | Outstanding balance after the installment is paid |       | NUMBER    | 36,18  | N        | PRINCIPAL_BALANCE |

**Table 3–4 (Cont.) Account Repayment Schedule**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name     |
|----------------------|---|--------------------------|-----------|--------|----------|-----------------|
| Installment Number   | Installment number as per the loan structure                            |                          | NUMBER    | 5,0    | N        | INSTALLMENT_NUM |
| Record Creation Date | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM       |
| Record Type          | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE        |

### 3.1.1.5 Account Warning Indicator

**Table Name:** Account Warning Indicator (CI\_FDR\_ACCT\_WARNING\_IND)

**Description:** This table holds account warning indicators data from host.

**Table 3–5 Account Warning Indicator**

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name   |
|-------------------------|---|--------------------------|-----------|--------|----------|---------------|
| Account No              | Account Number as stored in Host  |                          | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR |
| Host ID                 | Source Host ID for host   |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Warning Indicator Code  | Warning Indicator code as stored in host                                |                          | VARCHAR2  | 10     | Y        | WARN_IND_CD   |
| Warning Indicator Value | Warning Indicator Value   |                          | VARCHAR2  | 1      | Y        | WARN_IND_VAL  |
| Start Date              | Start Date for warning indicator  |                          | DATE      | 10     | N        | START_DT      |
| End Date                | End Date for warning indicator code                                     |                          | DATE      | 10     | N        | END_DT        |
| Record Creation Date    | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM     |
| Record Type             | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE      |

## 3.1.2 Party Data

This section provides information on the tables related to party.

### 3.1.2.1 Party Account Relationship

**Table Name:** Party Account Relationship (CI\_FDR\_ACCT\_PER)

**Description:** This table holds account party relationships data from host.



**Table 3–6 Account Party Relationship**

| Field Name            | Description  | Value   | Data Type | Length | Required | Column Name          |
|-----------------------|--|---|-----------|--------|----------|----------------------|
| Account Number        | Account Number in Host   |   | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR        |
| Host ID               | Source Host ID for host  |   | VARCHAR2  | 10     | Y        | SRC_HOST_ID          |
| Party ID              | Party ID as stored in Host   |   | VARCHAR2  | 40     | Y        | HOST_CUST_NBR        |
| Account Relationship  | Account Relationship Code  | Sole Owner, Joint and First, Joint and Others, Trustee Auth Signatory and Power of Attorney | CHAR      | 8      | Y        | ACCT_REL_TYPE_CD     |
| Phone Banking Flag    | This flag signifies whether phone banking flag is enabled for the customer account relationship if maintained at this level    |   | VARCHAR2  | 1      | N        | FDR_PHONE_BANK_SW    |
| Internet Banking Flag | This flag signifies whether internet banking flag is enabled for the customer account relationship if maintained at this level |   | VARCHAR2  | 1      | N        | FDR_INTERNET_BANK_SW |
| Mobile Banking Flag   | This flag signifies whether mobile banking flag is enabled for the customer account relationship if maintained at this level   |   | VARCHAR2  | 1      | N        | FDR_MOBILE_BANK_SW   |
| ATM Card Flag         | This flag signifies whether ATM Card has been issued to the customer for this account  |   | VARCHAR2  | 1      | N        | FDR_ATM_SW           |
| Debit Card Flag       | This flag signifies whether Debit Card has been issued to the customer for this account  |   | VARCHAR2  | 1      | N        | FDR_DEBITCARD_SW     |
| UDF1                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF1                 |
| UDF2                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF2                 |
| UDF3                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF3                 |
| UDF4                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF4                 |
| UDF5                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF5                 |
| UDF6                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF6                 |
| UDF7                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF7                 |
| UDF8                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF8                 |
| UDF9                  | User Defined Fields  |   | VARCHAR2  | 60     | N        | UDF9                 |

**Table 3–6 (Cont.) Account Party Relationship**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name |
|----------------------|---|--------------------------|-----------|--------|----------|-------------|
| UDF10                | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF10       |
| Record Creation Date | Date on which data is fed to Collections                                  |                          | DATE      | 7      | Y        | CRET_DTTM   |
| Record Type          | Signifies if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE    |

**3.1.2.2 Party Details****Table Name:** Party Details (CI\_FDR\_PER)**Description:** This table holds party data from host.**Table 3–7 Party Details**

| Field Name  | Description  | Value | Data Type | Length | Required | Column Name        |
|---|--|-------|-----------|--------|----------|--------------------|
| Party ID  | Party ID as stored in Host   |       | VARCHAR2  | 40     | Y        | HOST_CUST_NBR      |
| Host ID   | Source Host ID for host  |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID        |
| Determinant Value   | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features.  |       | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE  |
| Party Class   | This field displays the party class of the customer. Party Class is a sub category in the Party Type. Fixed values for Individual party type are: Salaried Self Employed |       | VARCHAR2  | 8      | Y        | PER_CL_CD          |
| Date of Birth / Date of Incorporation/ Date of Trust Deed | Date of Birth, Date of Incorporation or Date of Trust Deed   |       | DATE      | 10     | Y        | BIRTH_DT           |
| Marital Status  | Marital Status of Party in case of Individual Customer   |       | VARCHAR2  | 20     | N        | MARITAL_STAT_FLG   |
| Customer Since  | Customer Since   |       | DATE      | 10     | Y        | SETUP_DT           |
| Gender  | Gender of Individual Customer  |       | VARCHAR2  | 1      | N        | GENDER             |
| Preferred Language  | Preferred Language of Communication  |       | VARCHAR2  | 3      | Y        | LANGUAGE_CD        |
| Marketing Info Flag                                       | Marketing Information Flag to continue communication   |       | VARCHAR2  | 4      | N        | RECV_MKTG_INFO_FLG |

**Table 3-7 (Cont.) Party Details**

| Field Name                | Description  | Value | Data Type | Length | Required | Column Name              |
|---------------------------|--|-------|-----------|--------|----------|--------------------------|
| Probability of Default    | String value coming from third party interface                                     |       | VARCHAR2  | 1      | N        | PROBABILITY_OF_DEFLT_VAL |
| 3rd Party Flag            | Valid Values: Y, N - Indicates if a third party is associated to the party         |       | VARCHAR2  | 1      | N        | FDR_THIRD_PARTY_SW       |
| Internet Banking Flag     | This flag signifies if internet banking flag is enabled for the customer           | Y/N   | VARCHAR2  | 1      | N        | FDR_INTERNET_BANK_SW     |
| Phone Banking Flag        | This flag signifies if phone banking flag is enabled for the customer              | Y/N   | VARCHAR2  | 1      | N        | FDR_PHONE_BANK_SW        |
| VIP Flag                  | This flag signifies if this is a VIP customer                                      | Y/N   | VARCHAR2  | 1      | N        | FDR_VIP_PARTY_SW         |
| Behavior Score            | Also available at Customer Level - Numeric value coming from third party interface |       | VARCHAR2  | 10     | N        | FDR_BEHAVIOR_SCORE       |
| Customer Risk Score (CRS) | Customer Risk Score (CRS)  |       | VARCHAR2  | 10     | N        | FDR_CUSTOMER_RISK_SCORE  |
| Party Type                | This field displays the party type. Valid values: - Individual - Corporate - Trust |       | VARCHAR2  | 10     | Y        | FDR_PER_OR_BUS_FLG       |
| User Defined Value 1      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF1                     |
| User Defined Value 2      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF2                     |
| User Defined Value 3      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF3                     |
| User Defined Value 4      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF4                     |
| User Defined Value 5      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF5                     |
| User Defined Value 6      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF6                     |
| User Defined Value 7      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF7                     |
| User Defined Value 8      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF8                     |
| User Defined Value 9      | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF9                     |
| User Defined Value 10     | User Defined Fields  |       | VARCHAR2  | 60     | N        | UDF10                    |

**Table 3–7 (Cont.) Party Details**

| Field Name                 | Description   | Value                    | Data Type | Length | Required | Column Name             |
|----------------------------|---|--------------------------|-----------|--------|----------|-------------------------|
| User Defined Value 11      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF11                   |
| User Defined Value 12      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF12                   |
| User Defined Value 13      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF13                   |
| User Defined Value 14      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF14                   |
| User Defined Value 15      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF15                   |
| User Defined Value 16      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF16                   |
| User Defined Value 17      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF17                   |
| User Defined Value 18      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF18                   |
| User Defined Value 19      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF19                   |
| User Defined Value 20      | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF20                   |
| Enterprise customer number | OCH Number  |                          | VARCHAR2  | 60     | N        | FDR_ENTERPRISE_CUST_NBR |
| Record Creation Date       | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM               |
| Record Type                | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                |

**3.1.2.3 Party Address Details**

**Table Name:** Party Address Details (CI\_FDR\_PER\_ADDR)

**Description:** This table holds party address data from host.

**Table 3–8 Party Address Details**

| Field Name   | Description                          | Value                                     | Data Type | Length | Required | Column Name   |
|--------------|--------------------------------------|---|-----------|--------|----------|---------------|
| Party ID     | Party ID as stored in Host           |   | VARCHAR2  | 40     | Y        | HOST_CUST_NBR |
| Host ID      | Source Host ID for host              |   | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Address Type | Address Type Code maintained in Host | Home,<br>Business,<br>Postal,<br>Seasonal | VARCHAR2  | 20     | Y        | ADDR_TYPE_CD  |

**Table 3–8 (Cont.) Party Address Details**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name       |
|----------------------|---|--------------------------|-----------|--------|----------|-------------------|
| Sequence ID          | Sequence ID maintained in Host for each address type in case multiple addresses are maintained for same address type              |                          | VARCHAR2  | 40     | N        | FDR_ADDR_SEQ_ID   |
| Address 1            | Address Line 1  |                          | VARCHAR2  | 120    | Y        | ADDRESS_LINE1     |
| Address 2            | Address Line 2  |                          | VARCHAR2  | 120    | N        | ADDRESS_LINE2     |
| Address 3            | Address Line 3  |                          | VARCHAR2  | 120    | N        | ADDRESS_LINE3     |
| Address 4            | Address Line 4  |                          | VARCHAR2  | 120    | N        | ADDRESS_LINE4     |
| City                 | City Code   |                          | VARCHAR2  | 50     | Y        | CITY_CD           |
| Country              | Country Code  |                          | VARCHAR2  | 30     | Y        | COUNTRY_CD        |
| Post/ Zip/ Pin Code  | Zip Code  |                          | VARCHAR2  | 30     | Y        | ZIP_CD            |
| Determinant Value    | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features. |                          | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE |
| Status               | Active or Inactive status   |                          | VARCHAR2  | 10     | Y        | STATUS            |
| Record Creation Date | Date on which data is fed to Collections  |                          | DATE      | 10     | Y        | CRET_DTTM         |
| Record Type          | Signify if the data is created initially or is update for existing data   | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE          |
| FDR_STATE_CD         | FDR State CD  |                          | VARCHAR2  | 60     | N        | FDR_STATE_CD      |

**3.1.2.4 Party Employment Details**

**Table Name:** Party Employment Details (CI\_FDR\_PER\_EMPLOYMENT\_PROF)

**Description:** This table holds party employment details from host.

**Table 3–9 Party Employment Details**

| Field Name        | Description   | Value   | Data Type | Length | Required | Column Name        |
|-------------------|---|---|-----------|--------|----------|--------------------|
| Party ID          | Party ID as stored in Host  |   | VARCHAR2  | 40     | Y        | HOST_CUST_NBR      |
| Host ID           | Source Host ID for host   |   | VARCHAR2  | 10     | Y        | SRC_HOST_ID        |
| Determinant Value | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features. |   | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE  |
| Sequence ID       | Sequence ID of Employment details   |   | VARCHAR2  | 40     | Y        | FDR_EMP_SEQ_ID     |
| Employment Status | Employment Status Code  | <b>Employment Status examples:</b><br>For example, Full Time, Part Time, Home Duties, Non-Resident, Pensioner, Retired, Student, Superannuation, Unemployed | VARCHAR2  | 3      | Y        | EMPLOYMENT_STAT_CD |
| Employment Type   | Employment Type   | <b>Employment Type examples:</b><br>For example, Others, Salaried, Self Employed, Both- Salaried and Self Employed  | VARCHAR2  | 30     | Y        | EMPLOYMENT_TYPE    |
| Employer Name     | Name of the employer of the customer  |   | VARCHAR2  | 120    | Y        | EMPLOYER_NAME      |
| Industry Type     | Industry Type   |   | VARCHAR2  | 30     | N        | INDUSTRY_TYPE      |
| Company Type      | Company Type  | For example, Public Limited, Private Limited, Government Organization   | VARCHAR2  | 30     | N        | COMPANY_TYPE       |
| Occupation        | Occupation  |   | VARCHAR2  | 30     | Y        | PROFESSION_TYPE    |
| Designation       | Designation   |   | VARCHAR2  | 120    | N        | DESIGNATION_TXT    |

**Table 3–9 (Cont.) Party Employment Details**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name         |
|----------------------|---|--------------------------|-----------|--------|----------|---------------------|
| Gross Annual Salary  | Gross Annual Salary   |                          | NUMBER    | 22     | N        | GRS_ ANNUAL_ INCOME |
| Start Date           | Start Date  |                          | DATE      | 10     | Y        | START_DT            |
| End Date             | End Date  |                          | DATE      | 10     | N        | END_DT              |
| Status               | Status  |                          | VARCHAR2  | 10     | Y        | STATUS              |
| Record Creation Date | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM           |
| Record Type          | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE            |

**3.1.2.5 Party Identification Details****Table Name:** Party Identification Details (CI\_FDR\_PER\_ID)**Description:** This table holds party ID type details from host.**Table 3–10 Party Identification Details**

| Field Name           | Description   | Value   | Data Type | Length | Required | Column Name             |
|----------------------|---|---|-----------|--------|----------|-------------------------|
| Party ID             | Party ID as stored in Host  |   | VARCHAR2  | 40     | Y        | HOST_CUST_ NBR          |
| Host ID              | Source Host ID for host   |   | VARCHAR2  | 10     | Y        | SRC_HOST_ID             |
| Identification Type  | Value of Identification Type Code   | Passport No,<br>Driving License No,<br>and so on. | VARCHAR2  | 30     | Y        | FDR_ID_TYPE             |
| ID Value             | Identification Number corresponding to each of the identification types   |   | VARCHAR2  | 40     | Y        | FDR_ID_NBR              |
| Determinant Value    | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features. |   | VARCHAR2  | 60     | Y        | FDR_ DETERMINAN T_VALUE |
| Issue Date           | Issue Date for Identification Number  |   | DATE      | 10     | N        | FDR_ISSUE_ DT           |
| Expiry Date          | Expiry Date for Identification Number   |   | DATE      | 10     | N        | FDR_EXPIRY_ DT          |
| Record Creation Date | Date on which data is fed to Collections  |   | DATE      | 10     | Y        | CRET_DTTM               |
| Record Type          | Signify if the data is created initially or is update for existing data   | I - Insert<br>U - Update                          | VARCHAR2  | 10     | Y        | RCD_TYPE                |

### 3.1.2.6 Party Name Details

**Table Name:** Party Name Details (CI\_FDR\_PER\_NAME)

**Description:** This table holds party name details from host.

**Table 3–11 Party Name Details**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name            |
|----------------------|---|--------------------------|-----------|--------|----------|------------------------|
| Party ID             | Party ID as stored in Host  |                          | VARCHAR2  | 40     | Y        | HOST_CUST_NBR          |
| Host ID              | Source Host ID for host   |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID            |
| Name Type            | Type of Name  | Legal                    | VARCHAR2  | 10     | Y        | FDR_NAME_TYPE          |
| First Prefix         | Indicates the first prefix  |                          | VARCHAR2  | 30     | N        | FDR_FIRST_PREFIX_ID    |
| Second Prefix        | Indicates the second prefix   |                          | VARCHAR2  | 30     | N        | FDR_SECOND_PREFIX_ID   |
| First Name           | First Name of the customer  |                          | VARCHAR2  | 50     | N        | FDR_FIRST_NAME         |
| First Middle Name    | First middle name of the customer   |                          | VARCHAR2  | 50     | N        | FDR_MIDDLE_NAME_FIRST  |
| Second Middle Name   | Second Middle name of the customer  |                          | VARCHAR2  | 50     | N        | FDR_MIDDLE_NAME_SECOND |
| Last Name            | Last Name of the customer   |                          | VARCHAR2  | 50     | N        | FDR_LAST_NAME          |
| Suffix ID            | Suffix ID in the name   |                          | VARCHAR2  | 30     | N        | FDR_SUFFIX_ID          |
| Full Name            | Full name of the customer   |                          | VARCHAR2  | 250    | N        | FDR_FULL_NAME          |
| Short Name           | Short Name of the customer  |                          | VARCHAR2  | 60     | N        | FDR_SHORT_NAME         |
| Determinant Value    | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features. |                          | VARCHAR2  | 60     | Y        | FDR_DETERMINANT_VALUE  |
| Primary Name Flag    | Signifies if a particular name needs to be used as a primary name for the customer  | Y/N                      | CHAR      | 1      | Y        | FDR_PRIMARY_NAME_SW    |
| Record Creation Date | Date on which data is fed to Collections  |                          | DATE      | 10     | Y        | CRET_DTTM              |
| Record Type          | Signify if the data is created initially or is update for existing data   | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE               |

### 3.1.2.7 Party Contact Preference Details

**Table Name:** Party Contact Preference Details (CI\_FDR\_CONTACT\_PREF)



**Description:** This table holds the party contact preference data from host.

**Table 3–12 Party Contact Preference Details**

| Field Name           | Description   | Value  | Data Type | Length | Required | Column Name        |
|----------------------|---|--|-----------|--------|----------|--------------------|
| Party ID             | Party ID as stored in Host  |  | VARCHAR2  | 40     | Y        | HOST_CUST_NBR      |
| Host ID              | Source Host ID for host   |  | VARCHAR2  | 10     | Y        | SRC_HOST_ID        |
| Determinant Value    | Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features. |  | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE  |
| Contact Point        | Type of Contact Point   | Mobile, Landline, Email, and so on.                    | VARCHAR2  | 10     | Y        | CONTACT_POINT_TYPE |
| Purpose              | Purpose   |  | VARCHAR2  | 120    | N        | PURPOSE_TXT        |
| Value                | Contact Point Value, for example, if Contact Point is Mobile then provide mobile number, if Email then provide email ID           |  | VARCHAR2  | 400    | Y        | CONTACT_VALUE      |
| Contact Type         | Home, Work, Others  |  | VARCHAR2  | 10     | Y        | CONTACT_PREF_TYPE  |
| Start Date           | Start date for using this contact point and type  |  | DATE      | 10     | N        | START_DT           |
| End Date             | End date for using this contact point and type  |  | DATE      | 10     | N        | END_DT             |
| Time From (weekdays) | Start Time for contacting on weekdays   | In hundred hour format (for example, 1800 for 6:00 PM) | NUMBER    | 10     | N        | WKDAY_FROM_TM      |
| Time To (weekdays)   | End Time for contacting on weekdays   | In hundred hour format (for example, 1800 for 6:00 PM) | NUMBER    | 10     | N        | WKDAY_TO_TM        |
| Time From (weekends) | Start Time for contacting on weekends   | In hundred hour format (for example, 1800 for 6:00 PM) | NUMBER    | 10     | N        | WKEND_FROM_TM      |
| Time To (weekends)   | End Time for contacting on weekends   | In hundred hour format (for example, 1800 for 6:00 PM) | NUMBER    | 10     | N        | WKEND_TO_TM        |

**Table 3–12 (Cont.) Party Contact Preference Details**

| Field Name            | Description   | Value                    | Data Type | Length | Required | Column Name          |
|-----------------------|---|--------------------------|-----------|--------|----------|----------------------|
| Preference Frequency  | Preferred Frequency of contact  |                          | NUMBER    | 22     | N        | PREFERENCE_FREQUENCY |
| Primary Contact Point | Primary Contact Point Flag  |                          | VARCHAR2  | 10     | N        | FDR_PRIMARY_SW       |
| Status                | Status - Active or Dormant  |                          | VARCHAR2  | 10     | Y        | STATUS               |
| Record Creation Date  | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM            |
| Record Type           | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE             |

### 3.1.2.8 Party Warning Indicators

**Table Name:** Party Warning Indicators (CI\_FDR\_PARTY\_WARNING\_IND)

**Description:** This table holds the party warning indicators data from host.

**Table 3–13 Party Warning Indicators**

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name   |
|-------------------------|---|--------------------------|-----------|--------|----------|---------------|
| Party ID                | Party ID as stored in Host  |                          | VARCHAR2  | 40     | Y        | HOST_CUST_NBR |
| Host ID                 | Source Host ID for host   |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Warning Indicator Code  | Warning Indicator Code  |                          | VARCHAR2  | 10     | Y        | WARN_IND_CD   |
| Warning Indicator Value | Value of Warning Indicator Code   | Y/N                      | VARCHAR2  | 1      | Y        | WARN_IND_VAL  |
| Start Date              | Start Date of Warning Indicator   |                          | DATE      | 10     | N        | START_DT      |
| End Date                | End Date of warning Indicator   |                          | DATE      | 10     | N        | END_DT        |
| Record Creation Date    | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM     |
| Record Type             | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE      |

## 3.1.3 Collateral Data

This section provides information on the tables related to collaterals.

### 3.1.3.1 Collateral Details

**Table Name:** Collateral Details (CI\_FDR\_COLLATERAL)

**Description:** This table holds collateral data from host.

**Table 3–14 Collateral Details**

| Field Name               | Description  | Value                            | Data Type | Length | Required | Column Name          |
|--------------------------|--|----------------------------------|-----------|--------|----------|----------------------|
| Collateral Code          | Collateral Code as stored in host                                      |                                  | VARCHAR2  | 40     | Y        | COLLATERAL_CD        |
| Host ID                  | Source Host ID for host  |                                  | VARCHAR2  | 10     | Y        | SRC_HOST_ID          |
| Collateral Type          | Type of Collateral   |                                  | VARCHAR2  | 40     | Y        | COLLATERAL_TYPE      |
| Collateral Sub Type      | If there are any collateral sub type                                   |                                  | VARCHAR2  | 40     | N        | COLLATERAL_SUB_TYPE  |
| Collateral Category      | Collateral Category  |                                  | VARCHAR2  | 40     | N        | COLLATERAL_CAT       |
| Collateral Description   | Collateral Description   |                                  | VARCHAR2  | 300    | Y        | FDR_COLLATERAL_DESCR |
| Nature                   | Normal/ Guarantee  |                                  | VARCHAR2  | 40     | N        | COLLATERAL_NATURE    |
| Collateral Currency      | Collateral Currency  |                                  | VARCHAR2  | 3      | Y        | COLLATERAL_CUR       |
| Assessed Value           | Market Value   |                                  | NUMBER    | 36,18  | Y        | ASSESSED_VALUE       |
| Assessment Date          | Date of assessment   |                                  | DATE      | 10     | Y        | ASSESSED_DT          |
| Bank Value               | Book Value   |                                  | NUMBER    | 36,18  | Y        | BANK_VALUE           |
| Sold By                  | This property is required to identify entity which sold the collateral | Customer (Borrower), Bank, Court | VARCHAR2  | 255    | N        | SOLD_BY              |
| Date of Sale             | Date on which the collateral was sold                                  |                                  | DATE      | 10     | N        | SALE_DT              |
| Amount Realized          | Gross Sale amount  |                                  | NUMBER    | 36,18  | N        | AMT_REALIZED         |
| Date of Settlement       | Date on which settlement took place                                    |                                  | DATE      | 10     | N        | SETLMNT_DT           |
| Realization Status       | Final status of realization  |                                  | VARCHAR2  | 40     | N        | REALIZATION_STATUS   |
| Amount Recovered         | Gross Sale Amount less Costs incurred for sale of collateral           |                                  | NUMBER    | 36,18  | N        | FDR_AMT_RECOVERED    |
| Collateral Address Line1 | Collateral Address Line1   |                                  | VARCHAR2  | 120    | N        | ADDRESS_LINE1        |
| Collateral Address Line2 | Collateral Address Line2   |                                  | VARCHAR2  | 120    | N        | ADDRESS_LINE2        |
| Collateral Address Line3 | Collateral Address Line3   |                                  | VARCHAR2  | 120    | N        | ADDRESS_LINE3        |
| Collateral Address Line4 | Collateral Address Line4   |                                  | VARCHAR2  | 120    | N        | ADDRESS_LINE4        |
| City code                | City code  |                                  | VARCHAR2  | 50     | N        | CITY_CD              |
| Postal code              | Postal code  |                                  | VARCHAR2  | 30     | N        | ZIP_CD               |

**Table 3–14 (Cont.) Collateral Details**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name |
|----------------------|---|--------------------------|-----------|--------|----------|-------------|
| State code           | State code  |                          | VARCHAR2  | 6      | N        | STATE_CD    |
| Country code         | Country code  |                          | VARCHAR2  | 30     | N        | COUNTRY_CD  |
| Record Creation Date | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM   |
| Record Type          | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE    |

**3.1.3.2 Collateral Charge Details**

**Table Name:** Collateral Charge Details (CI\_FDR\_COLLATERAL\_CHRG)

**Description:** This table holds collateral charges details from host.

**Table 3–15 Collateral Charges Details**

| Field Name           | Description  | Value                    | Data Type | Length | Required | Column Name    |
|----------------------|--|--------------------------|-----------|--------|----------|----------------|
| Collateral Code      | Collateral Code as stored in host  |                          | VARCHAR2  | 40     | Y        | COLLATERAL_CD  |
| Host ID              | Source Host ID for host  |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID    |
| Charge Code          | Charge Codes maintained in the host  |                          | VARCHAR2  | 20     | Y        | CHRG_CD        |
| Bank Value Relied On | Bank value for each of the charge codes  |                          | NUMBER    | 36,18  | Y        | AVL_CHARGE_VAL |
| Charge Currency      | Currency in which Charge Value is calculated. Collateral currency and charge currency can differ |                          | CHAR      | 3      | Y        | CHARGE_CURR    |
| Record Creation Date | Date on which data is fed to Collections   |                          | DATE      | 10     | Y        | CRET_DTTM      |
| Record Type          | Signify if the data is created initially or is update for existing data                          | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE       |

**3.1.3.3 Collateral Entity Mapping**

**Table Name:** Collateral Entity Mapping (CI\_FDR\_COLLATERAL\_ENTITY)

**Description:** This table holds the collateral entity mapping from host. Collateral can be mapped to facility or to an account.

**Table 3–16 Collateral Entity Mapping**

| Field Name      | Description                          | Value                | Data Type | Length | Required | Column Name   |
|-----------------|--------------------------------------|----------------------|-----------|--------|----------|---------------|
| Collateral Code | Collateral Code as stored in host    |                      | VARCHAR2  | 40     | Y        | COLLATERAL_CD |
| Host ID         | Source Host ID for host              |                      | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Entity Type     | Entity to which collateral is mapped | ACCOUNT,<br>FACILITY | VARCHAR2  | 10     | Y        | ENTITY_TYPE   |

**Table 3–16 (Cont.) Collateral Entity Mapping**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name               |
|----------------------|---|--------------------------|-----------|--------|----------|---------------------------|
| Entity ID            | Entity ID of entity to which collateral is mapped                       |                          | VARCHAR2  | 40     | Y        | COL_ENTITY_ID             |
| Contribution Switch  | Identify if the collateral is contributing towards an entity            | Y/N                      | VARCHAR2  | 1      | Y        | FDR_LIMIT_CONTRIBUTION_SW |
| Record Creation Date | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM                 |
| Record Type          | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                  |

### 3.1.3.4 Collateral Guarantor Mapping

**Table Name:** Collateral Guarantor Mapping (CI\_FDR\_COLLATERAL\_GRNTR)

**Description:** This table holds the guarantors data for the collateral.

**Table 3–17 Collateral Guarantor Mapping**

| Field Name           | Description   | Value                    | Data Type | Length | Required | Column Name   |
|----------------------|---|--------------------------|-----------|--------|----------|---------------|
| Collateral Code      | Collateral Code as stored in host                                       |                          | VARCHAR2  | 40     | Y        | COLLATERAL_CD |
| Host ID              | Source Host ID for host   |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID   |
| Party ID             | Party ID of the guarantor   |                          | VARCHAR2  | 40     | Y        | HOST_CUST_NBR |
| Record Creation Date | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM     |
| Record Type          | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE      |

### 3.1.3.5 Collateral Owner Mapping

**Table Name:** Collateral Owner Mapping (CI\_FDR\_COLLATERAL\_PARTY)

**Description:** This table holds ownership of parties for the collateral.

**Table 3–18 Collateral Owner Mapping**

| Field Name      | Description                               | Value | Data Type | Length | Required | Column Name   |
|-----------------|---|-------|-----------|--------|----------|---------------|
| Collateral Code | Collateral Code as stored in host         |       | VARCHAR2  | 40     | N        | COLLATERAL_CD |
| Host ID         | Source Host ID for host                   |       | VARCHAR2  | 10     | N        | SRC_HOST_ID   |
| Party ID        | Party ID of Customer mapped to collateral |       | VARCHAR2  | 40     | N        | HOST_CUST_NBR |

**Table 3–18 (Cont.) Collateral Owner Mapping**

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name       |
|-------------------------|---|--------------------------|-----------|--------|----------|-------------------|
| Percentage of Ownership | Ownership Percentage of each of the Party                               |                          | VARCHAR2  | 10     | Y        | OWNERSHIP_PERCENT |
| Record Creation Date    | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM         |
| Record Type             | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | N        | RCD_TYPE          |

### 3.1.4 Insurance Data

This section provides information on the tables related to insurance.

#### 3.1.4.1 Insurance Details

**Table Name:** Insurance Details (CI\_FDR\_INSR\_DTLS)

**Description:** This table holds insurance records for collateral, party, or facility.

**Table 3–19 Insurance Details**

| Field Name            | Description                           | Value                              | Data Type | Length | Required | Column Name           |
|-----------------------|---------------------------------------|------------------------------------|-----------|--------|----------|-----------------------|
| Entity ID             | Value of Entity ID                    |                                    | VARCHAR2  | 40     | Y        | COL_ENTITY_ID         |
| Entity Type           | Entity on which Insurance is captured | COLLATERAL,<br>PERSON,<br>FACILITY | VARCHAR2  | 10     | Y        | ENTITY_TYPE           |
| Host ID               | Source Host ID for host               |                                    | VARCHAR2  | 10     | Y        | SRC_HOST_ID           |
| Insurance ID          | Insurance ID as stored in host        |                                    | VARCHAR2  | 60     | Y        | INSURANCE_ID          |
| Policy No             | Policy number of the Insurance        |                                    | VARCHAR2  | 30     | Y        | POLICY_NO             |
| Insurance Policy Name | Insurance Policy Name                 |                                    | VARCHAR2  | 100    | Y        | INSURANCE_POLICY_NAME |
| Insured Currency      | Currency Code of the Insured Amount   |                                    | VARCHAR2  | 3      | Y        | INSURED_CURR          |
| Insured Amount        | Insured Amount                        |                                    | NUMBER    | 36,18  | Y        | INSURED_AMT           |
| Insurer Code          | Insurer Code as stored in host        |                                    | VARCHAR2  | 40     | Y        | INSURER_CD            |
| Insurer Name          | Insurer Name as stored in host        |                                    | VARCHAR2  | 64     | Y        | INSURER_NAME          |
| Policy Start Date     | Start date of Policy                  |                                    | DATE      | 10     | Y        | POLICY_START_DT       |
| Policy End Date       | End date of Policy                    |                                    | DATE      | 10     | Y        | POLICY_END_DT         |
| Premium Amount        | Insurance Premium                     |                                    | NUMBER    | 36,18  | Y        | PREMIUM_AMT           |

**Table 3–19 (Cont.) Insurance Details**

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name             |
|-------------------------|---|--------------------------|-----------|--------|----------|-------------------------|
| Payment Frequency       | Premium payment frequency   |                          | VARCHAR2  | 30     | N        | PAYMENT_FREQ            |
| Insurance Type          | Insurance Type  | LMI PPI                  | VARCHAR2  | 30     | N        | INSURANCE_TYPE          |
| Record Creation Date    | Date on which data is fed to Collections                                |                          | DATE      | 10     | Y        | CRET_DTTM               |
| Record Type             | Signify if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                |
| DUA_APPLICABLE          | DUA Applicable  |                          | VARCHAR2  | 1      | N        | DUA_APPLICABLE          |
| NET_BORR_PREMIUM_AMOUNT | NET BORR Premium Amount   |                          | NUMBER    | 36,18  | N        | NET_BORR_PREMIUM_AMOUNT |
| FDR_PARTY_ID            | FDR Party ID  |                          | VARCHAR2  | 40     | N        | FDR_PARTY_ID            |

### 3.1.5 Payment Data

This section provides information on the tables related to payments.

#### 3.1.5.1 Online Payment Records

**Table Name:** Online Payment (CI\_FDR\_PAYMENTS)

**Description:** This table holds the failed online payment records which is used by payment processing batch for offline processing.

**Table 3–20 Online Payment**

| Field Name                   | Description  | Value | Data Type | Length | Required | Column Name             |
|------------------------------|--|-------|-----------|--------|----------|-------------------------|
| Account No                   | Account Number as stored in Host                             |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR           |
| Host ID                      | Source Host ID for host                                      |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID             |
| Transaction Reference Number | Transaction Reference Number for payment transaction in host |       | VARCHAR2  | 30     | Y        | XREF_NO                 |
| Transaction Date             | Date of Transaction  |       | DATE      | 10     | Y        | FDR_TRANSACTION_DT      |
| Transaction Time             | Time for Transaction   |       | DATE      | 10     | Y        | FDR_TRANSACTION_TM      |
| Value Date                   | Value Date on which the transaction was posted in the host   |       | DATE      | 10     | Y        | FDR_VALUE_DT            |
| Transaction Currency         | Currency code of the transaction                             |       | VARCHAR2  | 3      | Y        | FDR_TRANSACTION_CURR_CD |

**Table 3–20 (Cont.) Online Payment**

| Field Name                           | Description  | Value                    | Data Type | Length | Required | Column Name              |
|--------------------------------------|--|--------------------------|-----------|--------|----------|--------------------------|
| Transaction Amount                   | Payment Amount   |                          | NUMBER    | 36,18  | Y        | FDR_TRANSACTION_AMT      |
| Account Currency                     | Account Currency Code  |                          | VARCHAR2  | 3      | Y        | FDR_ACCT_CURR_CD         |
| Account Balance                      | Account Balance after Payment  |                          | NUMBER    | 36,18  | Y        | FDR_ACCT_AMT             |
| Transaction Code                     | Transaction Code as captured in the host   |                          | VARCHAR2  | 10     | N        | FDR_TRANSACTION_CD       |
| Narration Text                       | Narration text for the transaction   |                          | VARCHAR2  | 120    | N        | FDR_NARRATION_TXT        |
| Transaction Type Flag                | Identify if the transaction is Credit or Debit that is, actual payment transaction or reversal | C/D                      | CHAR      | 1      | Y        | FDR_TRANSACTION_TYPE_FLG |
| Record Creation Date                 | Date on which data is fed to Collections   |                          | DATE      | 10     | Y        | CRET_DTTM                |
| Record Type                          | Signify if the data is created initially or is update for existing data                        | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE                 |
| Original Transaction ref number      | Used for cancellation of payments  |                          | VARCHAR2  | 30     | N        | ORIG_XREF_NO             |
| Transaction sequence number          | Transaction sequence number  |                          | VARCHAR2  | 30     | Y        | FDR_XREF_SUB_SEQ_NO      |
| Original Transaction sequence number | Used for cancellation of payments  |                          | VARCHAR2  | 30     | N        | FDR_ORIG_XREF_SUB_SEQ_NO |

## 3.2 OBP Views

Collections system pulls delinquent account data from the following views provided by OBP.

### 3.2.1 Main Account Views

The main account views are as follows:

- FLX\_COL\_ACCT\_DATA\_XF
- FLX\_LN\_COL\_FD\_ACCT\_VW
- FLX\_DD\_COL\_DATA\_TOD\_XF\_VW
- FLX\_DD\_COL\_DATA\_XF\_VW
- FLX\_AC\_COL\_FD\_ACCT\_ARS\_VW
- FLX\_LN\_COL\_FD\_SCH\_VW



- FLX\_AC\_WARNING\_IND\_COL\_XF\_VW

### 3.2.2 Account Updateable Views

The account updateable views are as follows:

- FLX\_DD\_COL\_DATA\_XF\_UPD\_VW
- FLX\_LN\_COL\_ACCT\_UPDATE\_VW

### 3.2.3 Hardship Views

The hardship views are as follows:

- FLX\_COL\_ACCT\_HRDSHIP\_VW
- FLX\_LN\_COL\_ACCT\_HRDSHIP\_VW
- FLX\_DD\_COL\_ACCT\_HRDSHIP\_VW

### 3.2.4 Party Views

The party views are as follows:

- FLX\_PI\_COL\_FD\_ACCT\_PER\_VW
- FLX\_PI\_COL\_FD\_PER\_VW
- FLX\_PI\_COL\_FD\_PARTY\_IDENT\_VW
- FLX\_PI\_COL\_FD\_PER\_NAME\_VW
- FLX\_PI\_COL\_FD\_PER\_WARN\_IND\_VW
- FLX\_PI\_COL\_FD\_EMP\_PROF\_VW
- FLX\_PI\_COL\_FD\_PER\_ADDR\_VW
- FLX\_PI\_COL\_FD\_CONTACT\_PREF\_VW

### 3.2.5 LCM / Collateral Views

The LCM / Collateral views are as follows:

- FLX\_LM\_COL\_FD\_COL\_ENTITY\_VW
- FLX\_LM\_COL\_FD\_COLLATERAL\_VW
- FLX\_LM\_COL\_FD\_COL\_PARTY\_VW
- FLX\_LM\_COL\_FD\_COL\_CHRG\_VW
- FLX\_LM\_COL\_FD\_COL\_GRNTR\_VW
- FLX\_LM\_COL\_FD\_INSR\_DTLS\_VW



## Algorithms

This chapter provides information about list of algorithm types shipped out for OBP Collections.

### 4.1 Stop Contract: C1-CURENTITY

This section provides details of the Stop Contract: C1-CURENTITY algorithm.

**Table 4–1 Stop Contract: C1-CURENTITY**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm type is used to stop the contract.   |
| <b>Detailed Description</b> | Contract Stop Algorithm   |
| <b>Algorithm Entity</b>     | Cure Entity   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.CureEntityAlgorithm   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | This algorithm invokes the C1-StopServiceAgreement business service to set contract status as STOPPED. The contract end date is specified as system date. |

### 4.2 Cure Account: C1-FINCOLL

This section provides details of the Cure Account: C1-FINCOLL algorithm.

**Table 4–2 Cure Account: C1-FINCOLL**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to invoke the OBP Services when contract is stopped during the finalize collection process.  |
| <b>Detailed Description</b> | This algorithm type is used to invoke the OBP Services to update the delinquent flag=N when the contract is stopped during the finalize collection process. |
| <b>Algorithm Entity</b>     | Contract Type - Contract Stop   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.FinalizeCollectionContractStopAlgoComp  |

**Table 4–2 (Cont.) Cure Account: C1-FINCOLL**

|                        |   |
|------------------------|---|
| <b>Parameters</b>      | <p><b>Name:</b> contactMethods</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Contact Methods soft parameter has a comma-separated value of customer contact methods. For example, SMS, EM, and so on.</p> <p>This value is used to calculate the number of self cured statistic.</p>  |
| <b>Detailed Design</b> | <p>This algorithm invokes the OBP Services to update the delinquent flag =N and In collection flag = N in host (updateInCollectionIndicator()) when the contract is stopped during the finalize collection process.</p> <p>It also deletes the account review date from CI_ADM_RVW_SCH table, and updates the number of times an account is self-cured.</p> |

**Table 4–3 Cure Account: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-FINCOL   |
| <b>Parameters</b>     | <p><b>Name:</b> contactMethods</p> <p><b>Value:</b> SMS, EM</p> |

## 4.3 Queue Allocation: C1-ALLOCQUEU

This section provides details of the Queue Allocation: C1-ALLOCQUEU algorithm.

**Table 4–4 Queue Allocation: C1-ALLOCQUEU**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Allocation algorithm for allocation cases to queue in round-robin method.  |
| <b>Detailed Description</b> | This is an allocation algorithm for the allocation group to allocate cases to queues in round-robin method. This algorithm is invoked by the Allocation monitor batch (C1-ALOCM).  |
| <b>Algorithm Entity</b>     | Allocation Group -Queue Allocation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | Com.splwg.ccb.domain.collection.batch.algorithm.AllocationGroupQueueAlgoComp   |
| <b>Parameters</b>           | <p><b>Name:</b> queueAllocationView (soft parameter)</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> View for allocation</p> <p><b>Name:</b> qallocationGroup (hard parameter)</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Allocation Group code</p>   |
| <b>Detailed Design</b>      | <p>This algorithm receives input as Allocation Group code from the batch.</p> <p>The view used to filter cases is accepted as an algorithm soft parameter. Product will ship CI_ALLOCATION_MONITOR_VW view.</p> <p>For the given allocation group code, it allocates cases to linked queues of the allocation group in round-robin method. For detailed process, see batch process (C1-ALOCM).</p> |

**Table 4–5 Queue Allocation: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-ALLOCQUEU   |
| <b>Parameters</b>     | <b>Name:</b> queueAllocationView<br><b>Value:</b> CL_ALLOCATION_MONITOR_VW |

## 4.4 Update Customer Switch: C1-CUSTSW

This section provides details of the Update Customer Switch: C1-CUSTSW algorithm.

**Table 4–6 Update Customer Switch: C1-CUSTSW**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to update the customer level case switch.  |
| <b>Detailed Description</b> | This algorithm is used to update customer level case status on case enter processing.<br>Customer Level Switch Name: Specify the customer level case status switch that should be updated.<br>For example, BANKRUPT_SW, HARDSHIP_SW, IMPRISONED_SW, DECEASED_SW, ABSCONDING_SW, and so on.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.CustomerLevelSwitchUpdateAlgorithm  |
| <b>Parameters</b>           | <b>Name:</b> Customer Level Switch Name<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Name of column or switch to be processed<br><br><b>Name:</b> Switch Value<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Y or N  |
| <b>Detailed Design</b>      | This algorithm updates the customer level switch. This algorithm is attached to the Case Type Enter Status algorithm spot. This soft parameter identifies the field that must be updated with a value.<br>The Customer Level switch name soft parameter accepts the column name that must be updated with switch values as Y or N.<br>You must create different algorithm for each field with the value and attach it to the case type enter status algorithm spot. |

**Table 4–7 Update Customer Switch: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-BRUPTSW   |
| <b>Parameters</b>     | <b>Name:</b> Customer Level Switch Name<br><b>Value:</b> BANKRUPT_SW<br><br><b>Name:</b> Switch Value<br><b>Value:</b> Y |

## 4.5 Update Legal/Repo Switch: C1-LEREPOCT

This section provides details of the Update Legal/Repo Switch: C1-LEREPOCT algorithm.

**Table 4–8 Update Legal/Repo Switch: C1-LEREPOCT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to update Legal and Repo case status on enter processing.   |
| <b>Detailed Description</b> | Legal Repo Switch Name: Specify the Legal or Repo case switch column name of account extension<br>For example, LEGAL_CASE_EXISTS_SW or REPO_CASE_EXISTS_SW, and so on.<br>Switch Value: Please enter the switch value as Y or N.   |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.RepoAndLegalCaseUpdateAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> Legal Repo Switch Name<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Name of column or switch to be processed</p> <p><b>Name:</b> Switch Value<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Y or N</p>  |
| <b>Detailed Design</b>      | <p>This algorithm is created to update the Legal Case Switch and Repo Case Switch derived fields. This algorithm is attached to the Case Type Enter Status algorithm spot.</p> <p>The soft parameter is used to identify the fields that should be updated.</p> <p>For example,</p> <ul style="list-style-type: none"> <li>■ If the case is Legal then pass Legal Repo Switch name as LEGAL_CASE_EXISTS_SW and switch value as Y and then attach this algorithm to case life cycle where you want to update the switch.</li> <li>■ If the case is Repo then pass Legal Repo Switch name as REPO_CASE_EXISTS_SW and switch value as Y and then attach this algorithm to the case life cycle where you want to update the switch.</li> </ul> |

**Table 4–9 Update Legal/Repo Switch: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-LEGALSW   |
| <b>Parameters</b>     | <p><b>Name:</b> Legal Repo Switch Name<br/><b>Value:</b> LEGAL_CASE_EXISTS_SW</p> <p><b>Name:</b> Switch Value<br/><b>Value:</b> Y</p> |

## 4.6 User Allocation - Round Robin: C1-USRALCRR

This section provides details of the User Allocation - Round Robin: C1-USRALCRR algorithm.

**Table 4–10 User Allocation - Round Robin: C1-USRALCRR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to allocate cases to users or teams in round-robin method.  |
| <b>Detailed Description</b> | This algorithm is used to allocate cases to user or teams in round-robin method. This algorithm is invoked by the User Allocation batch (C1-USALC).  |
| <b>Algorithm Entity</b>     | User Allocation  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationRoundRobinAlgorithm  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | <p>This algorithm receives input as queue code. The computation logic is explained below:</p> <ul style="list-style-type: none"> <li>■ A1 = Total allocation for the user or team across all queues.</li> <li>■ B1 = Total capacity of the user or team. This has to be defined in user or collection team configuration.</li> <li>■ <math>C1 = B1 - A1</math> = Total available capacity of the user or team.</li> <li>■ A2 = Existing allocation to the user or team for the current queue.</li> <li>■ B2 = Capacity of the user or team for the queue. This is defined in queue master.</li> <li>■ <math>C2 = B2 - A2</math> = Total available capacity of the user or team for the current queue.</li> <li>■ Available capacity of the user or team for the queue is lower of C1 and C2.</li> <li>■ Get all cases which are allocated to the queue and: <ul style="list-style-type: none"> <li>- Have no users or teams attached to it OR</li> <li>- Current allocated user or team does not have active association with the queue</li> </ul> </li> <li>■ Get available capacity for each user or team.</li> <li>■ Allocate cases to users or teams in a round-robin manner starting with user with highest available capacity and then in decreasing order of capacity.</li> <li>■ A count of freshly allocated cases should be maintained for each user or team.</li> <li>■ Allocation to a particular user will be skipped if the user is on leave.</li> <li>■ Allocation to a particular user or team will be skipped if count of newly allocated cases = available capacity.</li> <li>■ If capacity of all users and teams are exhausted and there are still cases pending allocation, these should be allocated to exception user. There will be no check for exception user's/team's capacity. Exception user's expiry date will be checked against SC_USR_GRP_USR table.</li> </ul> |

## 4.7 User Allocation - % Based: C1-USRALCPR

This section provides details of the User Allocation - % Based: C1-USRALCPR algorithm.

**Table 4–11 User Allocation - % Based: C1-USRALCPR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used for allocating cases to users or teams in percentage-based method.  |
| <b>Detailed Description</b> | This algorithm allocates cases to user or teams in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC). |
| <b>Algorithm Entity</b>     | User Allocation  |

**Table 4–11 (Cont.) User Allocation - % Based: C1-USRALCPR**

|                        |   |
|------------------------|---|
| <b>Program Type</b>    | java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPercentageBaseAlgorithm   |
| <b>Parameters</b>      | NA  |
| <b>Detailed Design</b> | <p>This algorithm takes input as Queue code. The computation logic is as below:</p> <ul style="list-style-type: none"> <li>■ A1 = Total allocation for the user or team across all queues.</li> <li>■ B1 = Total capacity of the user or team. This has to be defined in user or collection team configuration.</li> <li>■ C1 = B1 - A1 = Total available capacity of the user or team.</li> <li>■ Available capacity of the user or team for the queue is C1.</li> <li>■ Get all cases which are allocated to the queue and <ul style="list-style-type: none"> <li>- Have no users or teams attached to it OR</li> <li>- Current allocated user or team does not have active association with the queue</li> </ul> </li> <li>■ Calculate % allocation for each user or team in the queue to find maximum cases of new cases that can be allocated to each user or team.</li> <li>■ Get "available capacity" for each user or team</li> <li>■ Allocate cases to users or teams in sequential manner starting with user with highest available capacity and then in decreasing order of capacity.</li> <li>■ A count of freshly allocated cases should be maintained for each user or team</li> <li>■ Allocation to a particular user will be skipped if the user is on leave.</li> <li>■ Allocation to a particular user or team will be skipped if count of newly allocated cases = available capacity.</li> <li>■ If capacity of all users and teams are exhausted and there are still cases pending allocation, these should be allocated to exception user or team. There will be no check for exception user's capacity. Exception user's expiry date will be checked against SC_USR_GRP_USR table.</li> </ul> |

## 4.8 Vendor Allocation - Round Robin: C1-VENALCRR

This section provides details of the Vendor Allocation - Round Robin: C1-VENALCRR algorithm.

**Table 4–12 Vendor Allocation - Round Robin: C1-VENALCRR**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used for allocating cases to vendors in round-robin method.   |
| <b>Detailed Description</b> | This algorithm allocates cases to vendors in round-robin method. This algorithm is invoked from the User Allocation batch (C1-USALC). |
| <b>Algorithm Entity</b>     | Vendor Allocation   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationRoundRobinAlgorithm   |
| <b>Parameters</b>           | NA  |



**Table 4–12 (Cont.) Vendor Allocation - Round Robin: C1-VENALCRR**

|                        |  |
|------------------------|--|
| <b>Detailed Design</b> | <p>This algorithm takes input as Queue code. The computation logic for case capacity is as below:</p> <ul style="list-style-type: none"> <li>■ A1 = Total existing allocation for the vendor across all queues.</li> <li>■ B1 = Total capacity of the vendor. This has to be defined in vendor on boarding screen.</li> <li>■ C1 = B1 - A1 = Total available capacity of the vendor across all service types.</li> <li>■ A2 = Existing allocation of the vendor for the current queue.</li> <li>■ B2 = Capacity of the vendor for the queue. This is defined in queue master.</li> <li>■ C2 = B2 - A2 = Total available capacity of the vendor for the current queue.</li> <li>■ D1 = Available capacity for number of cases of the vendor for the queue is lower of C1 and C2.</li> <li>■ A3 = Existing allocation to the vendor for a service type attached to the vendor.</li> <li>■ B3 = Total capacity of the vendor for that service type. This is defined on vendor on boarding screen under section 'Associated Service Types'. If the value is blank then do not calculate capacity (C3) for that service type.</li> <li>■ C3 = B3 - A3 = Total available capacity for number of cases for a vendor service type. Repeat above steps for each service type attached to the vendor.</li> <li>■ Available capacity for number of cases for the vendor for a service type attached to the vendor for the queue is lower of D1 and C3. If C3 is not available for a service type then D1 should be considered as capacity.</li> <li>■ Get all cases which are allocated to the queue and: <ul style="list-style-type: none"> <li>- Have no vendors attached to it OR</li> <li>- Current allocated vendor does not have active association with the queue.</li> </ul> </li> <li>■ Get "available capacity" of cases of each vendor for each service type attached (A).</li> <li>■ Get "available capacity" of OS amount of each vendor for each service type attached (B).</li> <li>■ Allocate cases to vendor in a round-robin manner starting with vendor with highest available capacity of number of cases for that queue (see D1 in round-robin based capacity calculation) and then in decreasing order of capacity.</li> <li>■ For every case to be allocated system should check that case type of the case matches with case type of the service types attached with vendor. Match found: <ul style="list-style-type: none"> <li>- Yes: Allocate if count of newly allocated cases for that service type and OS balance of newly allocated cases for that service type &lt; A and B respectively. If value for B is blank then ignore validating it.</li> <li>- No: Move to next vendor in queue.</li> </ul> </li> <li>■ A count of freshly allocated cases should be maintained for each vendor.</li> <li>■ Allocation to a particular vendor will be skipped if count of newly allocated cases for that service type or OS balance of newly allocated cases for that service type = A or B respectively.</li> <li>■ All cases for which case type does not match with case type of the service types attached with any vendor in the queue will be kept allocated at queue level only. These cases should not be allocated to exception user or team.</li> <li>■ If capacity of all vendors is exhausted and there are still cases pending allocation, these should be allocated to exception user or team. There will be no check for exception user's capacity. Exception user's expiry date will be checked against SC_USR_GRP_USR table.</li> </ul> |
|------------------------|--|

## 4.9 Vendor Allocation - % Based: C1-VENALCPR

This section provides details of the Vendor Allocation - % Based: C1-VENALCPR algorithm.

**Table 4–13 Vendor Allocation - % Based: C1-VENALCPR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used for allocating cases to vendors in percentage-based method.   |
| <b>Detailed Description</b> | This algorithm allocates cases to vendors in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC). |
| <b>Algorithm Entity</b>     | Vendor Allocation  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationPercentageBaseAlgorithm  |
| <b>Parameters</b>           | NA   |

**Table 4–13 (Cont.) Vendor Allocation - % Based: C1-VENALCRR**

|                 |  |
|-----------------|--|
| Detailed Design | <p>This algorithm takes input as Queue code. The computation logic for case capacity is as below:</p> <ul style="list-style-type: none"> <li>■ A1 = Total existing allocation for the vendor across all queues.</li> <li>■ B1 = Total capacity of the vendor. This has to be defined in vendor on boarding screen.</li> <li>■ C1 = B1 - A1 = Total available capacity of the vendor across all service types.</li> <li>■ D1 = Available capacity for no. of cases of the vendor for the queue is C1.</li> <li>■ A3 = Existing allocation to the vendor for a service type attached to the vendor.</li> <li>■ B3 = Total capacity of the vendor for that service type. This is defined on vendor on boarding screen under section 'Associated Service Types'. If the value is blank then do not calculate capacity (C3) for that service type.</li> <li>■ C3 = B3 - A3 = Total available capacity for number of cases for a vendor service type. Repeat above steps for each service type attached to the vendor.</li> <li>■ Available capacity for number of cases for the vendor, for a service type attached to the vendor for the queue is lower of D1 and C3. If C3 is not available for a service type then D1 should be considered as capacity.</li> <li>■ Get all cases which are allocated to the queue and <ul style="list-style-type: none"> <li>- Have no vendors attached to it OR</li> <li>- Current allocated vendor does not have active association with the queue.</li> </ul> </li> <li>■ Calculate % allocation for each vendor in the queue to find maximum cases of new cases that can be allocated to each vendor.</li> <li>■ Get "available capacity" of cases of each vendor for each service type attached (A).</li> <li>■ Get "available capacity" of OS amount of each vendor for each service type attached (B).</li> <li>■ Allocate cases to vendor in a sequential manner starting with vendor with highest available capacity of number of cases for that queue (see D1 in % based capacity calculation) and then in decreasing order of capacity.</li> <li>■ For every case to be allocated system should check that case type of the case matches with case type of the service types attached with vendor. Match found: <ul style="list-style-type: none"> <li>-Yes: Allocate if count of newly allocated cases for that service type and OS balance of newly allocated cases for that service type &lt; A and B respectively. If value for B is blank then ignore validating it</li> <li>- No: Move to next vendor in queue.</li> </ul> </li> <li>■ A count of freshly allocated cases should be maintained for each vendor.</li> <li>■ Allocation to a particular vendor will be skipped if count of newly allocated cases for that service type or OS balance of newly allocated cases for that service type = A or B respectively.</li> <li>■ All cases for which case type does not match with case type of the service types attached with any vendor in the queue will be kept allocated at queue level only. These cases should not be allocated to exception user or team.</li> <li>■ If capacity of all vendors is exhausted and there are still cases pending allocation, these should be allocated to exception user. There will be no check for exception user's capacity. Exception user's expiry date will be checked against SC_USR_GRP_USR table.</li> </ul> |
|-----------------|--|

## 4.10 Bulk Contact Creation: C1-BLKNTCRE

This section provides details of the Bulk Contact Creation: C1-BLKNTCRE algorithm.

**Table 4–14 Bulk Contact Creation: C1-BLKNTCRE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used for allocating cases to vendors in percentage-based method.  |
| <b>Detailed Description</b> | This algorithm allocates cases to vendors in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC).  |
| <b>Algorithm Entity</b>     | Bulk contact creation   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.BulkContactCreationAlgoComp   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | <p>This algorithm will be invoked from bulk contact creation batch from where the hard parameter values are set.</p> <p>The algorithm will call business service 'C1-GenerateCorrespondence'.</p> <p>addMultiple() method of 'C1-GenerateCorrespondence' will be called which in turn adds customer contact to CI_CC via add () method of the same service.</p> |

## 4.11 Cross Strategy Action Matrix: C1-CSAM

This section provides details of the Cross Strategy Action Matrix: C1-CSAM algorithm.

**Table 4–15 Cross Strategy Action Matrix: C1-CSAM**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used for Cross Strategy Action Matrix   |
| <b>Detailed Description</b> |   |
| <b>Algorithm Entity</b>     | Case Type- Enter status   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.CrossStrategyActionMatrixAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> CheckStatus</p> <p><b>Required (Yes/No):</b> N</p> <p><b>Description:</b> Y - Case types with Status</p> <p>N - Case types without status</p>   |
| <b>Detailed Design</b>      | <p>This algorithm will refer the CSAM admin configuration for case types and decide what action is to be taken for open case available on the entity being worked upon. It will also consider associated entity cases on the entity being worked upon.</p> <p>The two possible actions are:</p> <ul style="list-style-type: none"> <li>■ <b>Close the case:</b> Case status will be moved to next final status or the one with default switch. Business service to close the case (change case status) will be called. This action will not cure the account though. TO DO (TO DO type: C1-CSAM) will be created for the case if no final status is found for the case type or if case cannot be closed due to some other error.</li> <li>■ <b>Hold the case:</b> The business service for holding a case will be called. Hold expiry date will be set to a default value of 01-01-2100. Hold reason flag will be "CSAM".</li> </ul> <p>This algorithm should also get triggered during case association process.</p> |

**Table 4–16 Cross Strategy Action Matrix: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-CSAMY                                    |
| <b>Parameters</b>     | <b>Name:</b> CheckStatus<br><b>Value:</b> Y |

## 4.12 Last Payment for Account: C1-PAYDTAMTU

This section provides details of the Last Payment for Account: C1-PAYDTAMTU algorithm.

**Table 4–17 Last Payment for Account: C1-PAYDTAMTU**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to update last payment date and amount in account extension table.  |
| <b>Detailed Description</b> | This algorithm will be invoked on FT freeze algorithm spot and will update Last Payment date and amount in account extension table.                                      |
| <b>Algorithm Entity</b>     | Customer class - FT Freeze   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.LastPaymentDtAmtUpdateAlgorithm  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | It is invoked when the FT is frozen for payment. Algorithm will update the FT amount and FT date in Account extension table column LAST_PAYMENT_AMT and LAST_PAYMENT_DT. |

## 4.13 Association Review Check: C1-ASORVCHK

This section provides details of the Association Review Check: C1-ASORVCHK algorithm.

**Table 4–18 Association Review Check: C1-ASORVCHK**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to check if association review is required.   |
| <b>Detailed Description</b> | This is to decide if the user should review the system association of entities or not. If Association Review is Required - Stay in current status for user review. Set display date to current business date.<br>If association Review is not required then transition to specified next status. |
| <b>Algorithm Entity</b>     | Case Enter Validation  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal  |

**Table 4–18 (Cont.) Association Review Check: C1-ASORVCHK**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> NextStatus<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> AssociationReviewRequired<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p>                   |
| <b>Detailed Design</b> | It is invoked in the pending status of Legal Process. It decides whether the user should review the system association of entities or not. 'Y' in the algorithm parameter specifies that Association review is required. |

**Table 4–19 Association Review Check: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-ASORVCHK  |
| <b>Parameters</b>     | <p><b>Name:</b> NextStatus<br/> <b>Value:</b> ASSNEWLSP</p> <p><b>Name:</b> AssociationReviewRequired<br/> <b>Value:</b> Y</p> |

## 4.14 Validate Expired Default Notice: C1-DEFNOEXP

This section provides details of the Validate Expired Default Notice: C1-DEFNOEXP algorithm.

**Table 4–20 Validate Expired Default Notice: C1-DEFNOEXP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to validate expired default notices.   |
| <b>Detailed Description</b> | This algorithm returns an error if there is no default notice on a given account or a default notice has not yet expired. |
| <b>Algorithm Entity</b>     | Case Type - Enter Validation  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |

**Table 4–20 (Cont.) Validate Expired Default Notice: C1-DEFNOEXP**

|                        |   |
|------------------------|---|
| <b>Parameters</b>      | <p><b>Name:</b> associationType<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> validationfailureOption<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> toDoType<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> toDoRole<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in the pending status of the Legal Process case. It checks if the default notice has expired for a particular account.  |

**Table 4–21 Validate Expired Default Notice: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-DEFNOEXP  |
| <b>Parameters</b>     | <p><b>Name:</b> associationType<br/><b>Value:</b> P</p> <p><b>Name:</b> validationfailureOption<br/><b>Value:</b> N</p> <p><b>Name:</b> toDoType<br/><b>Value:</b> C1-TD-DN</p> <p><b>Name:</b> toDoRole<br/><b>Value:</b></p> |

## 4.15 Associate Related Entity: C1-ASSOENTY

This section provides details of the Associate Related Entity: C1-ASSOENTY algorithm.

**Table 4–22 Associate Related Entity: C1-ASSOENTY**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to associate related entities with the case. |
| <b>Detailed Description</b> | This algorithm pulls the related entities associated with the case. |
| <b>Algorithm Entity</b>     | Case Type - Enter Validation  |

**Table 4–22 (Cont.) Associate Related Entity: C1-ASSOENTY**

|                        |   |
|------------------------|---|
| <b>Program Type</b>    | java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>      | <p><b>Name:</b> hostId<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> toDoType<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> toDoRole<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in the pending state of the Legal Case process. The algorithm associates the primary account with the persons attached to it and also the accounts which have the same set of financially responsible customers as in the primary account.                    |

**Table 4–23 Associate Related Entity: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-ASSOENTY  |
| <b>Parameters</b>     | <p><b>Name:</b> hostId<br/> <b>Value:</b> NGP</p> <p><b>Name:</b> toDoType<br/> <b>Value:</b> C1-TD-AC</p> <p><b>Name:</b> toDoRole<br/> <b>Value:</b></p> |

## 4.16 Validate Legal Case Exists: C1-CHKLGL

This section provides details of the Validate Legal Case Exists: C1-CHKLGL algorithm.

**Table 4–24 Validate Legal Case Exists: C1-CHKLGL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to validate if an active legal case exists at the same time.  |
| <b>Detailed Description</b> | This algorithm checks if a legal case is already running on the primary account any account in the collection with the same owner. |
| <b>Algorithm Entity</b>     | Case Enter Validation  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal  |



**Table 4–24 (Cont.) Validate Legal Case Exists: C1-CHKLGL**

|                        |   |
|------------------------|---|
| <b>Parameters</b>      | <b>Name:</b> Case Category<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> NA<br><br><b>Name:</b> toDoType<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> NA<br><br><b>Name:</b> toDoRole<br><b>Required (Yes/No):</b> N<br><b>Description:</b> NA |
| <b>Detailed Design</b> | It is invoked in the pending state of the Legal Process case. It checks if there is any legal case running on the primary account or its related entities.  |

**Table 4–25 Validate Legal Case Exists: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-ASSOENTY   |
| <b>Parameters</b>     | <b>Name:</b> Case Category<br><b>Value:</b> LEGL<br><br><b>Name:</b> toDoType<br><b>Value:</b> C1-TD-CL<br><br><b>Name:</b> toDoRole<br><b>Value:</b> |

## 4.17 Assign New LSP: C1-ASGNLSP

This section provides details of the Assign New LSP: C1-ASGNLSP algorithm.

**Table 4–26 Assign New LSP: C1-ASGNLSP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to assign LSP to the case.   |
| <b>Detailed Description</b> | This algorithm assigns the LSP to the case either automatically or let the user assign manually depending on the value entered in the algorithm parameters. |
| <b>Algorithm Entity</b>     | Case Enter Status   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |

**Table 4–26 (Cont.) Assign New LSP: C1-ASGNLSP**

|                        |   |
|------------------------|---|
| <b>Parameters</b>      | <p><b>Name:</b> New Allocation And Review Option<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> Change LSP Allocation Option<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> Reset Document Submission Date<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> Previous Allocation Check<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> Next Status<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in the Assign New LSP status of the Legal Process case. Depending on the different algorithm parameter values, the LSP is assigned automatically or manually(both in cases of First time assignment or change assignment).  |

**Table 4–27 Assign New LSP: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-ASGNLSP   |
| <b>Parameters</b>     | <p><b>Name:</b> New Allocation And Review Option<br/><b>Value:</b> AUTO_WITH_REVIEW_PRVALLOC</p> <p><b>Name:</b> Change LSP Allocation Option<br/><b>Value:</b> AUTO_WITH_REVIEW</p> <p><b>Name:</b> Reset Document Submission Date<br/><b>Value:</b> N</p> <p><b>Name:</b> Previous Allocation Check<br/><b>Value:</b> Y</p> <p><b>Name:</b> Next Status<br/><b>Value:</b> PREPLGLDOC</p> |

## 4.18 Check Approval Requirement: C1-APPRCHK

This section provides details of the Check Approval Requirement: C1-APPRCHK algorithm.

**Table 4–28 Check Approval Requirement: C1-APPRCHK**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to check the need of approval.   |
| <b>Detailed Description</b> | This algorithm checks if LSP assignments should be approved.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Processing  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>           | <p><b>Name:</b> Exposure Threshold<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> Approval Request Status<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> approvedStatus<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> rejectRequestStatus<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b>      | It is invoked in the Prepare Legal Documents status of the Legal Process Case. It checks if the approval is required for the LSP assignment depending on the algorithm parameter values. It also decides where to transit the case.   |

**Table 4–29 Check Approval Requirement: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-ASGNLSP   |
| <b>Parameters</b>     | <p><b>Name:</b> Exposure Threshold<br/> <b>Value:</b> 10</p> <p><b>Name:</b> Approval Request Status<br/> <b>Value:</b> PENDINGAPP</p> <p><b>Name:</b> approvedStatus<br/> <b>Value:</b> WTFRLSPACK</p> <p><b>Name:</b> rejectRequestStatus<br/> <b>Value:</b> ASSNEWLSP</p> |

## 4.19 Save the Status Before Change LSP: C1-SAVESTATUS

This section provides details of the Save the Status Before Change LSP: C1-SAVESTATUS algorithm.

**Table 4–30 Save the Status Before Change LSP: C1-SAVESTATUS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to save the status before the status changes in LSP.  |
| <b>Detailed Description</b> | This algorithm saves the status from where it came to Change LSP status. This will be stored in CI_LSP_DTLS table.   |
| <b>Algorithm Entity</b>     | Case Type-Enter Processing   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | It is invoked in the Change or Retire LSP status of the Legal Process Case. It stores the previous state of the case so that it returns to that state after the LSP for the case is changed. |

## 4.20 Resume Status from Previous LSP: C1-RESSTATUS

This section provides details of the Resume Status from Previous LSP: C1-RESSTATUS algorithm.

**Table 4–31 Resume Status from Previous LSP: C1-RESSTATUS**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to resume status from previous LSP.  |
| <b>Detailed Description</b> | This algorithm resumes the previous state stored while changing LSP.  |
| <b>Algorithm Entity</b>     | Customer class - FT Freeze  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | It is invoked in the Legal in Progress status of the Legal Process Case. It resumes the status where the case was previously in before changing the LSP for the case. |

## 4.21 Check Submission Date: C1-CHKSUBDT1

This section provides details of the Check Submission Date: C1-CHKSUBDT1 algorithm.

**Table 4–32 Check Submission Date: C1-CHKSUBDT1**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to check submission date.  |
| <b>Detailed Description</b> | This algorithm checks if the document submission date is filled from screen. If it is present, the case is auto transitioned to 'WAIT FOR LSP ACKNOWLEDGMENT' status directly from 'ASSIGN NEW LSP' status. |
| <b>Algorithm Entity</b>     | Case Auto Transition Validation   |

**Table 4–32 (Cont.) Check Submission Date: C1-CHKSUBDT1**

|                        |   |
|------------------------|---|
| <b>Program Type</b>    | java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>      | <p><b>Name:</b> nextStatus<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> changeStatus<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p>   |
| <b>Detailed Design</b> | It is invoked in the Prepare Legal Documents status of the Legal Process case. This algorithm checks for the presence of document submission date in the database. If document submission date is present in the database, then based on the soft parameter it will transition the case to next status. |

**Table 4–33 Check Submission Date: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-CHKSUBDT1   |
| <b>Parameters</b>     | <p><b>Name:</b> nextStatus<br/> <b>Value:</b> WTFRLSPACK</p> <p><b>Name:</b> changeStatus<br/> <b>Value:</b> Y</p> |

## 4.22 Update LSP (CLOS): C1-LSPSTATUS

This section provides details of the Update LSP (CLOS): C1-LSPSTATUS algorithm.

**Table 4–34 Update LSP (CLOS): C1-LSPSTATUS**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Legal Proceedings - Update Status   |
| <b>Detailed Description</b> | This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either closed or cancelled.  |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>           | <p><b>Name:</b> lspAssignmentStatus<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p>  |
| <b>Detailed Design</b>      | It is invoked in the Complete, Withdraw status of the Legal Process case. This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either completed or withdrawn. |

**Table 4–35 Update LSP (CLOS): Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-LSPSTATUS   |
| <b>Parameters</b>     | <b>Name:</b> lspAssignmentStatus<br><b>Value:</b> CLOS |

## 4.23 Update LSP (CANCEL): C1-LSPSTACAN

This section provides details of the Update LSP (CANCEL): C1-LSPSTACAN algorithm.

**Table 4–36 Update LSP (CANCEL): C1-LSPSTACAN**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Legal Proceedings - Update Status   |
| <b>Detailed Description</b> | This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either closed or cancelled.  |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>           | <b>Name:</b> lspAssignmentStatus<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> NA   |
| <b>Detailed Design</b>      | It is invoked in the CANCEL status of the Legal Process case. This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is cancelled. |

**Table 4–37 Update LSP (CANCEL): Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-LSPSTACAN  |
| <b>Parameters</b>     | <b>Name:</b> lspAssignmentStatus<br><b>Value:</b> CAN |

## 4.24 Validate Expired Default Notice: C1-DEFNOTEXP

This section provides details of the Validate Expired Default Notice: C1-DEFNOTEXP algorithm.

**Table 4–38 Validate Expired Default Notice: C1-DEFNOTEXP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate Expired Default Notice   |
| <b>Detailed Description</b> | This algorithm returns an error if there is no default notice on a given account or a default notice has not yet expired. |
| <b>Algorithm Entity</b>     | Case Type - Enter Validation  |
| <b>Program Type</b>         | java  |

**Table 4–38 (Cont.) Validate Expired Default Notice: C1-DEFNOTEXP**

|                        |   |
|------------------------|---|
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal   |
| <b>Parameters</b>      | <p><b>Name:</b> associationType<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> validationfailureOption<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> toDoType<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> toDoRole<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in the Assign New LSP status of the Legal Process case. It checks if the default notice has expired for a particular account.   |

**Table 4–39 Validate Expired Default Notice: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-DEFNOTEXP   |
| <b>Parameters</b>     | <p><b>Name:</b> associationType<br/> <b>Value:</b> A</p> <p><b>Name:</b> validationfailureOption<br/> <b>Value:</b> Y</p> <p><b>Name:</b> toDoType<br/> <b>Value:</b> C1-TD-DN</p> <p><b>Name:</b> toDoRole<br/> <b>Value:</b></p> |

## 4.25 Collateral Verification: C1-VRFYCOLS

This section provides details of the Collateral Verification: C1-VRFYCOLS algorithm.

**Table 4–40 Collateral Verification: C1-VRFYCOLS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collateral Verification  |
| <b>Detailed Description</b> | <p>This will perform following validations for the collateral with the case:</p> <ul style="list-style-type: none"> <li>▪ If the soft parameter for Collateral type to this algorithm type is "PROPERTY", then one collateral is associated with the case and that Collateral is associated with Facility for the primary account associated with the case.</li> <li>▪ If collateral type soft parameter is blank, then above validation should be ignored and Collateral status is set to Not Sold.</li> <li>▪ It will also validate that if there is not active Asset repossession case running for the collateral. If any of the above validations fail, case creation process should be terminated.</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification   |
| <b>Parameters</b>           | <p><b>Name:</b> Collateral Type</p> <p><b>Required (Yes/No):</b> N</p> <p><b>Description:</b> NA</p>   |
| <b>Detailed Design</b>      | It is invoked in the Pending status of the Asset Repossession Process case. It Verifies the collateral associated with account.  |

**Table 4–41 Collateral Verification: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-VRFYCOLS   |
| <b>Parameters</b>     | <p><b>Name:</b> Collateral Type</p> <p><b>Value:</b> PROPERTY</p> |

## 4.26 Account Association for Asset Repossession Case: C1-ARSACCTS

This section provides details of the Account Association for Asset Repossession Case: C1-ARSACCTS algorithm.

**Table 4–42 Account Association for Asset Repossession Case: C1-ARSACCTS**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Account Association for Asset repossession case   |
| <b>Detailed Description</b> | <p>This algorithm will perform following actions:</p> <ul style="list-style-type: none"> <li>▪ It gets all facilities to which this collateral is associated and all accounts for these facilities.</li> <li>▪ It associates these accounts with the case.</li> </ul> <p>Scope of this association is limited to accounts already in collections. This process will not check for any accounts not in collections.</p> <p>This algorithm doesn't have any soft parameter.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |



**Table 4–42 (Cont.) Account Association for Asset Repossession Case: C1-ARSACCTS**

|                        |  |
|------------------------|--|
| <b>Program Type</b>    | java   |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AccountAssociationForAssetRepossessionCase           |
| <b>Parameters</b>      | NA   |
| <b>Detailed Design</b> | It is invoked in the Pending status of the Asset Repossession Process case. It will associate facilities of account with case. |

## 4.27 Customer Association for Asset Repossession Case: C1-ARSCUSTS

This section provides details of the Customer Association for Asset Repossession Case: C1-ARSCUSTS algorithm.

**Table 4–43 Customer Association for Asset Repossession Case: C1-ARSCUSTS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Customer Association for Asset repossession case   |
| <b>Detailed Description</b> | <p>This algorithm performs the following actions:</p> <ul style="list-style-type: none"> <li>■ It gets all customers who are the owners for the selected collateral</li> <li>■ It associates these customers with the case</li> </ul> <p>Scope of this association is limited to customers already in collections. This process will not check for any customers not in collections.</p> <p>This algorithm does not have any soft parameter.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CustomerAssociationForAssetRepossessionCase  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | It is invoked in the Pending status of the Asset Repossession Process case. It will associate facilities of customer with case.  |

## 4.28 Update Collateral Property: C1-SETCOLLPR

This section provides details of the Update Collateral Property: C1-SETCOLLPR algorithm.

**Table 4–44 Update Collateral Property: C1-SETCOLLPR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Property   |
| <b>Detailed Description</b> | <p>This algorithm will perform following operations:</p> <ul style="list-style-type: none"> <li>■ If the value of updateCollateralProperty soft parameter is "SET" and type of possession is "Warrant" then Fetch the collateral for which case is created and update the IS_LEGAL_SW= "Y" and populate the case_ID on this collateral.</li> <li>■ If the value of updateCollateralProperty soft parameter is "RESET" then Fetch the collateral for which case is created and update the IS_LEGAL_SW= "N" and IS_REPO_SW= "N" nullify the case_ID on this collateral.</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |

**Table 4–44 (Cont.) Update Collateral Property: C1-SETCOLLPR**

|                        |   |
|------------------------|---|
| <b>Program Type</b>    | java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralProperty  |
| <b>Parameters</b>      | <b>Name:</b> UpdateCollateralProperty<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> NA  |
| <b>Detailed Design</b> | It is invoked in the Pending status of the Asset Repossession Process case. It updates the collateral Properties like IS_LEGAL_SW, IS_REPO_SW depending on user inputs. |

## 4.29 Close To do's Algorithm: C1-CLSTODOA

This section provides details of the Close To do's Algorithm: C1-CLSTODOA algorithm.

**Table 4–45 Close To do's Algorithm: C1-CLSTODOA**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Close To do's algorithm   |
| <b>Detailed Description</b> | This process will close all To-Do's of specific To-do types associated with the case. Up to five To-Do types can be given to this algorithm to close. |
| <b>Algorithm Entity</b>     | Case Type-Exit Status   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo   |

**Table 4–45 (Cont.) Close To do's Algorithm: C1-CLSTODOA**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> To Do Type1<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type2<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type3<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type4<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type5<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked while exiting from Pending status of the Asset Repossession Process case. This process will close all To-Do's of "No activity" To-do types associated with the case.   |

**Table 4–46 Close To do's Algorithm: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-ARSCUSTS   |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Type1<br/><b>Value:</b> C1-ANA1</p> <p><b>Name:</b> To Do Type2<br/><b>Value:</b> C1-ANA2</p> <p><b>Name:</b> To Do Type3<br/><b>Value:</b></p> <p><b>Name:</b> To Do Type4<br/><b>Value:</b></p> <p><b>Name:</b> To Do Type5<br/><b>Value:</b></p> |

## 4.30 Validations for Mandatory Characteristics: C1-CHARVALU

This section provides details of the Validations for Mandatory Characteristics: C1-CHARVALU algorithm.

**Table 4–47 Validations for Mandatory Characteristics: C1-CHARVALU**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validations for Mandatory Characteristics   |
| <b>Detailed Description</b> | Subjective Validations for Mandatory Characteristics  |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics  |
| <b>Parameters</b>           | <p><b>Name:</b> ReferenceCharacteristicsValue<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> ReferenceCharacteristic<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics1<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics2<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics3<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics4<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics5<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b>      | It is invoked in Effected Possession status of the Asset Repossession Process case. This algorithm will carry out subjective validation based on the type of input.   |

**Table 4–48 Validations for Mandatory Characteristics: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-CHARVALU   |
| <b>Parameters</b>     | <p><b>Name:</b> ReferenceCharacteristicsValue<br/><b>Value:</b> Type of Possession</p> <p><b>Name:</b> ReferenceCharacteristic<br/><b>Value:</b> Voluntary Possession</p> <p><b>Name:</b> CaseCharacteristics1<br/><b>Value:</b> Vacancy Date</p> <p><b>Name:</b> CaseCharacteristics2<br/><b>Value:</b> Vacancy Possession Indemnity Policy Reference</p> <p><b>Name:</b> CaseCharacteristics3<br/><b>Value:</b> Property Surrender Letter Reference</p> <p><b>Name:</b> CaseCharacteristics4<br/><b>Value:</b> Property Surrender Letter Reference</p> <p><b>Name:</b> CaseCharacteristics5<br/><b>Value:</b></p> |

### 4.31 Validations for Mandatory Characteristics: C1-CHARVALA

This section provides details of the Validations for Mandatory Characteristics: C1-CHARVALA algorithm.

**Table 4–49 Validations for Mandatory Characteristics: C1-CHARVALA**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validations for Mandatory Characteristics  |
| <b>Detailed Description</b> | Subjective Validations for Mandatory Characteristics   |
| <b>Algorithm Entity</b>     | Case Type-Exit Status  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics |

**Table 4–49 (Cont.) Validations for Mandatory Characteristics: C1-CHARVALA**

|                               |   |
|-------------------------------|---|
| <p><b>Parameters</b></p>      | <p><b>Name:</b> ReferenceCharacteristicsValue<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> ReferenceCharacteristic<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics1<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics2<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics3<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics4<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics5<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <p><b>Detailed Design</b></p> | <p>It is invoked in Effected Possession status of the Asset Repossession Process case. This algorithm will carry out subjective validation based on the type of input.</p>  |

**Table 4–50 Validations for Mandatory Characteristics: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-CHARVALU  |
| <b>Parameters</b>     | <p><b>Name:</b> ReferenceCharacteristicsValue<br/><b>Value:</b> Type of Possession</p> <p><b>Name:</b> ReferenceCharacteristic<br/><b>Value:</b> Voluntary Possession</p> <p><b>Name:</b> CaseCharacteristics1<br/><b>Value:</b> Legal Case ID</p> <p><b>Name:</b> CaseCharacteristics2<br/><b>Value:</b></p> <p><b>Name:</b> CaseCharacteristics3<br/><b>Value:</b></p> <p><b>Name:</b> CaseCharacteristics4<br/><b>Value:</b></p> <p><b>Name:</b> CaseCharacteristics5<br/><b>Value:</b></p> |

## 4.32 Update Collateral Status in the Host: C1-UPCOLLSTZ

This section provides details of the Update Collateral Status in the Host: C1-UPCOLLSTZ algorithm.

**Table 4–51 Update Collateral Status in the Host: C1-UPCOLLSTZ**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Status in the host   |
| <b>Detailed Description</b> | <p>This process updates the collateral status in the host. The value of status to be set will be passed as parameter to the process.</p> <p>If the update fails for any reason, system should create a To-do. Message for the To-do should be configured based on type of update which failed.</p> <p>To-do should be assigned to the To-do Role set as parameter to this process. If the parameter is left blank, To-do should be assigned to the default role.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost  |

**Table 4–51 (Cont.) Update Collateral Status in the Host: C1-UPCOLLSTZ**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> To Do Role<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> Collateral Status<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in Effected Possession status of the Asset Repossession Process case. This process will update the collateral status in the host.  |

**Table 4–52 Update Collateral Status in the Host: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-UPCOLLSTZ   |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Role<br/> <b>Value:</b></p> <p><b>Name:</b> To Do Type<br/> <b>Value:</b> C1-TD-UC</p> <p><b>Name:</b> Collateral Status<br/> <b>Value:</b> Sold</p> |

### 4.33 Initiate Collateral Valuation: C1-COLLVALX

This section provides details of the Initiate Collateral Valuation: C1-COLLVALX algorithm.



**Table 4–53 Initiate Collateral Valuation: C1-COLLVALX**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Status in the host   |
| <b>Detailed Description</b> | <p>This algorithm works as follows:</p> <p>System should check if "X" days have elapsed since the last assessment was done for the collateral. That is check if (Assessment date + X) &lt;= Current business date. Number of days, X, will be set as Assessment Expiry Days parameter for this process.</p> <p>If yes - Create a To-do to alert the user that collateral valuation is required. This to-do should be associated with the case. To-do Type is passed as a parameter to the process.</p> <p>However, To-do should not be created if:</p> <ul style="list-style-type: none"> <li>■ A To-do of same to-do type is already open for the case</li> <li>■ A To-do of same to-do type was closed within past "Y" days</li> </ul> <p>To-do should be assigned to the To-do Role set as parameter to this process. If the parameter is left blank, To-do should be assigned to the default role.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Role<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> Days Since Closure Of Last To Do<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> Assessment Expiry Days<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p>  |
| <b>Detailed Design</b>      | It is invoked while exiting from Pending status of the Asset Repossession Process case. This process will close all To-Do's of "Asset repossession No activity" To-do types associated with the case.  |

**Table 4–54 Initiate Collateral Valuation: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-COLLVALX   |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Role<br/><b>Value:</b> C1-ASSETRE</p> <p><b>Name:</b> To Do Type<br/><b>Value:</b> C1-TD-UC</p> <p><b>Name:</b> Days Since Closure Of Last To Do<br/><b>Value:</b> 5</p> <p><b>Name:</b> Assessment Expiry Days<br/><b>Value:</b> 5</p> |

## 4.34 Close To do's Algorithm: C1-CLSTODOV

This section provides details of the Close To do's Algorithm: C1-CLSTODOV algorithm.

**Table 4–55 Close To do's Algorithm: C1-CLSTODOV**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Close To do's algorithm   |
| <b>Detailed Description</b> | This process will close all To-Do's of specific To-do types associated with the case. Up to five To-Do types can be given to this algorithm to close. |
| <b>Algorithm Entity</b>     | Case Type-Exit Status   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo   |

**Table 4–55 (Cont.) Close To do's Algorithm: C1-CLSTODOV**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> To Do Type1<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type2<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type3<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type4<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type5<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked while exiting from Sale In-Progress status of the Asset Repossession Process case. This process will close all To-Do's of "No activity" To-do types associated with the case.  |

**Table 4–56 Close To do's Algorithm: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-CLSTODOV  |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Type1<br/><b>Value:</b> C1-LNA1</p> <p><b>Name:</b> To Do Type2<br/><b>Value:</b> C1-LNA1</p> <p><b>Name:</b> To Do Type3<br/><b>Value:</b> C1-TD-CV</p> <p><b>Name:</b> To Do Type4<br/><b>Value:</b></p> <p><b>Name:</b> To Do Type5<br/><b>Value:</b></p> |

## 4.35 Validations for Mandatory Characteristics: C1-CHARVALC

This section provides details of the Validations for Mandatory Characteristics: C1-CHARVALC algorithm.

**Table 4–57 Validations for Mandatory Characteristics: C1-CHARVALC**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validations for Mandatory Characteristics   |
| <b>Detailed Description</b> | Subjective Validations for Mandatory Characteristics  |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics  |
| <b>Parameters</b>           | <p><b>Name:</b> ReferenceCharacteristicsValue<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> ReferenceCharacteristic<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics1<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics2<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics3<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics4<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> CaseCharacteristics5<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b>      | It is invoked in Settlement status of the Asset Repossession Process case. This algorithm will carry out subjective validation based on the type of input.  |

**Table 4–58 Validations for Mandatory Characteristics: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-CHARVALU   |
| <b>Parameters</b>     | <p><b>Name:</b> ReferenceCharacteristicsValue<br/><b>Value:</b> Type of Possession</p> <p><b>Name:</b> ReferenceCharacteristic<br/><b>Value:</b> Voluntary Possession</p> <p><b>Name:</b> CaseCharacteristics1<br/><b>Value:</b> Contactor Details</p> <p><b>Name:</b> CaseCharacteristics2<br/><b>Value:</b> Conveyance Details</p> <p><b>Name:</b> CaseCharacteristics3<br/><b>Value:</b></p> <p><b>Name:</b> CaseCharacteristics4<br/><b>Value:</b></p> <p><b>Name:</b> CaseCharacteristics5<br/><b>Value:</b></p> |

## 4.36 Update Collateral Status in the Host: C1-UPCOLLSTX

This section provides details of the Update Collateral Status in the Host: C1-UPCOLLSTX algorithm.

**Table 4–59 Update Collateral Status in the Host: C1-UPCOLLSTX**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Status in the host   |
| <b>Detailed Description</b> | <p>This process updates the collateral status in the host. The value of status to be set will be passed as parameter to the process.</p> <p>If the update fails for any reason, system should create a To-do. Message for the To-do should be configured based on type of update which failed.</p> <p>To-do should be assigned to the To-do Role set as parameter to this process. If the parameter is left blank, To-do should be assigned to the default role.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost  |

**Table 4–59 (Cont.) Update Collateral Status in the Host: C1-UPCOLLSTX**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> To Do Role<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> Collateral Status<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in Settlement status of the Asset Repossession Process case. This process will update the collateral status in the host.   |

**Table 4–60 Update Collateral Status in the Host: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-UPCOLLSTZ  |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Role<br/><b>Value:</b></p> <p><b>Name:</b> To Do Type<br/><b>Value:</b> C1-TD-UC</p> <p><b>Name:</b> Collateral Status<br/><b>Value:</b> Sold</p> |

## 4.37 Validation Settlement: C1-VALSET

This section provides details of the Validation Settlement: C1-VALSET algorithm.

**Table 4–61 Validation Settlement: C1-VALSET**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validation Settlement  |
| <b>Detailed Description</b> | <p>This algorithm will perform following actions:</p> <p>Before completing the asset repossession case, the below validations should be done for the case:</p> <ul style="list-style-type: none"> <li>■ Collateral should have a settlement date</li> <li>■ Realization status for the collateral should be "Complete"</li> </ul> <p>Transition to completed status should fail if above validations fail.</p> |
| <b>Algorithm Entity</b>     | Case Type-Exit Status  |
| <b>Program Type</b>         | java   |

**Table 4–61 (Cont.) Validation Settlement: C1-VALSET**

|                        |  |
|------------------------|--|
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateralSettlementStatus                           |
| <b>Parameters</b>      | <b>Name:</b> Realization Status<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> NA   |
| <b>Detailed Design</b> | It is invoked in Settlement status of the Asset Repossession Process case. This process will update the collateral status in the host. |

**Table 4–62 Validation Settlement: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-UPCOLLSTZ  |
| <b>Parameters</b>     | <b>Name:</b> Realization Status<br><b>Value:</b> REALIZATION_COMPLETE |

## 4.38 Initiate LMI Process: C1-INITLMI

This section provides details of the Initiate LMI Process: C1-INITLMI algorithm.

**Table 4–63 Initiate LMI Process: C1-INITLMI**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Initiate LMI Process  |
| <b>Detailed Description</b> | Parameters to the algorithm must be as follows: <ul style="list-style-type: none"> <li>■ For Initiate LMI Options: <ol style="list-style-type: none"> <li>1) "Initiate LMI with highest insured amount" use HIGH_INSUR_AMT</li> <li>2) "Initiate LMI from a specific insurer first" use SPEC_INSURER</li> </ol> </li> <li>■ For No LMI Option: <ol style="list-style-type: none"> <li>1) "Mark primary account for strategy review" use PRIMARY</li> <li>2) "Mark all accounts for strategy review" use ALL</li> <li>3) "No Action" use NA</li> </ol> </li> </ul> |
| <b>Algorithm Entity</b>     | Case Type-Exit Status   |
| <b>Program Type</b>         | java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateLMIP  |

**Table 4–63 (Cont.) Initiate LMI Process: C1-INITLMI**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> Balance Threshold<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> LMI Case Type<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> Initiate LMI Options<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> LMI Insurer Code<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> No LMI Option<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> |
| <b>Detailed Design</b> | It is invoked in Settlement status of the Asset Repossession Process case. This process will validate realization status and settlement date for collateral.   |

**Table 4–64 Initiate LMI Process: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-INITLMI   |
| <b>Parameters</b>     | <p><b>Name:</b> Balance Threshold<br/><b>Value:</b> 1000</p> <p><b>Name:</b> LMI Case Type<br/><b>Value:</b> C1_LMI</p> <p><b>Name:</b> Initiate LMI Options<br/><b>Value:</b> HIGH_INSUR_AMT</p> <p><b>Name:</b> LMI Insurer Code<br/><b>Value:</b> QBE</p> <p><b>Name:</b> No LMI Option<br/><b>Value:</b> ALL</p> |



## 4.39 Close To do's Algorithm: C1-CLSTODO

This section provides details of the Close To do's Algorithm: C1-CLSTODO algorithm.

**Table 4–65 Close To do's Algorithm: C1-CLSTODO**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Close To do's algorithm  |
| <b>Detailed Description</b> | This process will close all To-Do's of specific To-do types associated with the case. Up to five To-Do types can be given to this algorithm to close.  |
| <b>Algorithm Entity</b>     | Case Type-Exit Status  |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Type1<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type2<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type3<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type4<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> <p><b>Name:</b> To Do Type5<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> NA</p> |
| <b>Detailed Design</b>      | It is invoked while exiting from Settlement status of the Asset Repossession Process case. This process will close all To-Do's associated with the case.   |

**Table 4–66 Close To do's Algorithm: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-CLSTODO   |
| <b>Parameters</b>     | <p><b>Name:</b> To Do Type1<br/><b>Value:</b> C1-TD-CL</p> <p><b>Name:</b> To Do Type2<br/><b>Value:</b> C1-TD-AC</p> <p><b>Name:</b> To Do Type3<br/><b>Value:</b> C1-TD-DN</p> <p><b>Name:</b> To Do Type4<br/><b>Value:</b> C1-DNA1</p> <p><b>Name:</b> To Do Type5<br/><b>Value:</b></p> |

## 4.40 Update Collateral Property: C1-RESETCOLL

This section provides details of the Update Collateral Property: C1-RESETCOLL algorithm.

**Table 4–67 Update Collateral Property: C1-RESETCOLL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Property   |
| <b>Detailed Description</b> | <p>This algorithm will perform following operations:</p> <ul style="list-style-type: none"> <li>▪ If the value of updateCollateralProperty soft parameter is "SET" and type of possession is "Warrant" then Fetch the collateral for which case is created and update the IS_LEGAL_SW= "Y" and populate the case_ID on this collateral.</li> <li>▪ If the value of updateCollateralProperty soft parameter is "RESET" then Fetch the collateral for which case is created and update the IS_LEGAL_SW= "N" and IS_REPO_SW= "N" nullify the case_ID on this collateral.</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralProperty   |
| <b>Parameters</b>           | <p><b>Name:</b> UpdateCollateralProperty</p> <p><b>Required (Yes/No):</b> Y</p> <p><b>Description:</b> NA</p>  |
| <b>Detailed Design</b>      | It is invoked in the Cancelled status of the Asset Repossession Process case. It will update the collateral Properties like IS_LEGAL_SW, IS_REPO_SW depending upon user inputs.  |

**Table 4–68 Update Collateral Property: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-RESETCOLL   |
| <b>Parameters</b>     | <b>Name:</b> UpdateCollateralProperty<br><b>Value:</b> RESET |

## 4.41 Update Collateral Status in the Host: C1-UPCOLLSTY

This section provides details of the Update Collateral Status in the Host: C1-UPCOLLSTY algorithm.

**Table 4–69 Update Collateral Status in the Host: C1-UPCOLLSTY**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collateral Status in the host   |
| <b>Detailed Description</b> | <p>This process will update the collateral status in the host. The value of status to be set will be passed as parameter to the process.</p> <p>If the update fails for any reason, system should create a To-do. Message for the To-do should be configured based on type of update which failed.</p> <p>To-do should be assigned to the To-do Role set as parameter to this process. If the parameter is left blank, To-do should be assigned to the default role.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Role<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> NA</p> <p><b>Name:</b> To Do Type<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p> <p><b>Name:</b> Collateral Status<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> NA</p>   |
| <b>Detailed Design</b>      | It is invoked in Withdrawn status of the Asset Repossession Process case. This process will update the collateral status in the host.  |

**Table 4–70 Update Collateral Status in the Host: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-UPCOLLSTY   |
| <b>Parameters</b>     | <b>Name:</b> To Do Role<br><b>Value:</b><br><br><b>Name:</b> To Do Type<br><b>Value:</b> C1-TD-UC<br><br><b>Name:</b> Collateral Status<br><b>Value:</b> With the Customer |

## 4.42 PTP Active Algorithm: C1-PTPACTIVE

This section provides details of the PTP Active Algorithm: C1-PTPACTIVE algorithm.

**Table 4–71 PTP Active Algorithm: C1-PTPACTIVE**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm to generate letter or SMS on Active Status                             |
| <b>Detailed Description</b> | This algorithm is used to generate letter or SMS when PTP moves to Active state. |
| <b>Algorithm Entity</b>     | PTP Active Algorithm   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm |

**Table 4–71 (Cont.) PTP Active Algorithm: C1-PTPACTIVE**

|                        |  |
|------------------------|--|
| <b>Parameters</b>      | <p><b>Name:</b> contactTypeForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for Letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for SMS</p> <p><b>Name:</b> contactClassForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for SMS</p> <p><b>Name:</b> contactMethodForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for SMS</p> |
| <b>Detailed Design</b> | This algorithm invokes <b>GenerateContactForPTP</b> service which creates the contact (generate Letter or SMS) when PTP moves to Active state.   |

**Table 4–72 PTP Active Algorithm: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-PTPACTIVE   |
| <b>Parameters</b>     | <p><b>Name:</b> contactTypeForLetter<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForLetter<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Value:</b> OTBL</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForSMS<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForSMS<br/><b>Value:</b> OTBS</p> |

## 4.43 PTP Active Algorithm: C1-PTPKEPT

This section provides details of the PTP Active Algorithm: C1-PTPKEPT algorithm.

**Table 4–73 PTP Active Algorithm: C1-PTPKEPT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm to generate letter or SMS on <b>Kept</b> status.                     |
| <b>Detailed Description</b> | This algorithm is used to generate letter or SMS when PTP moves to Kept state. |
| <b>Algorithm Entity</b>     | PTP Kept Algorithm   |
| <b>Program Type</b>         | java   |

**Table 4–73 (Cont.) PTP Active Algorithm: C1-PTPKEPT**

|                        |  |
|------------------------|--|
| <b>Program Name</b>    | com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPKeptForNgpAlgorithm   |
| <b>Parameters</b>      | <p><b>Name:</b> contactTypeForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for Letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for SMS</p> <p><b>Name:</b> contactClassForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for SMS</p> <p><b>Name:</b> contactMethodForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for SMS</p> |
| <b>Detailed Design</b> | This algorithm invokes <b>GenerateContactForPTP</b> service, which creates the contact (generate Letter or SMS) when PTP moves to <b>Kept</b> state.   |

**Table 4–74 PTP Active Algorithm: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-CURENTITY   |
| <b>Parameters</b>     | <p><b>Name:</b> contactTypeForLetter<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForLetter<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Value:</b> OTBL</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForSMS<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForSMS<br/><b>Value:</b> OTBS</p> |

## 4.44 PTP Active Algorithm: C1\_PTPBRKLS

This section provides details of the PTP Active Algorithm: C1\_PTPBRKLS algorithm.

**Table 4–75 PTP Active Algorithm: C1\_PTPBRKLS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm to generate letter or SMS on Broken Status                             |
| <b>Detailed Description</b> | This algorithm is used to generate letter or SMS when PTP moves to broken state. |
| <b>Algorithm Entity</b>     | PTP Broken Algorithm   |
| <b>Program Type</b>         | java   |



**Table 4–75 (Cont.) PTP Active Algorithm: C1\_PTPBRKLS**

|                        |  |
|------------------------|--|
| <b>Program Name</b>    | com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPBrokenForNgpAlgorithm   |
| <b>Parameters</b>      | <p><b>Name:</b> contactTypeForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for Letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Type for SMS</p> <p><b>Name:</b> contactClassForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Class for SMS</p> <p><b>Name:</b> contactMethodForSMS<br/> <b>Required (Yes/No):</b><br/> <b>Description:</b> Contact Method for SMS</p> |
| <b>Detailed Design</b> | This algorithm invokes GenerateContactForPTP service, which creates the contact (generate Letter or SMS) when PTP moves to Broken state.   |

**Table 4–76 PTP Active Algorithm: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1_PTPBRKLS  |
| <b>Parameters</b>     | <p><b>Name:</b> contactTypeForLetter<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForLetter<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Value:</b> OTBL</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Value:</b> OVERDUE</p> <p><b>Name:</b> contactClassForSMS<br/><b>Value:</b> CCC</p> <p><b>Name:</b> contactMethodForSMS<br/><b>Value:</b> OTBS</p> |

If you want to generate letter, the following parameters are mandatory:

- contactTypeForLetter
- contactClassForLetter
- contactMethodForLetter

If you want to generate SMS, following parameters are mandatory:

- contactTypeForSMS
- contactClassForSMS
- contactMethodForSMS

If you want to generate both Letter and SMS, following parameters are mandatory:

- contactTypeForLetter
- contactClassForLetter
- contactMethodForLetter
- contactTypeForSMS
- contactClassForSMS
- contactMethodForSMS

## 4.45 Rule facts populating algorithm: C1-BRLSR

This section provides details of the rule facts populating Algorithm: C1\_BRLSR algorithm.

**Table 4-77 Rule Facts Populating Algorithm: C1-BRLSR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This algorithm is used to populate the facts required for Rule engine.   |
| <b>Detailed Description</b> | This algorithm populates rule facts for Rule/Ruleset from defined Business Object (BO).  |
| <b>Algorithm Entity</b>     | BO Rule Search - Rule Parameter Search   |
| <b>Program Type</b>         | java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.RuleFactsPopulation  |
| <b>Parameters</b>           | <p><b>Name:</b> Input Key1<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Primary Key name of defined BO.</p> <p><b>Name:</b> Input Key2<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Primary Key name of defined BO.</p> <p><b>Name:</b> Input Key3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Primary Key name of defined BO.</p> <p><b>Name:</b> Input Key4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Primary Key name of defined BO.</p> <p><b>Name:</b> Input Key5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Primary Key name of defined BO.</p> |

**Table 4-77 (Cont.) Rule Facts Populating Algorithm: C1-BRLSR**

|                        |   |
|------------------------|---|
| Parameters             | <p><b>Name:</b> Input B O Name1<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> BO name to fetch fact values. If BOName1 is defined then its primary key name must be defined in Input Key 1. Similarly configure other BO names.</p> <p><b>Name:</b> Input B O Name2<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> BO name to fetch fact values. If BOName1 is defined then its primary key name must be defined in Input Key 1. Similarly configure other BO names.</p> <p><b>Name:</b> Input B O Name3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> BO name to fetch fact values. If BOName1 is defined then its primary key name must be defined in Input Key 1. Similarly configure other BO names.</p> <p><b>Name:</b> Input B O Name4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> BO name to fetch fact values. If BOName1 is defined then its primary key name must be defined in Input Key 1. Similarly configure other BO names.</p> <p><b>Name:</b> Input B O Name5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> BO name to fetch fact values. If BOName1 is defined then its primary key name must be defined in Input Key 1. Similarly configure other BO names.</p> |
| Parameters             | <p><b>Name:</b> Bo Fields<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Comma separated BO fields of defined BO names.</p> <p><b>Name:</b> Rule Fact Codes<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Comma separated fact codes for rule to be executed. BO Fields and Rule Fact codes should be defined in the same order.</p> <p><b>Name:</b> Pre Populated Rule Facts Algorithm Code<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Algorithm code of algorithm holding pre populated facts. Rule facts which cannot be retrieved from BO fields can be pre populated in algorithm. These facts will be appended to input facts for rule under execution. Algorithm type must be defined on algorithm spot 'Rule Execution - Pre Populated Rule Facts' (For more information check sample implementation 'C1-PPSF').</p>   |
| <b>Detailed Design</b> | <p>This algorithm is used to populate rule facts from Business object (BO).</p> <p>Business object fields are fetched using combination of BO name and its respective primary key. Further these values are mapped to rule fact code.</p> <p>Also, pre-populated facts are appended to these values, if provided from external algorithm. These populated facts will act as input to defined rule through soft parameter.</p>   |

**Sample Algorithm**

**Table 4-78 Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-BRLSR   |
| <b>Parameters</b>     | <p><b>Name:</b> Input Key1<br/><b>Value:</b> accountId</p> <p><b>Name:</b> Input Key2<br/><b>Value:</b></p> <p><b>Name:</b> Input Key3<br/><b>Value:</b></p> <p><b>Name:</b> Input Key4<br/><b>Value:</b></p> <p><b>Name:</b> Input Key5<br/><b>Value:</b></p>   |
|                       | <p><b>Name:</b> Input B O Name1<br/><b>Value:</b> C1-ACCT-EXTN</p> <p><b>Name:</b> Input B O Name2<br/><b>Value:</b></p> <p><b>Name:</b> Input B O Name3<br/><b>Value:</b></p> <p><b>Name:</b> Input B O Name4<br/><b>Value:</b></p> <p><b>Name:</b> Input B O Name5<br/><b>Value:</b></p> <p><b>Name:</b> Bo Fields<br/><b>Value:</b> productClassCode, overdueAmount</p> <p><b>Name:</b> Rule Fact Codes<br/><b>Value:</b> ProductClass, OverdueAmount</p> <p><b>Name:</b> Pre Populated Rule Facts Algorithm Code<br/><b>Value:</b></p> |



## Feeder Services

Feeder tables in Collections act as an additional layer to validate incoming data pulled from the host. Since ORMB has its own architecture and framework, incoming data from any host is validated as per ORMB objects standard.

**Table 5–1 Feeder Services**

| Service Name                                | Method Name  | Description  | Mandatory Fields   |
|---|--|--|--|
| AccountFeederApplicationService             | AccountFeederResponse update(SessionContext sessionContext,AccountFeederWrapperDTO accountFeederWrapperDTO) throws FatalException                                      | This service adds or updates account related fields in the feeder table. It handles add, update and delete operations.   | hostAcctNumber, srcHostId  |
| AccountHardshipDtlsFeederApplicationService | AccountHardshipDtlsFeederResponse update(SessionContext sessionContext,AccountFeederHardshipDtlsWrapperDTO accountFeederHardshipDtlsWrapperDTO) throws FatalException; | This service adds or updates accounts hardship related fields in the feeder table. It handles add, update and delete operations.   | hostAcctNumber, srcHostId, reliefEffDt, reliefExpDt, reliefType, hrshipAppId |
| AccountArrearFeederApplicationService       | AccountArrearFeederResponse update(SessionContext sessionContext,AccountArrearFeederWrapperDTO accountArrearFeederWrapperDTO) throws FatalException;                   | This service adds or updates account arrears related fields in the feeder table. It handles add, update and delete operations. In case of delete, the service also deletes the record from main table. | hostAcctNumber, srcHostId, referenceVal                                      |
| AccountWarningIndFeederApplicationService   | AccountWarningIndFeederResponse update(SessionContext sessionContext,AccountWarningIndFeederWrapperDTO accountWarningIndFeederWrapperDTO) throws FatalException;       | This service adds or updates account warning indicator related fields in the feeder table. It handles add, update and delete operations.   | hostAcctNumber, srcHostId  |
| AcctPerFeederApplicationService             | AcctPerFeederResponse update(SessionContext sessionContext,AcctPerFeederWrapperDTO acctPerFeederWrapperDTO) throws FatalException;                                     | This service adds or updates account person relationship fields in the feeder table. It handles add, update and delete operations.   | hostAcctNumber, srcHostId, hostCustomerNbr                                   |

**Table 5–1 (Cont.) Feeder Services**

| <b>Service Name</b>                   | <b>Method Name</b>  | <b>Description</b>  | <b>Mandatory Fields</b>                                       |
|---------------------------------------|---|---|---|
| FeederPersonApplicationService        | FeederPersonResponse update(SessionContext sessionContext,AccountFeeder WrapperDTO accountFeederWrapperDTO) throws FatalException                     | This service adds or updates party related fields in the feeder table. It handles add, update and delete operations.  | srcHostId, hostCustomerNbr                                    |
| FeederPerAddrApplicationService       | FeederPerAddrResponse update(SessionContext sessionContext,FeederPerAddr WrapperDTO) throws FatalException  | This service adds or updates party address related fields in the feeder table. It handles add, update and delete operations.                                    | srcHostId, hostCustomerNbr, fdrAddrSeqId, addrTypeCd          |
| FeederPerEmpProfileApplicationService | FeederPerEmpProfileResponse update(SessionContext sessionContext,FeederPerEmpProfileWrapperDTO feederPerEmpProfileWrapperDTO) throws FatalException   | This service adds or updates party employment details fields in the feeder table. It handles add, update and delete operations.                                 | srcHostId, hostCustomerNbr, determinantValue, fdrEmpSeqId     |
| FeederContactPrefApplicationService   | FeederContactPrefResponse update(SessionContext p_ SessionContext, FeederContactPrefWrapperDTO p_ FeederContactPrefWrapperDTO ) throws FatalException | This service adds or updates party contact preferences fields in the feeder table. It handles add, update and delete operations.                                | srcHostId, hostCustomerNbr, contactPrefType, contactPointType |
| FeederPerIdApplicationService         | FeederPerIdResponse update(SessionContext p_ SessionContext, FeederPerIdWrapperDTO p_ FeederPerIdWrapperDTO) throws FatalException                    | This service adds or updates party ID type related fields, such as driving license and so on in the feeder table. It handles add, update and delete operations. | srcHostId, hostCustomerNbr, idType                            |