

# **Oracle® Banking Enterprise Default Management**

Interface Specification Guide

Release 2.10.0.0.0

**F29509-01**

April 2020

Oracle Banking Enterprise Default Management Interface Specification Guide, Release 2.10.0.0.0

F29509-01

Copyright © 2017, 2020, Oracle and/or its affiliates.

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

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

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

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

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

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. 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.

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

# Contents

---

|  |           |
|--|-----------|
| <b>Preface</b> .....                           | <b>27</b> |
| Audience .....                                 | 27        |
| Documentation Accessibility .....              | 27        |
| Organization of the Guide .....                | 27        |
| Related Documents .....                        | 28        |
| Conventions .....                              | 28        |
| <b>1 About this Guide</b> .....                | <b>30</b> |
| <b>2 Introduction</b> .....                    | <b>32</b> |
| <b>3 System Overview</b> .....                 | <b>34</b> |
| <b>4 Staging Area</b> .....                    | <b>36</b> |
| 4.1 Feeder Tables .....                        | 36        |
| 4.1.1 Account Data .....                       | 36        |
| 4.1.1.1 Account Details .....                  | 36        |
| 4.1.1.2 Account Arrears Details .....          | 47        |
| 4.1.1.3 Account Hardship Details .....         | 49        |
| 4.1.1.4 Account Repayment Schedule .....       | 50        |
| 4.1.1.5 Account Warning Indicator .....        | 51        |
| 4.1.1.6 Minimum Amount Due on Bill (MAD) ..... | 52        |
| 4.1.1.7 Payment Tracker Details .....          | 53        |
| 4.1.2 Party Data .....                         | 54        |
| 4.1.2.1 Party Account Relationship .....       | 54        |
| 4.1.2.2 Party Details .....                    | 56        |
| 4.1.2.3 Party Address Details .....            | 60        |
| 4.1.2.4 Party Employment Details .....         | 62        |

---

|  |    |
|--|----|
| 4.1.2.5 Party Identification Details .....       | 64 |
| 4.1.2.6 Party Name Details .....                 | 66 |
| 4.1.2.7 Party Contact Preference Details .....   | 67 |
| 4.1.2.8 Party Warning Indicators .....           | 70 |
| 4.1.2.9 Service Member History Details .....     | 71 |
| 4.1.3 Collateral Data .....                      | 73 |
| 4.1.3.1 Collateral Details .....                 | 73 |
| 4.1.3.2 Collateral Charge Details .....          | 75 |
| 4.1.3.3 Collateral Entity Mapping .....          | 76 |
| 4.1.3.4 Collateral Guarantor Mapping .....       | 77 |
| 4.1.3.5 Collateral Owner Mapping .....           | 78 |
| 4.1.3.6 Collateral Vehicle Mapping .....         | 79 |
| 4.1.4 Insurance Data .....                       | 80 |
| 4.1.4.1 Insurance Details .....                  | 80 |
| 4.1.5 Payment Data .....                         | 82 |
| 4.1.5.1 Online Payment Records .....             | 82 |
| 4.1.6 IRS Reporting .....                        | 83 |
| 4.2 Interfacing Tables .....                     | 84 |
| 4.2.1 Agency or Vendor Upload .....              | 84 |
| 4.2.1.1 Upload Followup Table Details .....      | 84 |
| 4.2.1.2 Upload Result Table Details .....        | 86 |
| 4.2.1.3 Upload PTP Table Details .....           | 87 |
| 4.2.1.4 Upload PTP Schedule Table Details .....  | 88 |
| 4.2.1.5 Upload Settlement Table Details .....    | 89 |
| 4.2.2 Dialer Results Upload .....                | 89 |
| 4.2.2.1 Upload Dialer Result Table Details ..... | 90 |

---

|   |            |
|---|------------|
| 4.2.3 Account Dialer Extract .....                      | 93         |
| 4.2.3.1 Dialer Extract Account Table Details .....      | 93         |
| 4.2.3.2 Dialer Extract Contact Table Details .....      | 95         |
| 4.3 OBP Views .....                                     | 97         |
| 4.3.1 Main Account Views .....                          | 97         |
| 4.3.2 Account Updateable Views .....                    | 97         |
| 4.3.3 Hardship Views .....                              | 97         |
| 4.3.4 Party Views .....                                 | 98         |
| 4.3.5 LCM / Collateral Views .....                      | 98         |
| 4.4 OBP Collections Views .....                         | 98         |
| <b>5 Algorithms .....</b>                               | <b>100</b> |
| 5.1 Stop Contract: C1-CURRENTITY .....                  | 100        |
| 5.2 Cure Account: C1-FINCOLL .....                      | 100        |
| 5.3 Queue Allocation: C1-ALLOCQUEUE .....               | 101        |
| 5.4 Update Customer Switch: C1-CUSTSW .....             | 102        |
| 5.5 Update Legal/Repo Switch: C1-LEREPOCT .....         | 103        |
| 5.6 User Allocation - Round Robin: C1-USRALCRR .....    | 104        |
| 5.7 User Allocation - % Based: C1-USRALCPR .....        | 105        |
| 5.8 Vendor Allocation - Round Robin: C1-VENALCRR .....  | 107        |
| 5.9 Vendor Allocation - % Based: C1-VENALCPR .....      | 108        |
| 5.10 Bulk Contact Creation: C1-BLKCNTCRE .....          | 110        |
| 5.11 Cross Strategy Action Matrix: C1-CSAM .....        | 110        |
| 5.12 Last Payment for Account: C1-PAYDTAMTU .....       | 111        |
| 5.13 Association Review Check: C1-ASORVCHK .....        | 112        |
| 5.14 Validate Expired Default Notice: C1-DEFNOEXP ..... | 113        |
| 5.15 Associate Related Entity: C1-ASSOENTY .....        | 114        |

---

|  |     |
|--|-----|
| 5.16 Validate Legal Case Exists: C1-CHKLGL .....                         | 115 |
| 5.17 Assign New LSP: C1-ASGNLSP .....                                    | 116 |
| 5.18 Check Approval Requirement: C1-APPRCHK .....                        | 117 |
| 5.19 Resume Status from Previous LSP: C1-RESSTATUS .....                 | 119 |
| 5.20 Check Submission Date: CI_CHKSUBDT1 .....                           | 119 |
| 5.21 Update LSP (CLOS): C1-LSPSTATUS .....                               | 120 |
| 5.22 Update LSP (CANCEL): C1-LSPSTACAN .....                             | 121 |
| 5.23 Collateral Verification: C1-VRFYCOLS .....                          | 121 |
| 5.24 Account Association for Asset Repossession Case: C1-ARSACCTS .....  | 122 |
| 5.25 Customer Association for Asset Repossession Case: C1-ARSCUSTS ..... | 123 |
| 5.26 Update Collateral Property: C1-UPCOLPROP .....                      | 124 |
| 5.27 Close To do's Algorithm: C1-CLSTODO .....                           | 124 |
| 5.28 Validations for Mandatory Characteristics: C1-CHARVALS .....        | 125 |
| 5.29 Validations for Mandatory Characteristics: C1-CHARVALA .....        | 127 |
| 5.30 Update Collateral Status in the Host: C1-UPCOLLSTX .....            | 129 |
| 5.31 Initiate Collateral Valuation: C1-COLLVALX .....                    | 130 |
| 5.32 Validation Settlement: C1-VALSET .....                              | 131 |
| 5.33 Initiate LMI Process: C1-INITLMI .....                              | 131 |
| 5.34 PTP Active Algorithm: C1-PTPACTIVE .....                            | 133 |
| 5.35 PTP Kept Algorithm: C1-PTPKEPT .....                                | 134 |
| 5.36 PTP Broken Algorithm: C1-BRKPTPNGP .....                            | 136 |
| 5.37 Rule facts populating algorithm: C1-BRLSR .....                     | 137 |
| 5.38 Borrower Centric Case Lifecycle .....                               | 140 |
| 5.39 Update Collection Address on Borrower Panel .....                   | 141 |
| 5.40 Update Collection Contact Point .....                               | 141 |
| 5.41 Bankruptcy Process .....  | 142 |

---

|   |     |
|---|-----|
| 5.42 Task - Automatic Allocation of tasks to Vendors .....              | 157 |
| 5.43 Hardship - Associate Accounts of Main Customer .....               | 158 |
| 5.44 Early Collection .....   | 158 |
| 5.45 Asset Repossession .....   | 177 |
| 5.46 Miscellaneous .....  | 192 |
| 5.47 Derived Field .....  | 195 |
| 5.48 Task .....   | 196 |
| 5.49 Event Manager .....  | 198 |
| 5.50 Legal Vendor Allocation C1-LGLVNDRAL .....                         | 198 |
| 5.51 Extend Expiry Date C1-EXT-EXPDT .....                              | 199 |
| 5.52 Capture Case Status Update Date/Time C1-CASE-STAT .....            | 200 |
| 5.53 Create To Do C1-TO-DO .....  | 200 |
| 5.54 Check customer eligibility C1-CHKCUST .....                        | 201 |
| 5.55 Capture Hardship Approval Date C1-HARAP-DT .....                   | 203 |
| 5.56 Algorithm that will interface with Rule Engine. C1-RULEADAPT ..... | 203 |
| 5.57 Perform Validation for Collateral C1-VERFYCOL .....                | 203 |
| 5.58 Check target account eligibility C1-CHKTRGT .....                  | 204 |
| 5.59 Approval check for set-off transaction C1-ROSOAPPR .....           | 205 |
| 5.60 Get target accounts C1-GETTRGT .....                               | 206 |
| 5.61 Update status of relief to Expired in Hardship C1-UPDHDSTAT .....  | 208 |
| 5.62 To Do Completion for case C1-TO-DO-END .....                       | 208 |
| 5.63 Update Marketing Consent flag C1-MKT-FLG .....                     | 209 |
| 5.64 Check Default Notice for Voluntary possession C1-CHKDFLT .....     | 209 |
| 5.65 Check Submission Date C1-CHKSUBDT2 .....                           | 210 |
| 5.66 Update Financial Hardship flag C1-FNHRD-FLG .....                  | 210 |
| 5.67 Result Type Case Transition Algorithm C1-RTCT .....                | 211 |

---

|      |   |     |
|------|---|-----|
| 5.68 | Algorithm to see if case is running before closing C1-CHKCASE                                 | 211 |
| 5.69 | Check Deceased status for the customer C1-CHKDCD  | 212 |
| 5.70 | Associated accounts with deceased customer case C1-DCDACCTS                                   | 212 |
| 5.71 | Execute Fund Transfer C1-FUNDTRFR   | 213 |
| 5.72 | Algorithm to save previous state's status code C1-SAVPRESTA                                   | 214 |
| 5.73 | Attach case type from feature config attach to BO C1-ATCHCS                                   | 214 |
| 5.74 | Update Collection Warning Indicator C1-UPD-WRIND  | 215 |
| 5.75 | Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSO | 216 |
| 5.76 | Assign Applicable Relief Type C1-RELIF-TYP  | 216 |
| 5.77 | Create Customer Contact for Resultype Algo C1-CREATCC   | 217 |
| 5.78 | Calculate an expiry date when entering case status C1-CSEXPDT                                 | 217 |
| 5.79 | Create Customer Contact C1-CUST-CONT  | 218 |
| 5.80 | Transition to Next Status x days before expiry C1-NXT-BX-DY                                   | 219 |
| 5.81 | Validate Hardship Expiry Date. C1-VAL-FHEXP   | 219 |
| 5.82 | Update Account in collections flag C1-ACTINCOL  | 220 |
| 5.83 | Associate related entities with the case C1-ARSENTITY   | 220 |
| 5.84 | Revalidate target account C1-REVALTRGT  | 221 |
| 5.85 | Initiate LMI C1-INITLMIS  | 223 |
| 5.86 | NGP Collection case creation algorithm C1-COLLCASE  | 224 |
| 5.87 | Stop Contract Algorithm C1-CONTSTOP   | 225 |
| 5.88 | Check for existing Hardship C1-CHKHRDSHP  | 225 |
| 5.89 | Algorithm for contact processing C1-CNTCT   | 226 |
| 5.90 | Check application expiry date C1-CHKEXP   | 226 |
| 5.91 | New Customer Contact Creation Algorithm C1-CCCREATE   | 226 |
| 5.92 | Removes a case characteristic on case status exit C1-REMCSCH                                  | 227 |
| 5.93 | Transition case on a date on a case characteristic C1-TRANSDT                                 | 227 |



---

|   |     |
|---|-----|
| 5.94 Set Account Nxt Credit Review Date to current date C1-NXTRVWDT ..... | 228 |
| 5.95 Mark accounts for strategy review C1-REVIW-ACT .....                 | 228 |
| 5.96 Wait Time Out (in days) C1-WAIT-DAYS .....                           | 229 |
| 5.97 Validate Hardship Application inputs C1-V-FH-APP .....               | 229 |
| 5.98 Check for Operational Relief Types C1-OP-RT .....                    | 230 |
| 5.99 Auto-Approval Check C1-FH-AUTOAP .....                               | 231 |
| 5.100 Apply Hardship Relief Types for accounts in Host C1-FH-EVAL .....   | 231 |
| 5.101 Update Party Warning Indicator C1-UPD-PRTWI .....                   | 232 |
| 5.102 Transition to Next status when all reliefs are app C1-RAPP .....    | 233 |
| 5.103 Collection - Entity Activity Population. C1-ENTACTPOP .....         | 233 |
| 5.104 Cancel Hardship Application C1-CXLFH .....                          | 234 |
| 5.105 Perform Payment Transfer for ROSO C1-ROSOPMTXR .....                | 234 |
| 5.106 Validate ROSO Target Account inputs C1-RS-VALIN .....               | 235 |
| 5.107 Create RMB Entities from Host Data C1-VCREATE .....                 | 235 |
| 5.108 Populate Activity Table For Notes Creation C1-NTACTIVITY .....      | 236 |
| 5.109 Suspend Activity for Account Pre Processing C1-SPATACPRE .....      | 236 |
| 5.110 Sample TAM Algorithm Type C1-TAMALG .....                           | 237 |
| 5.111 Cancel Approval Request C1-CANAPPR .....                            | 237 |
| 5.112 Update Customer in collections flag C1-CUSINCOL .....               | 238 |
| 5.113 Set Display Date C1-SETDSPDT .....                                  | 239 |
| 5.114 Transition to Default next status after N Days C1-TRNDFLT .....     | 239 |
| 5.115 Check current cases on account for exclusion C1-EXCLCASE .....      | 240 |
| 5.116 Update Collateral Status in the host C1-UPCOLLST .....              | 241 |
| 5.117 Initiate collateral valuation C1-COLLVAL .....                      | 242 |
| 5.118 Mandatory characteristics check for Asset Repo C1-CHARVAL .....     | 242 |
| 5.119 Update Collateral Status in the Host C1-UPCOLLSTS .....             | 243 |

---

|  |     |
|--|-----|
| 5.120 Set exclusion date for delinquent account. C1-EXCLROSO .....                             | 244 |
| 5.121 Cancel Set-off C1-CANROSO .....  | 245 |
| 5.122 Complete Set-off C1-COMPROSO .....   | 245 |
| 5.123 Reverse Set-off C1-REVROSO .....   | 246 |
| 5.124 Algorithm type for update case id for Insurance C1-UPCASFINS .....                       | 246 |
| 5.125 Case Creation on enter processing C1-CCOENTER .....                                      | 246 |
| 5.126 Collection - Case Creation On Exit of Status C1-CCOE .....                               | 247 |
| 5.127 Action category Validation algorithm C1-ACTCAT .....                                     | 247 |
| 5.128 Action Type Algorithm Type C1-ACTTYP .....   | 248 |
| 5.129 Case Type Status Mapping Algorithm Type C1-CASETYMP .....                                | 248 |
| 5.130 Collection - Close Processing Algorithm C1-CCALG .....                                   | 248 |
| 5.131 Algorithm type for case list update C1-CASELIST .....                                    | 249 |
| 5.132 Copy Characteristics Algorithm C1-COPYCHAR .....   | 250 |
| 5.133 Call Advice - Red/Green logic calculation C1-CALADVICE .....                             | 250 |
| 5.134 Task Case Mapping Validation Algorithm C1-TCVAL .....                                    | 251 |
| 5.135 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status<br>C1-DISMON ..... | 252 |
| 5.136 SLA Parameters validation algorithm C1-SLAPARAM .....                                    | 253 |
| 5.137 Case Group add validation algorithm C1-CGVAL .....                                       | 254 |
| 5.138 Collection - Get Strategy Algorithm C1-COLGS .....                                       | 254 |
| 5.139 Create/Move Collection Strategy Cases for Account C1-COLOMR .....                        | 254 |
| 5.140 Collection - Case SA Update for Manual Creation C1-CSAUPD .....                          | 256 |
| 5.141 Promise to Pay - Additional Grace Days Sample Algo C1-PPADDLGRD .....                    | 256 |
| 5.142 Promise to Pay Threshold Percentage C1-PPTHRESH .....                                    | 256 |
| 5.143 Result type Post Processing Case Transition Algo C1-RTPCC .....                          | 257 |
| 5.144 Inbound Customer algorithm C1-IN-CUST .....  | 258 |
| 5.145 Result Type Pre-processing Algorithm Type C1-RSTPRE .....                                | 259 |

---

|   |            |
|---|------------|
| 5.146 Result Type Post-processing Algorithm Type C1-RSTPOST .....   | 259        |
| 5.147 Characteristic Type :Validate Date Field (Custom) C1-ADHDATE .....  | 259        |
| 5.148 Algorithm Type for Dialer Results Upload C1-DLRRSUPLD .....   | 260        |
| 5.149 Algorithm for Hardship case creation activity C1-CRTHDSP .....  | 260        |
| 5.150 This Algorithm is used to abort Approval work item C1-ABORTAPP .....  | 261        |
| 5.151 Cancel Process Approval Request:Financial Hardship C1-CANFHAPPR   | 262        |
| 5.152 Hardship Characteristic Association C1-FHCHARASC .....  | 262        |
| 5.153 Pre-Populated system facts to be used for Rule C1-PPSF .....  | 263        |
| 5.154 Assign Batch level TODOs(task) to a queue C1-ASGNTASK .....   | 263        |
| 5.155 Validate Date Field :Custom C1-ADHV-DTD .....   | 263        |
| 5.156 Characteristic Date field Validation C1-CHARDTVVAL .....  | 265        |
| 5.157 Retry Case in Error C1-RCASEE .....   | 266        |
| 5.158 Allocate Queue for Customer Level Case C1-ALLOCQUE .....  | 267        |
| 5.159 Person Address – Collection C1-PERADDRRC .....  | 267        |
| 5.160 Person Contact Point Update - Post Processing C1-PERCONTPP .....  | 268        |
| 5.161 Update Self Serve Flag Algorithm C1-SELFSERVE .....   | 268        |
| 5.162 Create Task for Self Serve Request for Assistance transaction C1-<br>FLWRTSK .....                          | 268        |
| 5.163 Transition to Under Settlement Offer Status .....   | 269        |
| 5.164 Create PTP Validation Algorithm .....   | 269        |
| 5.165 Under Settlement Offer Status Monitor Algorithm for Settlement Expiry/Of-<br>fer Acceptance/Rejection ..... | 270        |
| 5.166 Algorithm to identify Partial Charge off Accounts .....   | 271        |
| 5.167 Monitor Settlement Offer Payments .....   | 272        |
| <b>6 Localized Algorithms .....</b>   | <b>274</b> |
| 6.1 Localized Algorithms .....  | 274        |
| <b>7 Feeder Services .....</b>  | <b>282</b> |

---

|  |            |
|--|------------|
| <b>8 Dialer Webservice Integration</b> ..... | <b>290</b> |
| 8.1 Generic Data Type .....                  | 290        |
| 8.2 Summary .....                            | 291        |
| 8.3 Interface .....                          | 291        |
| 8.4 Service Management .....                 | 292        |
| 8.5 Request Message Details .....            | 292        |
| 8.6 Header Record .....                      | 293        |
| 8.7 Detail Record .....                      | 293        |
| 8.8 Translation Rules .....                  | 293        |
| 8.9 Response Message Details .....           | 294        |
| 8.10 Customer Information .....              | 294        |
| 8.11 Constraints .....                       | 294        |

# List of Tables

---

|   |    |
|---|----|
| Table 4–1 Account Details .....                   | 36 |
| Table 4–2 Account Arrears Details .....           | 47 |
| Table 4–3 Account Hardship Details .....          | 49 |
| Table 4–4 Account Repayment Schedule .....        | 50 |
| Table 4–5 Account Warning Indicator .....         | 51 |
| Table 4–6 Minimum Amount Due Bill .....           | 52 |
| Table 4–7 Payment Tracker Details .....           | 53 |
| Table 4–8 Account Party Relationship .....        | 54 |
| Table 4–9 Party Details .....                     | 57 |
| Table 4–10 Party Address Details .....            | 60 |
| Table 4–11 Party Employment Details .....         | 62 |
| Table 4–12 Party Identification Details .....     | 64 |
| Table 4–13 Party Name Details .....               | 66 |
| Table 4–14 Party Contact Preference Details ..... | 68 |
| Table 4–15 Party Warning Indicators .....         | 71 |
| Table 4–16 Service Member History Details .....   | 72 |
| Table 4–17 Collateral Details .....               | 73 |
| Table 4–18 Collateral Charges Details .....       | 75 |
| Table 4–19 Collateral Entity Mapping .....        | 76 |
| Table 4–20 Collateral Guarantor Mapping .....     | 77 |
| Table 4–21 Collateral Owner Mapping .....         | 78 |
| Table 4–22 Collateral Vehicle Mapping .....       | 79 |
| Table 4–23 Insurance Details .....                | 80 |
| Table 4–24 Online Payment .....                   | 82 |

---

|   |     |
|---|-----|
| Table 4–25 IRS Report Data .....                            | 84  |
| Table 4–26 Upload Followup Table .....                      | 84  |
| Table 4–27 Upload Result Table .....                        | 86  |
| Table 4–28 PTP Upload data .....                            | 87  |
| Table 4–29 PTP Schedule Upload data .....                   | 88  |
| Table 4–30 Settlement Table data .....                      | 89  |
| Table 4–31 Dialer Result Upload data .....                  | 90  |
| Table 4–32 Account Dialer Extract data. ....                | 94  |
| Table 4–33 Dialer Extract Contact data. ....                | 96  |
| Table 5–1 Stop Contract: C1-CURENTITY .....                 | 100 |
| Table 5–2 Cure Account: C1-FINCOLL .....                    | 100 |
| Table 5–3 Cure Account: Sample Algorithm .....              | 101 |
| Table 5–4 Queue Allocation: C1-ALLOCQUEUE .....             | 101 |
| Table 5–5 Queue Allocation: Sample Algorithm .....          | 102 |
| Table 5–6 Update Customer Switch: C1-CUSTSW .....           | 102 |
| Table 5–7 Update Customer Switch: Sample Algorithm .....    | 102 |
| Table 5–8 Update Legal/Repo Switch: C1-LEREPOCT .....       | 103 |
| Table 5–9 Update Legal/Repo Switch: Sample Algorithm .....  | 103 |
| Table 5–10 User Allocation - Round Robin: C1-USRALCRR ..... | 104 |
| Table 5–11 User Allocation - % Based: C1-USRALCPR .....     | 105 |
| Table 5–12 User allocation percentage of Queue Q1 .....     | 105 |
| Table 5–13 Calculations for allocating cases .....          | 105 |
| Table 5–14 Final Distribution .....                         | 106 |
| Table 5–15 User allocation percentage of Queue Q2 .....     | 106 |
| Table 5–16 Calculations for allocating cases .....          | 106 |
| Table 5–17 Final distribution in cases .....                | 107 |

---

|  |     |
|--|-----|
| Table 5–18 Vendor Allocation - Round Robin: C1-VENALCRR .....      | 107 |
| Table 5–19 Vendor Allocation - % Based: C1-VENALCRR .....          | 109 |
| Table 5–20 Bulk Contact Creation: C1-BLKCNTCRE .....               | 110 |
| Table 5–21 Cross Strategy Action Matrix: C1-CSAM .....             | 111 |
| Table 5–22 Cross Strategy Action Matrix: Sample Algorithm .....    | 111 |
| Table 5–23 Last Payment for Account: C1-PAYDTAMTU .....            | 112 |
| Table 5–24 Association Review Check: C1-ASORVCHK .....             | 112 |
| Table 5–25 Association Review Check: Sample Algorithm .....        | 113 |
| Table 5–26 Validate Expired Default Notice: C1-DEFNOEXP .....      | 113 |
| Table 5–27 Validate Expired Default Notice: Sample Algorithm ..... | 114 |
| Table 5–28 Associate Related Entity: C1-ASSOENTY .....             | 114 |
| Table 5–29 Associate Related Entity: Sample Algorithm .....        | 115 |
| Table 5–30 Validate Legal Case Exists: C1-CHKLGL .....             | 115 |
| Table 5–31 Validate Legal Case Exists: Sample Algorithm .....      | 115 |
| Table 5–32 Assign New LSP: C1-ASGNLSP .....                        | 116 |
| Table 5–33 Assign New LSP: Sample Algorithm .....                  | 117 |
| Table 5–34 Check Approval Requirement: C1-APPRCHK .....            | 117 |
| Table 5–35 Check Approval Requirement: Sample Algorithm .....      | 118 |
| Table 5–36 Resume Status from Previous LSP: C1-RESSTATUS .....     | 119 |
| Table 5–37 Check Submission Date: CI_CHKSUBDT1 .....               | 119 |
| Table 5–38 Check Submission Date: Sample Algorithm .....           | 120 |
| Table 5–39 Update LSP (CLOS): C1-LSPSTATUS .....                   | 120 |
| Table 5–40 Update LSP (CLOS): Sample Algorithm .....               | 121 |
| Table 5–41 Update LSP (CANCEL): C1-LSPSTACAN .....                 | 121 |
| Table 5–42 Update LSP (CANCEL): Sample Algorithm .....             | 121 |
| Table 5–43 Collateral Verification: C1-VRFYCOLS .....              | 122 |

---

|  |     |
|--|-----|
| Table 5–44 Collateral Verification: Sample Algorithm .....                         | 122 |
| Table 5–45 Account Association for Asset Repossession Case: C1-ARSACCTS ..         | 122 |
| Table 5–46 Customer Association for Asset Repossession Case: C1-<br>ARSCUSTS ..... | 123 |
| Table 5–47 Update Collateral Property: C1-UPCOLPROP .....                          | 124 |
| Table 5–48 Close To do's Algorithm:C1-CLSTODO .....                                | 124 |
| Table 5–49 Close To do's Algorithm: Sample Algorithm .....                         | 125 |
| Table 5–50 Validations for Mandatory Characteristics:C1-CHARVALS .....             | 126 |
| Table 5–51 Validations for Mandatory Characteristics: Sample Algorithm .....       | 127 |
| Table 5–52 Validations for Mandatory Characteristics: C1-CHARVALA .....            | 127 |
| Table 5–53 Validations for Mandatory Characteristics: Sample Algorithm .....       | 128 |
| Table 5–54 Update Collateral Status in the Host: C1-UPCOLLSTX .....                | 129 |
| Table 5–55 Update Collateral Status in the Host: Sample Algorithm .....            | 129 |
| Table 5–56 Initiate Collateral Valuation: C1-COLLVALX .....                        | 130 |
| Table 5–57 Initiate Collateral Valuation: Sample Algorithm .....                   | 130 |
| Table 5–58 Validation Settlement: C1-VALSET .....                                  | 131 |
| Table 5–59 Validation Settlement: Sample Algorithm .....                           | 131 |
| Table 5–60 Initiate LMI Process: C1-INITLMI .....                                  | 132 |
| Table 5–61 Initiate LMI Process: Sample Algorithm .....                            | 133 |
| Table 5–62 PTP Active Algorithm: C1-PTPACTIVE .....                                | 133 |
| Table 5–63 PTP Active Algorithm: Sample Algorithm .....                            | 134 |
| Table 5–64 PTP Kept Algorithm: C1-PTPKEPT .....                                    | 134 |
| Table 5–65 PTP Active Algorithm: Sample Algorithm .....                            | 135 |
| Table 5–66 PTP Broken Algorithm: C1-BRKPTPNGP .....                                | 136 |
| Table 5–67 Rule Facts Populating Algorithm: C1-BRLSR .....                         | 137 |
| Table 5–68 Sample Algorithm .....  | 139 |
| Table 5–69 Borrower Level: C1-ASSODELAC .....                                      | 140 |



---

|  |     |
|--|-----|
| Table 5–70 Borrower Level : C1-BRWRSW_Y .....  | 140 |
| Table 5–71 Borrower Level : C1-BRWTRNDF .....  | 140 |
| Table 5–72 Borrower Level : C1-BRWRSW_N .....  | 141 |
| Table 5–73 Person Address Update -Pre-Processing: C1-PADDPRE .....   | 141 |
| Table 5–74 Collection Address Post Processing: C1-PERADDP .....  | 141 |
| Table 5–75 Person Contact Point Update - Pre Processing: C1-PCONTPRE .....   | 142 |
| Table 5–76 Check if Special Case Already Exist on the Customer- Enter Pro-<br>cessing: C1-CKSPLCASE .....  | 142 |
| Table 5–77 Pull all the non delinquent accounts of the customer into collections -<br>Enter Processing: C1-PullINDAcc .....                        | 143 |
| Table 5–78 Associate all accounts to the case where customer is a primary bor-<br>rower- Enter Processing: C1-ASSCTEACC .....                      | 144 |
| Table 5–79 Exclude all the associated accounts from Dialer- Enter Processing:<br>C1-ExcAccDir .....  | 144 |
| Table 5–80 Initiate Collateral Valuation for all collaterals whose last valuation<br>was done 'X' days before- Enter Processing: C1-IniClVal ..... | 145 |
| Table 5–81 Monitor if any of the associated account need to be charged off and<br>monitor delinquency- Monitoring: C1-MTRCRGDQY .....              | 146 |
| Table 5–82 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-<br>MTR341HRG .....   | 147 |
| Table 5–83 Monitor if the payment plan on any of the associated accounts is<br>Broken for more than x days- Monitoring: C1-MTRPYMPLN .....         | 148 |
| Table 5–84 Notify the Bankruptcy Specialist if the Liquidation reaches a specific<br>status- Monitoring: C1-MNTRASLQD .....                        | 148 |
| Table 5–85 Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring:<br>C1-MTRHRNGDT .....   | 149 |
| Table 5–86 Determine in which status the case should proceed for Bankruptcy<br>Treatment- Post Processing C1-DTMBKTRTM .....                       | 150 |
| Table 5–87 Validate if appropriate Case Details have been entered by the user-<br>Post Processing C1-VLBCDATA .....                                | 151 |

---

|  |     |
|--|-----|
| Table 5–88 Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK .....        | 153 |
| Table 5–89 Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP .....                             | 153 |
| Table 5–90 Joint Bankruptcy - Associate other customers to the Bankruptcy case C1-ASSCUSTJB .....                          | 154 |
| Table 5–91 Create Pay Plan for a Bankruptcy Case - Enter Processing: C1-CRTATP .....                                       | 154 |
| Table 5–92 Close Pay Plan for a Bankruptcy Case - Exit Processing: C1-CLDATP .....   | 155 |
| Table 5–93 Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days - Monitoring: C1-MTRARPLNT ..... | 155 |
| Table 5–94 Set or Reset Account level Warning Indicator for Bankruptcy - Enter Processing: C1-SETWI .....                  | 156 |
| Table 5–95 Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing C1-TSKVNDR .....    | 157 |
| Table 5–96 Hardship - Associate Accounts of Main Customer - Enter Processing C1-HARASOPND .....                            | 158 |
| Table 5–97 Transition to Contact Statuses - Monitoring C1-ECIC .....   | 158 |
| Table 5–98 Park Small Balance Accounts - Monitoring C1-ECPSBA .....  | 160 |
| Table 5–99 Initiate Skip Tracking - No Telephone Number- Enter Processing C1-ECISTNTN .....                                | 160 |
| Table 5–100 Initiate Skip Tracking - No Telephone Number- Monitoring C1-ECTTSS .....                                       | 161 |
| Table 5–101 Validate Contact Cap- Monitoring C1-ECVCC .....  | 162 |
| Table 5–102 Schedule Contact - Monitoring C1-ECSC .....  | 163 |
| Table 5–103 Initiate Skip Tracing - Wrong Telephone Number- Monitoring C1-ECISTITN .....                                   | 164 |
| Table 5–104 Transition to Under Resolution Status- Monitoring C1-ECTTURS .....   | 165 |
| Table 5–105 Resume Contact From Under Resolution- Monitoring C1-RCFR .....   | 166 |
| Table 5–106 Resume Contact from Small Balance- Monitoring C1-ECRCSB .....  | 167 |

---

|  |     |
|--|-----|
| Table 5–107 Determine Contact Intensity - Monitoring C1-ECDCI .....  | 168 |
| Table 5–108 Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing C1-CTRANTCRE ..... | 169 |
| Table 5–109 Refer to Supervisor- Result Type - Post Processing C1-ECRTS .....  | 171 |
| Table 5–110 Resume Collections- Result Type - Post Processing C1-RESCOLL .....   | 172 |
| Table 5–111 Create case on Follow up- Result Type - Post Processing C1-CRETCSFL .....  | 173 |
| Table 5–112 Hold Case- Result Type - Post Processing C1-HOLDCASE .....   | 173 |
| Table 5–113 Set Case Data- Result Type - Enter Processing C1-ECUPCASE .....  | 174 |
| Table 5–114 This algorithm will transition the case status to the Suspension status if Cease and Desist = Y C1-CSETRANS .....                | 175 |
| Table 5–115 Algorithm is used for scheduling call C1-SCHCALL .....   | 176 |
| Table 5–116 Reset WI in the host C1-RESETWISCHCALL .....   | 177 |
| Table 5–117 Validate Collateral - Enter Validation C1-VALDCOLL .....   | 177 |
| Table 5–118 Validate Demand Letter and Acceleration Letter - Enter Validation C1-VALIDDLAL .....   | 178 |
| Table 5–119 Associate Customers in Repossession Case - Enter Validation C1-ASSOCUST .....  | 179 |
| Table 5–120 Bankruptcy Check on Associate Customers - Enter Status C1-CHKBKPTCY .....  | 180 |
| Table 5–121 Monitor if Demand letter and Acceleration letter have been sent on the account. C1-MNTRDLAL .....                                | 180 |
| Table 5–122 Auto Approval Check for Repossession C1-REPOAPRV .....   | 182 |
| Table 5–123 Repossession Setup Complete C1-RSTUPCMPL .....   | 183 |
| Table 5–124 Automatic task creation for vendors C1-AUTOTASKC .....   | 183 |
| Table 5–125 Notify Repossession Specialist on Task Completion C1-NOTRSTSK .....  | 184 |
| Table 5–126 Automatic sending of Redemption letters C1-REDEMPLTR .....   | 185 |
| Table 5–127 Monitor for Redemption Proceeds C1-REDEPROC .....  | 186 |

---

|  |     |
|--|-----|
| Table 5–128 Validate if appropriate Case Details have been entered by the user and transition C1-VALDATAPR ..... | 186 |
| Table 5–129 Monitor for Liquidation Setup Complete C1-LIQSETCMP .....  | 187 |
| Table 5–130 Send Repossession Alert to Vendor C1-REPOASAL .....  | 188 |
| Table 5–131 Extract Algorithm Repossession Assignment C1-REPEMTEMP .....   | 189 |
| Table 5–132 Monitor Redemption Clear Date C1-REDCLRDT .....  | 189 |
| Table 5–133 Result Post Processing Algorithm for Approvals C1-RAPRVRSLT ...                                      | 190 |
| Table 5–134 Adhoc Characteristic Value Validation Algorithm PASTDATE_VAL ..                                      | 190 |
| Table 5–135 Result Post Processing Algorithm for Redemption Clear Date C1-RDEEMDATE .....                        | 192 |
| Table 5–136 Update Review Date for associated accounts C1-UPDRVWDT .....   | 192 |
| Table 5–137 Case Monitoring CS-MONITOR .....   | 193 |
| Table 5–138 Update warning indicator for the customer C1-UPDWARN .....   | 194 |
| Table 5–139 Transition to Default Next Status C1-TRAN-STAT .....   | 195 |
| Table 5–140 Set Account Warning Indicator C1-ACTCSWGID .....   | 195 |
| Table 5–141 Timezone derivation field update algorithm C1-TZDRFLD .....  | 195 |
| Table 5–142 Validate Task Completion (Case Closure) C1-VALTASKCM .....   | 196 |
| Table 5–143 Validate Task Completion C1-VALTASKEX .....  | 196 |
| Table 5–144 Automatic Task Creation when case enters a particular status C1-CREATTASK .....                      | 197 |
| Table 5–145 Set Account Warning Indicator .....  | 198 |
| Table 5–146 Legal vendor Allocation C1-LGLVNDRAL .....   | 198 |
| Table 5–147 Validate Extended Expiry Date C1-VAL-EXPDT .....   | 199 |
| Table 5–148 Extend Expiry Date C1-EXT-EXPDT .....  | 199 |
| Table 5–149 Capture Case Status Update Date/Time C1-CASE-STAT .....  | 200 |
| Table 5–150 Create To Do C1-TO-DO .....  | 200 |
| Table 5–151 Check customer eligibility C1-CHKCUST .....  | 201 |

---

|   |     |
|---|-----|
| Table 5–152 Capture Hardship Approval Date C1-HARAP-DT .....  | 203 |
| Table 5–153 Algorithm that will interface with Rule Engine. C1-RULEADAPT .....                                      | 203 |
| Table 5–154 Perform Validation for Collateral C1-VRFYCOL .....  | 203 |
| Table 5–155 Check target account eligibility C1-CHKTRGT .....   | 204 |
| Table 5–156 Approval check for set-off transaction C1-ROSOAPPR .....  | 205 |
| Table 5–157 Get target accounts C1-GETTRGT .....  | 206 |
| Table 5–158 Update status of relief to Expired in Hardship C1-UPDHDSTAT .....                                       | 208 |
| Table 5–159 To Do Completion for case C1-TO-DO-END .....  | 208 |
| Table 5–160 Update Marketing Consent flag C1-MKT-FLG .....  | 209 |
| Table 5–161 Check Default Notice for Voluntary possession C1-CHKDFLT .....  | 209 |
| Table 5–162 Check Submission Date C1-CHKSUBDT2 .....  | 210 |
| Table 5–163 Update Financial Hardship flag C1-FNHRD-FLG .....   | 210 |
| Table 5–164 Update Financial Hardship flag C1-FNHRD-FLG .....   | 211 |
| Table 5–165 Algorithm to see if case is running before closing. C1-CHKCASE .....                                    | 211 |
| Table 5–166 Check Deceased status for the customer .C1-CHKDCD .....   | 212 |
| Table 5–167 Associated accounts with deceased customer case.C1-DCDACCTS   | 212 |
| Table 5–168 Execute Fund Transfer C1-FUNDTRFR .....   | 213 |
| Table 5–169 Algorithm to save previous state's status code.C1-SAVPRESTA .....                                       | 214 |
| Table 5–170 Attach case type from feature config attach to BO. C1-ATCHCS .....                                      | 214 |
| Table 5–171 Update Collection Warning Indicator. C1-UPD-WRIND .....   | 215 |
| Table 5–172 Hardship Entity Association to nominated accounts and financial<br>owners of account C1-HARDASSOD ..... | 216 |
| Table 5–173 Assign Applicable Relief Type C1-RELIF-TYP .....  | 216 |
| Table 5–174 Create Customer Contact for Resultype Algo. C1-CREATCC .....  | 217 |
| Table 5–175 Calculate an expiry date when entering case status. C1-CSEXPDT .....                                    | 217 |
| Table 5–176 Create Customer Contact. C1-CUST-CONT .....   | 218 |
| Table 5–177 Transition to Next Status x days before expiry C1-NXT-BX-DY .....                                       | 219 |

---

|   |     |
|---|-----|
| Table 5–178 Validate Hardship Expiry Date. C1-VAL-FHEXP .....                         | 219 |
| Table 5–179 Update Account in collections flag. C1-ACTINCOL .....                     | 220 |
| Table 5–180 Associate related entities with the case. C1-ARSENTITY .....              | 220 |
| Table 5–181 Revalidate target account. C1-REVALTRGT .....                             | 221 |
| Table 5–182 Initiate LMI. C1-INITLMIS .....   | 223 |
| Table 5–183 NGP Collection case creation algorithm. C1-COLLCASE .....                 | 224 |
| Table 5–184 Stop Contract Algorithm. C1-CONTSTOP .....                                | 225 |
| Table 5–185 Check for existing Hardship. C1-CHKHRDSHP .....                           | 225 |
| Table 5–186 Algorithm for contact processing. C1-CNTCT .....                          | 226 |
| Table 5–187 Check application expiry date. C1-CHKEXP .....                            | 226 |
| Table 5–188 New Customer Contact Creation Algorithm. C1-CCCREATE .....                | 226 |
| Table 5–189 Removes a case characteristic on case status exit. C1-REMCSCHE .....      | 227 |
| Table 5–190 Transition case on a date on a case characteristic. C1-TRANSDET .....     | 227 |
| Table 5–191 Set Account Nxt Credit Review Date to current date. C1-<br>NXTRVWDT ..... | 228 |
| Table 5–192 Mark accounts for strategy review. C1-REVIW-ACT .....                     | 228 |
| Table 5–193 Wait Time Out (in days). C1-WAIT-DAYS .....                               | 229 |
| Table 5–194 Validate Hardship Application inputs. C1-V-FH-APP .....                   | 229 |
| Table 5–195 Check for Operational Relief Types .C1-OP-RT .....                        | 230 |
| Table 5–196 Auto-Approval Check. C1-FH-AUTOAP .....                                   | 231 |
| Table 5–197 Apply Hardship Relief Types for accounts in Host. C1-FH-EVAL .....        | 231 |
| Table 5–198 Update Party Warning Indicator. C1-UPD-PRTWI .....                        | 232 |
| Table 5–199 Transition to Next status when all reliefs are app. C1-RAPP .....         | 233 |
| Table 5–200 Collection - Entity Activity Population. C1-ENTACTPOP .....               | 233 |
| Table 5–201 Cancel Hardship Application. C1-CXLFH .....                               | 234 |
| Table 5–202 Perform Payment Transfer for ROSO. C1-ROSOPMTXR .....                     | 234 |
| Table 5–203 Validate ROSO Target Account inputs C1-RS-VALIN .....                     | 235 |

---

|  |     |
|--|-----|
| Table 5–204 Create RMB Entities from Host Data. C1-VCREATE .....               | 235 |
| Table 5–205 Populate Activity Table For Notes Creation. C1-NACTIVITY .....     | 236 |
| Table 5–206 Suspend Activity for Account Pre Processing C1-USRALCRR .....      | 236 |
| Table 5–207 Sample TAM Algorithm Type. C1-TAMALG .....                         | 237 |
| Table 5–208 Cancel Approval Request. C1-CANAPPR .....                          | 237 |
| Table 5–209 Update Customer in collections flag. C1-CUSINCOL .....             | 238 |
| Table 5–210 Set Display Date. C1-SETDSPDT .....                                | 239 |
| Table 5–211 Transition to Default next status after N Days. C1-TRNDFLT .....   | 239 |
| Table 5–212 Check current cases on account for exclusion. C1-EXCLCASE .....    | 240 |
| Table 5–213 Update Collateral Status in the host. C1-UPCOLLST .....            | 241 |
| Table 5–214 Initiate collateral valuation. C1-COLLVAL .....                    | 242 |
| Table 5–215 Mandatory characteristics check for Asset Repo. C1-CHARVAL .....   | 242 |
| Table 5–216 Update Collateral Status in the Host. C1-UPCOLLSTS .....           | 243 |
| Table 5–217 Set exclusion date for delinquent account. C1-EXCLROSO .....       | 244 |
| Table 5–218 Cancel Set-off. C1-CANROSO .....                                   | 245 |
| Table 5–219 Complete Set-off. C1-COMPROSO .....                                | 245 |
| Table 5–220 Reverse Set-off. C1-REVROSO .....                                  | 246 |
| Table 5–221 Algorithm type for update case id for Insurance. C1-UPCASFINS .... | 246 |
| Table 5–222 Case Creation on enter processing. C1-CCOENTER .....               | 246 |
| Table 5–223 Collection - Case Creation On Exit of Status. C1-CCOE .....        | 247 |
| Table 5–224 Action category Validation algorithm. C1-ACTCAT .....              | 247 |
| Table 5–225 Action Type Algorithm Type. C1-ACTTYP .....                        | 248 |
| Table 5–226 Case Type Status Mapping Algorithm Type C1-CASETYMP .....          | 248 |
| Table 5–227 Collection - Close Processing Algorithm. C1-CCALG .....            | 248 |
| Table 5–228 Algorithm type for case list update. C1-CASELIST .....             | 249 |
| Table 5–229 Copy Characteristics Algorithm. C1-COPYCHAR .....                  | 250 |

---

|  |     |
|--|-----|
| Table 5–230 Call Advice - Red/Green logic calculation. C1-CALADVICE .....                          | 250 |
| Table 5–231 Task Case Mapping Validation Algorithm. C1-TCVAL .....                                 | 251 |
| Table 5–232 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status. C1-DISMON ..... | 252 |
| Table 5–233 SLA Parameters validation algorithm. C1-SLAPARAM .....                                 | 253 |
| Table 5–234 Case Group add validation algorithm. C1-CGVAL .....                                    | 254 |
| Table 5–235 Collection - Get Strategy Algorithm. C1-COLGS .....                                    | 254 |
| Table 5–236 Create/Move Collection Strategy Cases for Account. C1-COLOMR ..                        | 254 |
| Table 5–237 Collection - Case SA Update for Manual Creation. C1-CSAUPD .....                       | 256 |
| Table 5–238 Promise to Pay - Additional Grace Days Sample Algo. C1-PPADDLGRD .....                 | 256 |
| Table 5–239 Promise to Pay Threshold Percentage. C1-PPTHRESH .....                                 | 256 |
| Table 5–240 Result type Post Processing Case Transition Algo. C1-RTPCC .....                       | 257 |
| Table 5–241 Inbound Customer algorithm. C1-IN-CUST .....   | 258 |
| Table 5–242 Result Type Pre-processing Algorithm Type. C1-RSTPRE .....                             | 259 |
| Table 5–243 Result Type Post-processing Algorithm Type. C1-RSTPOST .....                           | 259 |
| Table 5–244 Characteristic Type :Validate Date Field (Custom). C1-ADHDATE ..                       | 259 |
| Table 5–245 Algorithm Type for Dialer Results Upload. C1-DLRRSUPLD .....                           | 260 |
| Table 5–246 Algorithm for Hardship case creation activity. C1-CRTHDSP .....                        | 260 |
| Table 5–247 This Algorithm is used to abort Approval work item. C1-ABORTAPP                        | 261 |
| Table 5–248 Cancel Process Approval Request:Financial Hardship. C1-CANFHAPPR .....                 | 262 |
| Table 5–249 Hardship Characteristic Association. C1-FHCHARASC .....                                | 262 |
| Table 5–250 Pre-Populated system facts to be used for Rule. C1-PPSF .....                          | 263 |
| Table 5–251 Assign Batch level TODOs(task) to a queue. C1-ASGNTASK .....                           | 263 |
| Table 5–252 Validate Date Field :Custom. C1-ADHV-DTD .....   | 263 |
| Table 5–253 Characteristic Date field Validation. C1-CHARDTVVAL .....                              | 265 |



---

|   |     |
|---|-----|
| Table 5–254 Retry Case in Error C1-RCASEE .....   | 266 |
| Table 5–255 Allocate Queue for Customer Level Case. C1-ALLOCQUE .....   | 267 |
| Table 5–256 Person Address – Collection. C1-PERADDRRC .....   | 267 |
| Table 5–257 Person Contact Point Update - Post Processing. C1-PERCONTPP ..  | 268 |
| Table 5–258 Update Self Serve Flag Algorithm. C1-SELFSERVE .....  | 268 |
| Table 5–259 Create Task for Self Serve Request for Assistance transaction. C1-<br>FLWRTSK .....                                 | 268 |
| Table 5–260 Transition to Under Settlement Offer Status. C1-TSOF .....  | 269 |
| Table 5–261 Create PTP Validation Algorithm. C1-VPTP .....  | 269 |
| Table 5–262 Under Settlement Offer Status Monitor Algorithm for Settlement<br>Expiry/Offer Acceptance/Rejection. C1-MONUS ..... | 270 |
| Table 5–263 Algorithm to identify Partial Charge off Accounts BATCHJOBPARM  | 271 |
| Table 5–264 Monitor Settlement Offer Payments C1-MSOP .....   | 272 |
| Table 6–1 Case Transition for Active Service Member C1-ACTMEMCHK .....  | 274 |
| Table 6–2 Active Military Check on Associated Customers - Enter Validation C1-<br>BLOCKREPO .....                               | 275 |
| Table 6–3 Metro 2 Reporting - Account Status Code post Liquidation C1-<br>ASCLIQU .....   | 275 |
| Table 6–4 Metro 2 Reporting - Account Status Code C1- ASCREPO .....   | 276 |
| Table 6–5 Metro 2 Reporting - Compliance condition code C1- COMCODE .....   | 277 |
| Table 6–6 Metro 2 Reporting - Marking Account as Close C1- CFOSEP .....   | 278 |
| Table 6–7 Metro 2 Reporting - Consumer Information Indicator C1- CONINFOIN ..   | 278 |
| Table 6–8 Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post<br>Discharge C1- CIIPSTDIS .....                   | 279 |
| Table 6–9 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1-<br>CGCLC .....  | 280 |
| Table 6–10 Metro 2 Reporting - Consumer Now Located (Removes previously<br>reported T Indicator) C1-CNLREM .....                | 280 |

---

|  |     |
|--|-----|
| Table 6–11 Metro 2 Reporting - Set DPD and Outstanding amount to all associated accounts C1-SETDPD ..... | 281 |
| Table 7–1 Feeder Services .....  | 282 |
| Table 8–1 Generic Data Type .....  | 290 |
| Table 8–2 Summary .....  | 291 |
| Table 8–3 Interface .....  | 291 |
| Table 8–4 Service Management .....   | 292 |
| Table 8–5 Detail Record .....  | 293 |
| Table 8–6 Customer information .....   | 294 |

# 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/us/corporate/accessibility/index.html>.

### Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/us/corporate/accessibility/support/index.html#info> or visit <http://www.oracle.com/us/corporate/accessibility/support/index.html#trs> if you are hearing impaired.

## Organization of the Guide

This document contains:

### [Chapter 1 About this Guide](#)

This chapter provides details about applicability of this guide.

### [Chapter 2 Introduction](#)

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

### [Chapter 3 System Overview](#)

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

### [Chapter 4 Staging Area](#)

This chapter provides details of the feeder tables.

### [Chapter 5 Algorithms](#)

This chapter outlines the pre-shipped algorithm details.

#### [Chapter 6 Localized Algorithms](#)

This chapter provides a list of Localized algorithm details.

#### [Chapter 7 Feeder Services](#)

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

#### [Chapter 8 Dialer Webservice Integration](#)

This chapter provides details of the dialer web service integration.

## Related Documents

For more information, see the following documentation:

- For the complete list of the adapters for integration with Oracle Banking Enterprise Default Management modules and technology stacks such as DMS / Alert /Email systems, see the Oracle Banking Enterprise Default Management 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.                          |
| <code>monospace</code> | Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter. |



# 1 About this Guide

This guide is applicable for the following products:

- Oracle Banking Platform (Oracle Banking Collections and Oracle Banking Recovery)
- Oracle Banking Enterprise Default Management (Oracle Banking Enterprise Collections and Oracle Banking Enterprise Recovery)

References to Oracle Banking Platform or OBP in this guide apply to all the above mentioned products.



## 2 Introduction

In Oracle Banking Platform, Collections 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.



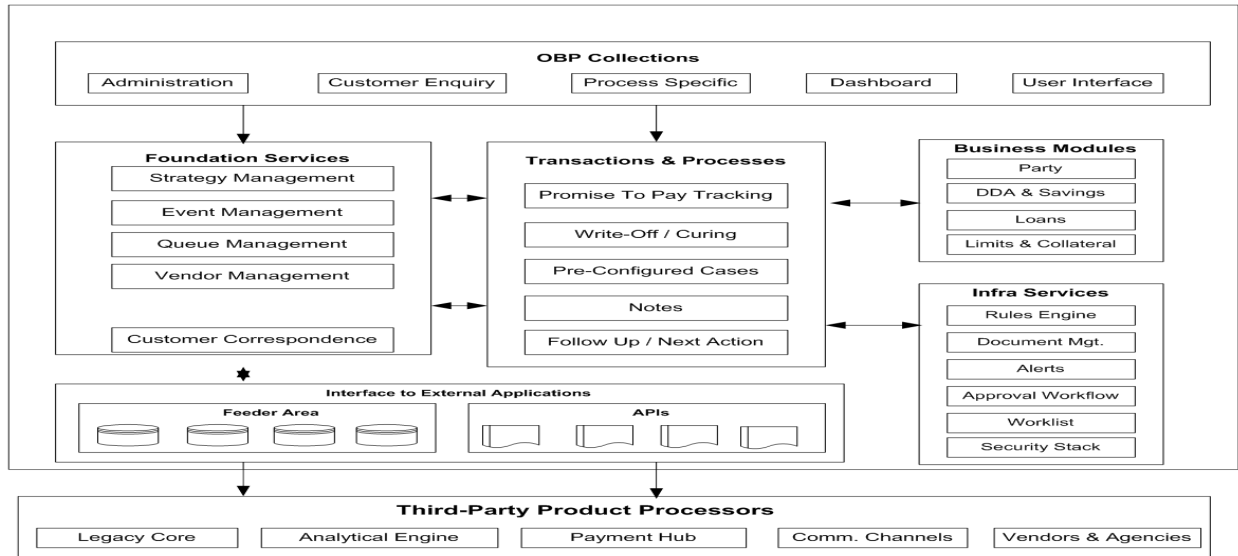


# 3 System Overview

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 3–1 System Overview





# 4 Staging Area

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

## 4.1 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.

### 4.1.1 Account Data

This section provides information on the tables related to accounts.

#### 4.1.1.1 Account Details

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

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

*Table 4–1 Account Details*

| Field Name    | Description  | Values | 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   |
| Business Unit | Business Unit of the Account. This field is used only if multi-branding features are to be used.           |        | VARCHAR2  | 40     | N        | BUSINESS_UNIT |
| Market Entity | Market Entity to which account belongs. This field is used only if multi-branding features are to be used. |        | VARCHAR2  | 40     | N        | MARKET_ENTITY |
| Facility ID   | Facility ID under which account is created. This field is used based on the structure of accounts in the   |        | VARCHAR2  | 40     | N        | FACILITY_ID   |

#### 4.1 Feeder Tables

| Field Name                  | Description  | Values   | Data Type | Length | Required | Column Name            |
|-----------------------------|--|--|-----------|--------|----------|------------------------|
|                             | host.  |  |           |        |          |                        |
| 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  | 30     | 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                      | Number of  |  | NUMBER    | 3,0    | N        | REDRAW_CNT             |

| Field Name                    | Description   | Values | Data Type | Length | Required | Column Name               |
|-------------------------------|---|--------|-----------|--------|----------|---------------------------|
| Count                         | times a redraw has been performed   |        |           |        |          |                           |
| 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            |
| 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                |

4.1 Feeder Tables

| Field Name  | Description  | Values  | Data Type | Length | Required | Column Name     |
|---|--|---|-----------|--------|----------|-----------------|
| Account Opening Date or Initial Disbursement Date | Term Loan:<br>First Disbursement Date<br>OD: Date on which OD facility is provided<br>Current Account with TOD facility:<br>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 + AUF Limit Utilized + Overdue Amount<br><b>Term Loans :</b><br>Outstanding Principal - RPA Balance + Overdue Amount | NUMBER    | 36,18  | Y        | OUTSTANDING_AMT |
| Overdue Amount                                    | Overdue amount for the account   | <b>OD Accounts:</b><br>TOD utilized + Overline utilized + Temporary Excess utilized<br><b>Term Loans :</b> All amounts due and still unpaid                   | NUMBER    | 36,18  | Y        | OVERDUE_AMT     |
| Account Limit                                     | Sanctioned Limit offered to the account  | <b>OD Accounts :</b><br>OD limit + Temporary Excess limit<br><b>Term</b>  | NUMBER    | 36,18  | Y        | OVERLIMIT_AMT   |

| Field Name                       | Description  | Values  | Data Type | Length | Required | Column Name         |
|----------------------------------|--|---|-----------|--------|----------|---------------------|
|                                  |  | <b>Loans :</b><br>Sanctioned<br>Amount  |           |        |          |                     |
| 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        | INSTALLMENT_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    |



#### 4.1 Feeder Tables

| Field Name            | Description   | Values   | Data Type | Length | Required | Column Name               |
|-----------------------|---|--|-----------|--------|----------|---------------------------|
| Interest Rate         | Rate of interest for current applicable stage                     |  | NUMBER    | 5,0    | Y        | INT_RATE                  |
| Interest Type         | Fixed or Floating   |  | VARCHAR2  | 14     | Y        | INT_TYPE                  |
| Address Type Code     | Overriding address type configured for an account                 |  | VARCHAR2  | 20     | N        | ADDR_TYPE_CD              |
| Employee Account Flag | Indicate if the account belongs to a bank employee                | Y/N  | VARCHAR2  | 1      | Y        | EMPLOYEE_ACCT_FLG         |
| Minor Account Status  | Indicate if the account belongs to a minor                        | Y/N  | VARCHAR2  | 40     | Y        | MINOR_ACCOUNT_STATUS_TYPE |
| Home Branch           | Home Branch of the account  |  | VARCHAR2  | 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             | VARCHAR2  | 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              | VARCHAR2  | 60     | N        | UDF2                      |
| User Defined Field 3  | User Defined Field in case any additional attributes are required | <b>Expected Loss :</b><br>String value coming from third party interface                   | VARCHAR2  | 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 | VARCHAR2  | 60     | N        | UDF4                      |
| User Defined          | User Defined  |  | VARCHAR2  | 60     | N        | UDF5                      |

| Field Name            | Description   | Values | Data Type | Length | Required | Column Name |
|-----------------------|---|--------|-----------|--------|----------|-------------|
| Field 5               | Field in case any additional attributes are required              |        |           |        |          |             |
| User Defined Field 6  | User Defined Field in case any additional attributes are required |        | VARCHAR2  | 60     | N        | UDF6        |
| User Defined Field 7  | User Defined Field in case any additional attributes are required |        | VARCHAR2  | 60     | N        | UDF7        |
| User Defined Field 8  | User Defined Field in case any additional attributes are required |        | VARCHAR2  | 60     | N        | UDF8        |
| User Defined Field 9  | User Defined Field in case any additional attributes are required |        | VARCHAR2  | 60     | N        | UDF9        |
| 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       |

#### 4.1 Feeder Tables

| Field Name             | Description   | Values | Data Type | Length | Required | Column Name                |
|------------------------|---|--------|-----------|--------|----------|----------------------------|
| 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_DELIQUENCY |
| Redraw Availability    | Facility to redraw loan   | Y/N    | CHAR      | 1      | Y        | FDR_REDRAW_AVL_SW          |
| Joint Applicant        | Indicates if the account has a Joint Applicant                              | Y/N    | VARCHAR2  | 1      | Y        | FDR_JOINT_APPLICANT_SW     |
| Delinquent             | Indicates if the account is delinquent                                      | Y/N    | VARCHAR2  | 1      | Y        | FDR_IS_DELIQUENT_SW        |
| Non Starter            | Indicates if the customer defaults the first installment after disbursement | Y/N    | VARCHAR2  | 1      | Y        | FDR_NON_STARTER_SW         |
| Behavior Score         | Current Behavior Score  |        | VARCHAR2  | 10     | N        | FDR_BEHAVIOR_              |

| Field Name                | Description   | Values                   | Data Type | Length | Required | Column Name              |
|---------------------------|---|--------------------------|-----------|--------|----------|--------------------------|
|                           | captured at account level   |                          |           |        |          | 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               | Signifies if the data is created initially or is updated 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                |
| BICOE loan account Switch | BICOE loan account Switch   |                          | CHAR      | 1      | N        | BICOE_LOAN_SW            |
| Customer Class Code       | Customer Class Code   |                          | VARCHAR2  | 8      | N        | CUST_CL_CD               |

#### 4.1 Feeder Tables

| Field Name                     | Description                   | Values | Data Type | Length | Required | Column Name           |
|--------------------------------|-------------------------------|--------|-----------|--------|----------|-----------------------|
| First Default date             | First Default date            |        | DATE      | 10     | N        | FIRST_DEFAULT_DATE    |
| Last Days Past Due update Date | Last Days Past Due            |        | DATE      | 10     | N        | LAST_DPD_UPDATE_DT    |
| Relationship Officer Code      | Relationship Officer Code     |        | VARCHAR2  | 40     | N        | RELATION_OFFICER_CODE |
| Feeder Forced Switch           | FDR Forced SW                 |        | VARCHAR2  | 1      | Y        | FDR_FORCED_SW         |
| Forced Reason Code             | Forced Reason CD              |        | VARCHAR2  | 4      | Y        | FORCED_REASON_CD      |
| IOA Balance Amount             | IOA Balance Amount            |        | NUMBER    | 36,18  | N        | IOA_BALANCE_AMT       |
| Autopay Instructions           | Autopay Instructions          |        | VARCHAR2  | 30     | N        | AUTO_PAY_INSTRUCTIONS |
| Charge off Date                | Charge off Date               |        | DATE      |        | N        | CHARGE_OFF_DT         |
| Service Member Benefit         | Service Member Benefit Switch |        | NUMBER    | 1      | Y        | SCRA_BENEFIT_SW       |
| Charge off Amount              | Charge off Amount             |        | NUMBER    | 36,18  | N        | CHARGE_OFF_AMT        |
| Number of Time Re-aged         | Number of Time Re-aged        |        | NUMBER    | 4      | N        | NUM_OF_TIME_REAGED    |
| Number of Time Extended        | Number of Time Extended       |        | NUMBER    | 4      | N        | NUM_OF_TIME_EXTENDED  |
| Number of Time Deferred        | Number of Time Deferred       |        | NUMBER    | 4      | N        | NUM_OF_TIME_DEFERRED  |
| Number of Time Renewed         | Number of Time Renewed        |        | NUMBER    | 4      | N        | NUM_OF_TIME_RENEWED   |
| Number of Time Re-written      | Number of Time Re-written     |        | NUMBER    | 4      | N        | NUM_OF_TIME_REWRITTEN |
| Billing Switch                 | Billing Switch                |        | NUMBER    | 1      | N        | BILLING_SW            |
| Account Reopen Date            | Account Reopen Date           |        | DATE      |        | N        | ACCT_REOPEN_DT        |
| Account Closed Date            | Account Closed Date           |        | DATE      |        | N        | ACCT_CLOSED_DT        |

| Field Name                      | Description                     | Values | Data Type | Length | Required | Column Name              |
|---------------------------------|---------------------------------|--------|-----------|--------|----------|--------------------------|
| Account Reopen Switch           | Account Reopen Switch           |        | NUMBER    | 1      | N        | ACCT_REOPEN_SW           |
| Charge Off Primary Reason       | Charge Off Primary Reason       |        | VARCHAR2  | 60     | N        | CHARGE_OFF_PRIMARY_RSN   |
| Charge Off Secondary Reason     | Charge Off Secondary Reason     |        | VARCHAR2  | 60     | N        | CHARGE_OFF_SECONDARY_RSN |
| Recovery Score                  | Recovery Score                  |        | VARCHAR2  | 10     | N        | RECOVERY_SCORE           |
| Fee Charge                      | Fee Charge                      |        | NUMBER    | 36,18  | N        | FEE_CHARGES              |
| Insurance Amount                | Insurance Amount                |        | NUMBER    | 36,18  | N        | INSURANCE                |
| Interest Amount                 | Interest Amount                 |        | NUMBER    | 36,18  | N        | INTEREST                 |
| Principal Amount                | Principal Amount                |        | NUMBER    | 36,18  | N        | PRINCIPAL_AMT            |
| Interest Rate                   | Interest Rate                   |        | NUMBER    | 36,18  | N        | INTEREST_RATE            |
| Account Term                    | Account Term                    |        | NUMBER    | 4      | N        | ACCT_TERM                |
| Account Title                   | Account Title                   |        | VARCHAR2  | 500    | N        | ACCT_TITLE               |
| Type Code                       | Account Type Code               |        | VARCHAR2  | 8      | N        | ACCT_TYPE_CD             |
| Asset Class Value               | Asset Class Value               |        | VARCHAR2  | 60     | N        | ASST_CLASS_VALUE         |
| Feeder Regulated Account Switch | Feeder Regulated Account Switch |        | VARCHAR2  | 1      | N        | FDR_REGULATED_ACCOUNT_SW |
| Feeder Regulated Account Switch | Feeder Regulated Account Switch |        | VARCHAR2  | 1      | N        | FDR_REGULATED_ACCOUNT_SW |
| Message Category                | Message Category                |        | NUMBER    | 5,0    | N        | MESSAGE_CAT_NBR          |
| Message Number                  | Message Number                  |        | NUMBER    | 5,0    | N        | MESSAGE_NBR              |
| Process Status                  | Process Status                  |        | VARCHAR2  | 1      | N        | PROCESS_STATUS           |
| Record Update Date              | Record Update Date              |        | DATE      |        | N        | RECORD_UPDATE_DT         |

| Field Name                                | Description                               | Values | Data Type | Length | Required | Column Name             |
|---|---|--------|-----------|--------|----------|-------------------------|
| Record Exists Switch                      | Record Exists Switch                      |        | VARCHAR2  | 1      | N        | RECORD_EXISTS_SW        |
| RMB CIS Division                          | RMB CIS Division                          |        | CHAR      | 5      | N        | RMB_CIS_DIVISION        |
| RMB COLL Code                             | RMB COLL Code                             |        | VARCHAR2  | 10     | N        | RMB_COLL_CL_CD          |
| RMB Custom Code                           | RMB Custom Code                           |        | VARCHAR2  | 10     | N        | RMB_CUST_CL_CD          |
| RMB Debt Code                             | RMB Debt Code                             |        | VARCHAR2  | 10     | N        | RMB_DEBT_CL_CD          |
| RMB Service Agreement Type Code           | RMB Service Agreement Type Code           |        | CHAR      | 8      | N        | RMB_SA_TYPE_CD          |
| User Defined Instant Switch               | User Defined Instant Switch               |        | VARCHAR2  | 1      | N        | USR_DEF_INST_SW         |
| Account Non Due Amount                    | Account Non Due Amount                    |        | NUMBER    | 36,18  | N        | ACT_NON_DUE_AMT         |
| Over Due Amount of Payment Tracker        | Over Due Amount of Payment Tracker        |        | NUMBER    | 36,18  | N        | PAY_TRACK_OVERDUE_AMT   |
| Over Due Days Past Due of Payment Tracker | Over Due Days Past Due of Payment Tracker |        | NUMBER    | 4      | N        | PAY_TRACK_DAYS_PAST_DUE |

#### 4.1.1.2 Account Arrears Details

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

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

**Table 4–2 Account Arrears Details**

| Field Name      | Description                               | Values | 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 |
| Arrear Type     | Arrear type like interest, fee, and so on |        | VARCHAR2  | 40     | N        | ARS_TYPE      |

| Field Name              | Description   | Values                   | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
| Arrear Amount           | Total arrear rose per arrear type. Details of arrear type should be sent only where arrear amount > 0 |                          | NUMBER    | 36,18  | N        | 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  | N        | 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    | 5,0    | N        | INSTALLMENT_NUM  |
| Record Creation Date    | Date on which data is fed to Collections.   |                          | DATE      | 10     | Y        | CRET_DTTM        |
| Record Type             | Signifies if the data is created initially or is updated for existing data                            | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE         |
| Process Status          | To check the current status of process. Default is P-Pending.   |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated   |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether record is already available or not   |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |
| RES due date            | RES due date  |                          | DATE      | 10     | N        | ARS_DUE_DT       |
| Sub Arrear Type         | Sub Arrear Type   |                          | VARCHAR2  | 40     | Y        | SUB_ARREAR_TYPE  |



| Field Name           | Description          | Values | Data Type | Length | Required | Column Name     |
|----------------------|----------------------|--------|-----------|--------|----------|-----------------|
| Account Non Due Flag | Account Non Due Flag |        | VARCHAR2  | 1      | N        | ACT_NON_DUE_FLG |

### 4.1.1.3 Account Hardship Details

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

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

**Table 4–3 Account Hardship Details**

| Field Name                        | Description  | Values | 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        | HARSHIP_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_PAYMNT_WAIVED          |
| 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  | 200    | N        | USR_DISCRTN_MRGN_RSN      |
| Status                            | Current Status of  |        | CHAR      | 60     | N        | STATUS                    |

| Field Name              | Description   | Values                   | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
|                         | Hardship Relief if applicable   |                          |           |        |          |                  |
| Original Relief Type    | Original Relief Type  |                          | VARCHAR2  | 40     | N        | ORIG_RELIEF_TYPE |
| Record Creation Date    | Date on which the data is fed to Collections                              |                          | DATE      | 10     | N        | 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         |
| Process Status          | To check the current status of process. Default is P-Pending.             |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether record is already available or not                       |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.1.4 Account Repayment Schedule

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

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

**Table 4-4 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  | N        | INSTALLMENT_AMT |
| Principal  | Principal component                            |       | NUMBER    | 36,18  | N        | PRINCIPAL_AMT   |

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name        |
|-------------------------|---|--------------------------|-----------|--------|----------|--------------------|
| 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 |
| 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     | N        | 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          |
| Process Status          | To check the current status of the process. Default is P- Pending.        |                          | VARCHAR2  | 1      | Y        | PROCESS_ STATUS    |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_ CAT_ NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_ NBR       |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_ UPDATE_ DT |
| Record Exist Switch     | To check whether record is already available or not                       |                          | VARCHAR2  | 1      | Y        | RECORD_ EXISTS_ SW |

#### 4.1.1.5 Account Warning Indicator

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

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

**Table 4–5 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   |

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
| Warning Indicator Code  | Warning Indicator code as stored in host                                  |                          | VARCHAR2  | 50     | Y        | WARN_IND_CD      |
| Warning Indicator Value | Warning Indicator Value   |                          | VARCHAR2  | 1      | N        | WARN_IND_VAL     |
| Start Date              | Start Date for warning indicator  |                          | DATE      | 10     | N        | START_DT         |
| End Date                | End Date for the warning indicator code                                   |                          | DATE      | 10     | N        | END_DT           |
| Record Creation Date    | Date on which data is fed to Collections                                  |                          | DATE      | 10     | N        | 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         |
| Process Status          | To check the current status of process. Default is P-Pending.             |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category        | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether record is already available or not                       |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.1.6 Minimum Amount Due on Bill (MAD)

**Table Name:** Minimum Amount Due on Bill (CI\_FDR\_MIN\_AMT\_DUE\_BILL)

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

**Table 4–6 Minimum Amount Due Bill**

| Field Name          | Description         | Value | Data Type | Length | Required | Column Name    |
|---------------------|---------------------|-------|-----------|--------|----------|----------------|
| Source Host ID      | Source Host ID      |       | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID |
| Host Account Number | Host Account Number |       | VARCHAR2  | 40     | Y        | HOST_ACCNT_NBR |

| Field Name               | Description   | Value                    | Data Type | Length | Required | Column Name        |
|--------------------------|---|--------------------------|-----------|--------|----------|--------------------|
| Bill Due Date            | Bill Due Date   |                          | DATE      |        | Y        | DUE_DATE           |
| Bill Date                | Bill Date   |                          | DATE      |        | N        | BILL_CYCLE         |
| Minimum Due Amount       | Minimum Due Amount  |                          | NUMBER    | 36,18  | N        | MIN_AMT_DUE        |
| Bill Status on Due Date  | Bill Status on Due Date   |                          | VARCHAR2  | 10     | N        | STATUS_ON_DUE_DATE |
| Total Minimum Amount Due | Total Minimum Amount Due  |                          | NUMBER    | 36,18  | N        | TOT_MIN_AMT_DUE    |
| 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           |
| Process Status           | To check the current status of process. Default is P-Pending.             |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS     |
| Message Category         | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR    |
| Message Number           | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR        |

#### 4.1.1.7 Payment Tracker Details

**Table Name:** Payment Tracker Details (CI\_FDR\_PAY\_TRACKER\_DETLS)

**Description:** This table tracks payments done on an account.

**Table 4–7 Payment Tracker Details**

| Field Name          | Description         | Value | Data Type | Length | Required | Column Name    |
|---------------------|---------------------|-------|-----------|--------|----------|----------------|
| Source Host ID      | Source Host ID      |       | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID |
| Host Account Number | Host Account Number |       | VARCHAR2  | 40     | Y        | HOST_ACCNT_NBR |
| Due Date            | Due Date            |       | DATE      |        | Y        | DUE_DATE       |
| Due Amount          | Due Amount          |       | NUMBER    | 36,18  | N        | DUE_AMOUNT     |
| Payment Amount      | Payment Amount      |       | NUMBER    | 36,18  | N        | PAYMENT_AMOUNT |
| Feeder              | Feeder Account      |       | VARCHAR2  | 3      | N        | FDR_ACCT_      |

| Field Name            | Description   | Value                    | Data Type | Length | Required | Column Name     |
|-----------------------|---|--------------------------|-----------|--------|----------|-----------------|
| Account Currency Code | Currency Code   |                          |           |        |          | CURR_CD         |
| Record Type           | Signifies if the data is created initially or is update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | N        | RCD_TYPE        |
| Process Status        | To check the current status of process. Default is P-Pending.             |                          | VARCHAR2  | 1      | N        | PROCESS_STATUS  |
| Message Category      | Defined error message category  |                          | NUMBER    | 5,0    | N        | MESSAGE_CAT_NBR |
| Message Number        | Error message number  |                          | NUMBER    | 5,0    | N        | MESSAGE_NBR     |

## 4.1.2 Party Data

This section provides information on the tables related to party.

### 4.1.2.1 Party Account Relationship

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

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

**Table 4–8 Account Party Relationship**

| Field Name          | Description   | Value                    | Data Type | Length | Required | Column Name     |
|---------------------|---|--------------------------|-----------|--------|----------|-----------------|
| Source Host ID      | Source Host ID  |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID     |
| Host Account Number | Host Account Number   |                          | VARCHAR2  | 40     | Y        | HOST_ACCNT_NBR  |
| 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        |
| Process Status      | To check the current status of process. Default is P-Pending.             |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS  |
| Message Category    | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR |
| Message Number      | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR     |

## 4.1 Feeder Tables

| Field Name                   | Description   | Value | Data Type | Length | Required | Column Name          |
|------------------------------|---|-------|-----------|--------|----------|----------------------|
| Record Updated Date          | Record Updated Date   |       | DATE      | 7      | N        | RECORD_UPDATE_DT     |
| Record Exists Switch         | Record Exists Switch  |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW     |
| RMB Main Customer Switch     | RMB Main Customer Switch  |       | CHAR      | 1      | N        | RMB_MAIN_CUST        |
| Financial Responsible Switch | Financial Responsible Switch                                      |       | CHAR      | 1      | N        | RMB_FIN_RESP         |
| Internet Banking Switch      | Internet Banking Switch   |       | VARCHAR2  | 1      | N        | FDR_INTERNET_BANK_SW |
| Phone Banking Switch         | Phone Banking Switch  |       | VARCHAR2  | 1      | N        | FDR_PHONE_BANK_SW    |
| Mobile Banking Switch        | Mobile Banking Switch   |       | VARCHAR2  | 1      | N        | FDR_MOBILE_BANK_SW   |
| ATM Switch                   | ATM Switch  |       | VARCHAR2  | 1      | N        | FDR_ATM_SW           |
| Debit Card Switch            | Debit Card Switch   |       | VARCHAR2  | 1      | N        | FDR_DEBITCARD_SW     |
| User Defined Field 1         | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF1                 |
| User Defined Field 2         | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF2                 |
| User Defined Field 3         | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF3                 |
| User Defined Field 4         | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF4                 |
| User Defined Field 5         | User Defined Field  |       | VARCHAR2  | 60     | N        | UDF5                 |

| Field Name                     | Description   | Value | Data Type | Length | Required | Column Name          |
|--------------------------------|---|-------|-----------|--------|----------|----------------------|
|                                | in case any additional attributes are required                    |       |           |        |          |                      |
| User Defined Field 6           | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF6                 |
| User Defined Field 7           | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF7                 |
| User Defined Field 8           | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF8                 |
| User Defined Field 9           | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF9                 |
| User Defined Field 10          | User Defined Field in case any additional attributes are required |       | VARCHAR2  | 60     | N        | UDF10                |
| Account Nickname               | Account Nickname  |       | VARCHAR2  | 120    | N        | ACCT_NICKNAME        |
| Host Customer Number           | Host Customer Number  |       | VARCHAR2  | 40     | N        | HOST_CUST_NBR        |
| Account Relationship Type Code | Account Relationship Type Code                                    |       | VARCHAR2  | 8      | N        | ACCT_REL_TYPE_CD     |
| Creation DateTime              | Creation DateTime   |       | DATE      |        | N        | CRET_DTTM            |
| Corresponde Nomination Switch  | Corresponde Nomination Switch                                     |       | CHAR      | 1      | N        | CORRES_NOMINATION_SW |

#### 4.1.2.2 Party Details

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

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



**Table 4–9 Party Details**

| Field Name  | Description  | Value | Data Type | Length | Required | Column Name              |
|---|--|-------|-----------|--------|----------|--------------------------|
| Feeder Person Id  |  |       | VARCHAR2  | 10     | Y        | FDR_PER_ID               |
| 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  | 40     | N        | PER_CL_CD                |
| Date of Birth / Date of Incorporation/ Date of Trust Deed |  |       | DATE      | 10     | N        | BIRTH_DT                 |
| Marital Status  | Marital Status of Party in case of Individual Customer   |       | VARCHAR2  | 20     | N        | MARITAL_STAT_FLG         |
| Customer Since  |  |       | DATE      | 10     | N        | SETUP_DT                 |
| Gender  | Gender of Individual Customer  |       | VARCHAR2  | 4      | N        | GENDER                   |
| Preferred Language  | Preferred Language of Communication  |       | VARCHAR2  | 3      | N        | LANGUAGE_CD              |
| Marketing Info Flag                                       | Marketing Information Flag to continue communication   |       | VARCHAR2  | 4      | N        | FDR_RECV_MKTG_INFO_FLG   |
| Probability of Default                                    | String value coming from third   |       | VARCHAR2  | 60     | N        | PROBABILITY_OF_DEFLT_VAL |

| Field Name                | Description  | Value | Data Type | Length | Required | Column Name             |
|---------------------------|--|-------|-----------|--------|----------|-------------------------|
|                           | party interface  |       |           |        |          |                         |
| 3rd Party Flag            | Indicates if a third party is associated to the party                              | Y/N   | 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                    |

#### 4.1 Feeder Tables

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name            |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------------|
| User Defined Value 9    | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF9                   |
| User Defined Value 10   | User Defined Fields   |                          | VARCHAR2  | 60     | N        | UDF10                  |
| 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                  |
| Record Creation Date    | Date on which data is fed to Collections                                  |                          | DATE      | 10     | N        | 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               |
| Ability to pay          | Ability to pay  |                          | VARCHAR2  | 4      | N        | FDR_ABILITY_TO_PAY_FLG |
| Realization Stat        | Realization Stat  |                          | VARCHAR2  | 10     | N        | REALIZN_STAT           |
| Process Status          | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS         |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR        |

| Field Name                         | Description                                     | Value | Data Type | Length | Required | Column Name              |
|------------------------------------|---|-------|-----------|--------|----------|--------------------------|
| Message Number                     | Error message number                            |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR              |
| Record Update Date                 | Date on which the record is updated             |       | DATE      | 10     | N        | RECORD_UPDATE_DT         |
| Record Exist Switch                | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW         |
| Enterprise customer number         | OCH Number                                      |       | VARCHAR2  | 60     | N        | FDR_ENTERPRISE_CUST_NBR  |
| Service Member Switch              | Service Member Switch                           |       | NUMBER    | 1      | Y        | SCRA_MEMBER_SW           |
| Service Member Dependent Switch    | Service Member Dependent Switch                 |       | NUMBER    | 1      | Y        | SCRA_MEMBER_DEPENDANT_SW |
| Service Member Benefit Waiver Flag | Service Member Benefit Waiver Flag              |       | NUMBER    | 1      | Y        | SCRA_BENEFIT_WAIVER      |
| Service Member on Active DutySCRA  | Service Member on Active Duty                   |       | NUMBER    | 1      | Y        | SCRA_SVC_ACTIVE_SW       |
| Service Member Missing on DutySCRA | Service Member Missing on Duty                  |       | NUMBER    | 1      | Y        | SCRA_MEMBER_MISSING_FLG  |
| Service Member Active Dependent    | Service Member Active Dependent                 |       | NUMBER    | 1      | Y        | SCRA_DEP_ACTIVE_SW       |
| Service Member Updated Switch      | Service Member Updated Switch                   |       | CHAR      | 1      | N        | SCRA_UPDATE_SW           |

#### 4.1.2.3 Party Address Details

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

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

**Table 4–10 Party Address Details**

| Field Name | Description        | Value | Data Type | Length | Required | Column Name |
|------------|--------------------|-------|-----------|--------|----------|-------------|
| Party ID   | Party ID as stored |       | VARCHAR2  | 40     | Y        | HOST_CUST_  |

#### 4.1 Feeder Tables

| Field Name           | Description   | Value                            | Data Type | Length | Required | Column Name       |
|----------------------|---|----------------------------------|-----------|--------|----------|-------------------|
|                      | in Host   |                                  |           |        |          | NBR               |
| Host ID              | Source Host ID for host   |                                  | VARCHAR2  | 10     | Y        | SRC_HOST_ID       |
| Address Type         | Address Type Code maintained in Host  | Home, Business, Postal, Seasonal | VARCHAR2  | 20     | Y        | ADDR_TYPE_CD      |
| Sequence ID          | Sequence ID maintained in Host for each address type in case multiple addresses are maintained for same address type              |                                  | VARCHAR2  | 40     | Y        | FDR_ADDR_SEQ_ID   |
| Address 1            | Address Line 1  |                                  | VARCHAR2  | 120    | N        | 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     | N        | CITY_CD           |
| Country              | Country Code  |                                  | VARCHAR2  | 30     | N        | COUNTRY_CD        |
| Post/ Zip/ Pin Code  | Zip Code  |                                  | VARCHAR2  | 30     | N        | 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  | 60     | N        | STATUS            |
| Record Creation Date | Date on which data is fed to Collections  |                                  | DATE      | 10     | N        | CRET_DTTM         |
| Record Type          | Signifies if the data is created initially or is  | I - Insert<br>U - Update         | VARCHAR2  | 10     | N        | RCD_TYPE          |

| Field Name              | Description   | Value | Data Type | Length | Required | Column Name       |
|-------------------------|---|-------|-----------|--------|----------|-------------------|
|                         | update for existing data  |       |           |        |          |                   |
| Effective date          | Effective date  |       | DATE      | 10     | Y        | EFFECTIVE_DT      |
| State code              | State code  |       | VARCHAR2  | 60     | N        | FDR_STATE_CD      |
| Process Status          | To check the current status of the process. Default is P-Pending. |       | VARCHAR2  | 1      | N        | PROCESS_STATUS    |
| Message Category Number | Defined error message category                                    |       | NUMBER    | 5,0    | N        | MESSAGE_CAT_NBR   |
| Message Number          | Error message number  |       | NUMBER    | 5,0    | N        | MESSAGE_NBR       |
| Record Update Date      | Date on which the record is updated                               |       | DATE      | 10     | N        | RECORD_UPDATE_DT  |
| Record Exist Switch     | To check whether the record is available or not                   |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW  |
| Session Start Date      | Session Start Date  |       | VARCHAR2  | 4      | N        | SEASON_START_MMDD |
| Session End Date        | Session End Date  |       | VARCHAR2  | 4      | N        | SEASON_END_MMDD   |
| Address Status          | Address Status  |       | VARCHAR2  | 30     | Y        | ADDRESS_STATUS    |
| Address Id              | Address Id  |       | VARCHAR2  | 40     | N        | ADDRESS_ID        |

#### 4.1.2.4 Party Employment Details

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

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

**Table 4-11 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 |       | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE |

4.1 Feeder Tables

| Field Name        | Description  | Value   | Data Type | Length | Required | Column Name        |
|-------------------|--|---|-----------|--------|----------|--------------------|
|                   | of Party. This will depend on setups in host and is used in case of multi-branding features. |   |           |        |          |                    |
| Sequence ID       | Sequence ID of Employment details  |   | VARCHAR2  | 40     | Y        | FDR_EMP_SEQ_ID     |
| Employment Status | Employment Status Code   | <b>Employment Status:</b><br>For example:, Full Time, Part Time, Home Duties, Non-Resident, Pensioner, Retired, Student, Superannuation, Unemployed | VARCHAR2  | 4      | N        | EMPLOYMENT_STAT_CD |
| Employment Type   | Employment Type  | <b>Employment Type:</b><br>For example, Others, Salaried, Self Employed, Both- Salaried and Self Employed   | VARCHAR2  | 30     | N        | EMPLOYMENT_TYPE    |
| Employer Name     | Name of the employer of the customer   |   | VARCHAR2  | 120    | N        | EMPLOYER_NAME      |
| Industry Type     | Industry Type  |   | VARCHAR2  | 30     | N        | INDUSTRY_TYPE      |
| Company Type      |  | For example, Public Limited, Private Limited, Government Organization   | VARCHAR2  | 30     | N        | COMPANY_TYPE       |
| Occupation        | Occupation   |   | VARCHAR2  | 30     | N        | PROFESSION_TYPE    |
| Designation       | Designation  |   | VARCHAR2  | 120    | N        | DESIGNATION_TXT    |
| Gross             | Gross Annual Salary  |   | NUMBER    | 36,18  | N        | GRS_ANNUAL_INCOME  |

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
| Annual Salary           |   |                          |           |        |          |                  |
| Start Date              | Start Date  |                          | DATE      | 10     | N        | START_DT         |
| End Date                | End Date  |                          | DATE      | 10     | N        | END_DT           |
| Status                  | Status  |                          | VARCHAR2  | 60     | N        | STATUS           |
| Record Creation Date    | Date on which data is fed to Collections                                  |                          | DATE      | 10     | N        | 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         |
| Process Status          | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not                           |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.2.5 Party Identification Details

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

**Description:** This table holds party ID type details from host.

**Table 4–12 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 |



#### 4.1 Feeder Tables

| Field Name              | Description   | Value                                       | Data Type | Length | Required | Column Name           |
|-------------------------|---|---|-----------|--------|----------|-----------------------|
| Host ID                 | Source Host ID for host   |   | VARCHAR2  | 10     | Y        | SRC_HOST_ID           |
| Identification Type     | Value of Identification Type Code   | Passport No, Driving License No, and so on. | VARCHAR2  | 30     | Y        | FDR_ID_TYPE           |
| ID Value                | Identification Number corresponding to each of the identification types   |   | VARCHAR2  | 40     | N        | 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_DETERMINANT_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     | N        | 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              |
| Process Status          | Used to check current status of process. Default is P-Pending.  |   | VARCHAR2  | 1      | Y        | PROCESS_STATUS        |
| Message Category Number | Defined error message category  |   | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR       |
| Message Number          | Error message number  |   | NUMBER    | 5,0    | Y        | MESSAGE_NBR           |
| Record Update Date      | Date on which the record is updated   |   | DATE      | 10     | N        | RECORD_UPDATE_DT      |

| Field Name          | Description                                     | Value | Data Type | Length | Required | Column Name        |
|---------------------|---|-------|-----------|--------|----------|--------------------|
| Record Exist Switch | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW   |
| ID_TYPE_VAL_STATUS  | ID Type Status                                  |       | VARCHAR2  | 10     | N        | ID_TYPE_VAL_STATUS |

#### 4.1.2.6 Party Name Details

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

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

**Table 4–13 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 |       | VARCHAR2  | 60     | Y        | FDR_DETERMINANT_VALUE  |

| Field Name              | Description  | Value                    | Data Type | Length | Required | Column Name         |
|-------------------------|--|--------------------------|-----------|--------|----------|---------------------|
|                         | in host and is used in case of multi-branding features.                            |                          |           |        |          |                     |
| Primary Name Flag       | Signifies if a particular name needs to be used as a primary name for the customer | Y/N                      | CHAR      | 1      | N        | FDR_PRIMARY_NAME_SW |
| Record Creation Date    | Date on which data is fed to Collections   |                          | DATE      | 10     | N        | 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            |
| PER_NAME_STATUS         | Person name status   |                          | VARCHAR2  | 10     | N        | PER_NAME_STATUS     |
| Process Status          | To check the current status of the process. Default is P-Pending.                  |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS      |
| Message Category Number | Defined error message category   |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR     |
| Message Number          | Error message number   |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR         |
| Record Update Date      | Date on which the record is updated  |                          | DATE      | 10     | N        | RECORD_UPDATE_DT    |
| Record Exist Switch     | To check whether the record is available or not                                    |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW    |
| First name prefix       | First name prefix  |                          | VARCHAR2  | 120    | N        | FIRST_PREFIX_DESC   |
| Second name prefix      | Second name prefix   |                          | VARCHAR2  | 120    | N        | SECOND_PREFIX_DESC  |
| SUFFIX_DESC             | Suffix description   |                          | VARCHAR2  | 120    | N        | SUFFIX_DESC         |

#### 4.1.2.7 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 4–14 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              |   |  | 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    | N        | 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    | 10,0   | N        | WKDAY_FROM_TM      |

| Field Name              | Description   | Value  | Data Type | Length | Required | Column Name          |
|-------------------------|---|--|-----------|--------|----------|----------------------|
| Time To (weekdays)      | End Time for contacting on weekdays                                       | In hundred hour format (for example, 1800 for 6:00 PM) | NUMBER    | 10,0   | 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    | 10,0   | 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    | 10,0   | N        | WKEND_TO_TM          |
| Preference Frequency    | Preferred Frequency of contact  |  | NUMBER    | 20     | N        | PREFERENCE_FREQUENCY |
| Primary Contact Point   | Primary Contact Point Flag  |  | VARCHAR2  | 10     | N        | FDR_PRIMARY_SW       |
| Status                  | Status - if Active or Dormant   |  | VARCHAR2  | 60     | Y        | STATUS               |
| Record Creation Date    | Date on which data is fed to Collections                                  |  | DATE      | 10     | N        | CRET_DTTM            |
| Record Type             | Signifies if the data is created initially or is update for existing data | I - Insert<br>U - Update                               | VARCHAR2  | 10     | N        | RCD_TYPE             |
| Process Status          | To check the current status of the process. Default is P-Pending.         |  | VARCHAR2  | 1      | Y        | PROCESS_STATUS       |
| Message Category Number | Defined error message category  |  | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR      |
| Message                 | Error message   |  | NUMBER    | 5,0    | Y        | MESSAGE_NBR          |

| Field Name                       | Description                                     | Value | Data Type | Length | Required | Column Name                |
|----------------------------------|---|-------|-----------|--------|----------|----------------------------|
| Number                           | number  |       |           |        |          |                            |
| Record Update Date               | Date on which the record is updated             |       | DATE      | 10     | N        | RECORD_UPDATE_DT           |
| Record Exist Switch              | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW           |
| Do Not Disturb Flag              | Do Not Disturb Flag                             |       | NUMBER    | 1      | Y        | DND_FLG                    |
| DND Start Date                   | DND Start Date                                  |       | DATE      |        | N        | DND_START                  |
| DND End Date                     | DND End Date                                    |       | DATE      |        | N        | DND_END                    |
| Proffered Flag                   | Proffered Flag                                  |       | NUMBER    | 1      | Y        | IS_PREFERRED_FLAG          |
| Proffered Alert SMS              | Proffered Alert SMS                             |       | NUMBER    | 1      | Y        | IS_PREFERRED_FOR_ALERT_SMS |
| Permission to Call or Not        | Permission to Call or Not                       |       | NUMBER    | 1      | Y        | IS_PERMISSION_CALL         |
| Permission to Record Calls       | Permission to Record Calls                      |       | NUMBER    | 1      | Y        | IS_PERMISSION_RECORD_CALLS |
| Email Communication Consent Flag | Email Communication Consent Flag                |       | NUMBER    | 1      | Y        | ELEC_COMM_CONSENT          |
| Host Update Flag                 | Host Update Flag                                |       | NUMBER    | 1      | Y        | HOST_UPDATED_FLG           |
| Time Zone                        | Time Zone                                       |       | VARCHAR2  | 50     | N        | TIME_ZONE                  |
| Country Code                     | Country Code                                    |       | VARCHAR2  | 10     | N        | COUNTRY_CD                 |
| Phone Communication Consent      | Phone Communication Consent Flag                |       | NUMBER    | 1      | Y        | ELEC_COMM_CONSENT_PHONE    |

#### 4.1.2.8 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 4–15 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  | 50     | Y        | WARN_IND_CD      |
| Warning Indicator Value | Value of Warning Indicator Code   | Y/N                      | VARCHAR2  | 1      | N        | 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     | N        | 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         |
| Process Status          | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not                           |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.2.9 Service Member History Details

**Table Name:**Service Member History Details (CI\_FDR\_SCRA\_HIST\_DTLS)

**Description:** This table holds the Service Member History data from host.

**Table 4–16 Service Member History Details**

| Field Name                                | Description   | Value                    | Data Type | Length | Required | Column Name                 |
|---|---|--------------------------|-----------|--------|----------|-----------------------------|
| Source Host ID                            | Source Host ID  |                          | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID              |
| Party ID                                  | Party ID  |                          | VARCHAR2  | 40     | Y        | HOST_CUST_NBR               |
| Determinant Value                         | Determinant Value   |                          | VARCHAR2  | 50     | Y        | DETERMINANT_VALUE           |
| Service Member Order Number               | Service Member Order Number   |                          | VARCHAR2  | 50     | Y        | SCRA_ORDER_NUM              |
| Service Member Notification Date          | Service Member Notification Date  |                          | DATE      |        | Y        | SCRA_NOTIFICATION_DT        |
| Active Duty Start Date for Service Member | Active Duty Start Date for Service Member                                 |                          | DATE      |        | Y        | SCRA_START_DT_OF_ACTIVE_SVC |
| Unit name of Service Member               | Unit name of Service Member   |                          | VARCHAR2  | 50     | Y        | SCRA_UNIT_NAME              |
| End Date of Active Duty of Service Member | End Date of Active Duty of Service Member                                 |                          | DATE      |        | N        | SCRA_END_DT_OF_ACTIVE_SVC   |
| Service Member Order Status               | Service Member Order Status   |                          | VARCHAR2  | 50     | N        | SCRA_ORDER_STATUS           |
| Court Order Start Date                    | Court Order Start Date  |                          | DATE      |        | N        | COURT_ORDER_START_DATE      |
| Court Order End Date                      | Court Order End Date  |                          | DATE      |        | N        | COURT_ORDER_END_DATE        |
| Court Order Applicable Switch             | Court Order Applicable Switch   |                          | NUMBER    | 22     |          | IS_CURT_ORDER_APP_SW        |
| 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                    |
| Court Order Applicable Switch             | Court Order Applicable Switch   |                          | NUMBER    | 22     |          | IS_CURT_ORDER_APP_SW        |



| Field Name              | Description   | Value | Data Type | Length | Required | Column Name     |
|-------------------------|---|-------|-----------|--------|----------|-----------------|
| Process Status          | To check the current status of the process. Default is P-Pending. |       | VARCHAR2  | 1      | Y        | PROCESS_STATUS  |
| Message Category Number | Defined error message category                                    |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR |
| Message Number          | Error message number  |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR     |

### 4.1.3 Collateral Data

This section provides information on the tables related to collaterals.

#### 4.1.3.1 Collateral Details

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

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

**Table 4–17 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  | 50     | N        | COLLATERAL_TYPE      |
| Collateral Sub Type    | If there are any collateral sub type |       | VARCHAR2  | 50     | N        | COLLATERAL_SUB_TYPE  |
| Collateral Category    | Collateral Category                  |       | VARCHAR2  | 50     | N        | COLLATERAL_CAT       |
| Collateral Description | Collateral Description               |       | VARCHAR2  | 300    | N        | FDR_COLLATERAL_DESCR |
| Nature                 | Normal/ Guarantee                    |       | VARCHAR2  | 40     | N        | COLLATERAL_NATURE    |
| Collateral Currency    | Collateral Currency                  |       | VARCHAR2  | 3      | N        | COLLATERAL_CUR       |
| Assessed Value         | Market Value                         |       | NUMBER    | 36,18  | N        | ASSESD_VALUE         |
| Assessment Date        | Date of assessment                   |       | DATE      | 10     | N        | ASSESD_DT            |

| Field Name               | Description   | Value                            | Data Type | Length | Required | Column Name        |
|--------------------------|---|----------------------------------|-----------|--------|----------|--------------------|
| Bank Value               | Book Value  |                                  | NUMBER    | 36,18  | N        | 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  | 60     | 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             |
| 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     | N        | 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           |
| Process Status           | To check the current status of  |                                  | VARCHAR2  | 1      | Y        | PROCESS_STATUS     |

| Field Name              | Description                                     | Value | Data Type | Length | Required | Column Name       |
|-------------------------|---|-------|-----------|--------|----------|-------------------|
|                         | the process.<br>Default is P-<br>Pending.       |       |           |        |          |                   |
| Message Category Number | Defined error message category                  |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR   |
| Message Number          | Error message number                            |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR       |
| Record Update Date      | Date on which the record is updated             |       | DATE      | 10     | N        | RECORD_UPDATE_DT  |
| Record Exist Switch     | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW  |
| Realization ID          | Realization ID                                  |       | VARCHAR2  | 50     | N        | REALIZATION_ID    |
| Collateral State Code   | Collateral State Code                           |       | VARCHAR2  | 60     |          | FDR_COLL_STATE_CD |

#### 4.1.3.2 Collateral Charge Details

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

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

**Table 4–18 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     | N        | CRET_DTTM      |
| Record Type          | Signifies if the data is created initially or  | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE       |

| Field Name              | Description   | Value | Data Type | Length | Required | Column Name      |
|-------------------------|---|-------|-----------|--------|----------|------------------|
|                         | is update for existing data                                       |       |           |        |          |                  |
| Process Status          | To check the current status of the process. Default is P-Pending. |       | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category                                    |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                               |       | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not                   |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |
| Registration Number     | Registration Number   |       | VARCHAR2  | 20     | N        | CHARGE_REG_NUM   |
| Charge Status           | Charge Status   |       | VARCHAR2  | 60     | N        | CHARGE_STATUS    |

#### 4.1.3.3 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 4–19 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, FACILITY | VARCHAR2  | 10     | Y        | ENTITY_TYPE               |
| 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      | N        | FDR_LIMIT_CONTRIBUTION_SW |

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
| Record Creation Date    | Date on which data is fed to Collections                                  |                          | DATE      | 10     | N        | 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         |
| Process Status          | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not                           |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |
| Charge Code             | Charge Codes maintained in the host                                       |                          | VARCHAR2  | 20     | N        | CHRG_CD          |

#### 4.1.3.4 Collateral Guarantor Mapping

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

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

**Table 4–20 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     | N        | CRET_DTTM     |

| Field Name              | Description   | Value                    | Data Type | Length | Required | Column Name      |
|-------------------------|---|--------------------------|-----------|--------|----------|------------------|
| 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         |
| Process Status          | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS   |
| Message Category Number | Defined error message category  |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number  |                          | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                                       |                          | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not                           |                          | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.3.5 Collateral Owner Mapping

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

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

**Table 4–21 Collateral Owner 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 Customer mapped to collateral                                    |                          | VARCHAR2  | 40     | Y        | HOST_CUST_NBR     |
| Percentage of Ownership | Ownership Percentage of each of the Party                                    |                          | VARCHAR2  | 10     | N        | OWNERSHIP_PERCENT |
| Record Creation Date    | Date on which data is fed to Collections                                     |                          | DATE      | 10     | N        | CRET_DTTM         |
| Record Type             | Signifies if the data is created initially or is an update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE          |
| Process Status          | To check the current status of the process. Default is                       |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS    |

| Field Name              | Description                                     | Value | Data Type | Length | Required | Column Name      |
|-------------------------|---|-------|-----------|--------|----------|------------------|
|                         | P-Pending.                                      |       |           |        |          |                  |
| Message Category Number | Defined error message category                  |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number                            |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated             |       | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

#### 4.1.3.6 Collateral Vehicle Mapping

**Table Name:** Collateral Vehicle Mapping (CI\_FDR\_COLLATERAL\_AUTOMOBILE)

**Description:** This table holds Vehicle information for the collateral.

**Table 4–22 Collateral Vehicle Mapping**

| Field Name                    | Description  | Value                    | Data Type | Length | Required | Column Name       |
|-------------------------------|--|--------------------------|-----------|--------|----------|-------------------|
| Vehicle Identification Number | Vehicle Identification Number  |                          | VARCHAR2  | 30     | Y        | VHCL_IDENTIFY_NO  |
| Vehicle Make                  | Vehicle Make   |                          | VARCHAR2  | 20     | N        | VHCL_MAKE         |
| Vehicle model                 | Vehicle model  |                          | VARCHAR2  | 20     | N        | VHCL_MODEL        |
| Vehicle Trim                  | Vehicle Trim   |                          | VARCHAR2  | 20     | N        | VHCL_TRIM         |
| Manufacturing Year            | Manufacturing Year   |                          | VARCHAR2  | 5      | N        | MANUFACTUR_YEAR   |
| Licence Plate Zip Code        | Licence Plate Zip Code   |                          | VARCHAR2  | 10     | N        | LICNC_PLT_ZIPCODE |
| Licence Plate Number          | Licence Plate Number   |                          | VARCHAR2  | 20     | N        | LICNC_PLT_NUMBER  |
| Licence Plate State           | Licence Plate State  |                          | VARCHAR2  | 20     | N        | LICNC_PLT_STATE   |
| Source Host Id                | Source Host Id   |                          | VARCHAR2  | 10     | Y        | SRC_HOST_ID       |
| Record Creation Date          | Record Creation Date   |                          | DATE      |        | N        | CRET_DTTM         |
| Record Type                   | Signifies if the data is created initially or is an update for existing data | I - Insert<br>U - Update | VARCHAR2  | 10     | Y        | RCD_TYPE          |
| Process                       | To check the   |                          | VARCHAR2  | 1      | Y        | PROCESS_          |

| Field Name              | Description  | Value | Data Type | Length | Required | Column Name      |
|-------------------------|--|-------|-----------|--------|----------|------------------|
| Status                  | current status of the process. Default is P-Pending. |       |           |        |          | STATUS           |
| Message Category Number | Defined error message category                       |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR  |
| Message Number          | Error message number                                 |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date      | Date on which the record is updated                  |       | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch     | To check whether the record is available or not      |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |
| Collateral Code         | Collateral Code                                      |       | VARCHAR2  | 40     | Y        | COLLATERAL_CD    |

## 4.1.4 Insurance Data

This section provides information on the tables related to insurance.

### 4.1.4.1 Insurance Details

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

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

**Table 4–23 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. Possible Values | COLLATERAL, PERSON, or 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  | 50     | Y        | POLICY_NUM                |
| Insurance Policy Name | Insurance Policy Name                                  |                                 | VARCHAR2  | 100    | N        | FDR_INSURANCE_POLICY_NAME |



#### 4.1 Feeder Tables

| Field Name                  | Description   | Value                    | Data Type | Length | Required | Column Name             |
|-----------------------------|---|--------------------------|-----------|--------|----------|-------------------------|
| Insured Currency            | Currency Code of the Insured Amount                                       |                          | VARCHAR2  | 3      | N        | INSURED_CURR            |
| Insured Amount              | Insured Amount  |                          | NUMBER    | 36,18  | N        | INSURED_AMT             |
| Insurer Code                | Insurer Code as stored in host  |                          | VARCHAR2  | 50     | N        | INSURER_CD              |
| Insurer Name                | Insurer Name as stored in host  |                          | VARCHAR2  | 64     | N        | INSURER_NAME            |
| Policy Start Date           | Start date of Policy  |                          | DATE      | 10     | N        | POLICY_START_DT         |
| Policy End Date             | End date of Policy  |                          | DATE      | 10     | N        | POLICY_END_DT           |
| Premium Amount              | Insurance Premium   |                          | NUMBER    | 36,18  | N        | PREMIUM_AMT             |
| 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     | N        | 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                |
| DUA Switch                  | A DUA Switch applicable for LMI Insurance                                 |                          | VARCHAR2  | 1      | N        | DUA_APPLICABLE          |
| Net borrower premium amount | Net borrower premium amount   |                          | NUMBER    | 36,18  | N        | NET_BORR_PREMIUM_AMOUNT |
| Party ID                    | Party ID  |                          | VARCHAR2  | 40     | Y        | FDR_PARTY_ID            |
| Process Status              | To check the current status of the process. Default is P-Pending.         |                          | VARCHAR2  | 1      | Y        | PROCESS_STATUS          |
| Message                     | Defined error   |                          | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR         |

| Field Name          | Description                                     | Value | Data Type | Length | Required | Column Name      |
|---------------------|---|-------|-----------|--------|----------|------------------|
| Category Number     | message category                                |       |           |        |          |                  |
| Message Number      | Error message number                            |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR      |
| Record Update Date  | Date on which the record is updated             |       | DATE      | 10     | N        | RECORD_UPDATE_DT |
| Record Exist Switch | To check whether the record is available or not |       | VARCHAR2  | 1      | Y        | RECORD_EXISTS_SW |

### 4.1.5 Payment Data

This section provides information on the tables related to payments.

#### 4.1.5.1 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 4–24 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     | N        | FDR_TRANSACTION_DT      |
| Transaction Time             | Time for Transaction   |       | DATE      | 10     | N        | FDR_TRANSACTION_TM      |
| Value Date                   | Value Date on which the transaction was posted in the host   |       | DATE      | 10     | N        | FDR_VALUE_DT            |
| Transaction Currency         | Currency code of the transaction                             |       | VARCHAR2  | 3      | N        | FDR_TRANSACTION_CURR_CD |

| Field Name                           | Description  | Value | Data Type | Length | Required | Column Name              |
|--------------------------------------|--|-------|-----------|--------|----------|--------------------------|
| Transaction Amount                   | Payment Amount   |       | NUMBER    | 36,18  | N        | FDR_TRANSACTION_AMT      |
| Account Currency                     | Account Currency Code  |       | VARCHAR2  | 3      | N        | FDR_ACCT_CURR_CD         |
| Account Balance                      | Account Balance after Payment  |       | NUMBER    | 36,18  | N        | FDR_ACCT_AMT             |
| Transaction Code                     | Transaction Code as captured in the host   |       | VARCHAR2  | 30     | 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     | N        | CRET_DTTM                |
| 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 |
| Process Status                       | To check the current status of the process. Default is P-Pending.                              |       | VARCHAR2  | 1      | Y        | PROCESS_STATUS           |
| Message Category Number              | Defined error message category   |       | NUMBER    | 5,0    | Y        | MESSAGE_CAT_NBR          |
| Message Number                       | Error message number   |       | NUMBER    | 5,0    | Y        | MESSAGE_NBR              |

### 4.1.6 IRS Reporting

This section provides information on the tables related to IRS reporting.

**Table Name:** IRS Report Feeder Table (CI\_FDR\_IRS\_REPORT)

**Description:** This table holds IRS Report data.

**Table 4–25 IRS Report Data**

| Field Name        | Description              | Value | Data Type | Length  | Required | Column Name       |
|-------------------|--------------------------|-------|-----------|---------|----------|-------------------|
| DEBTOR_ACCNT_NBR  | Debtor Account Number    |       | VARCHAR2  | 40      | Y        | DEBTOR_ACCNT_NBR  |
| EVT_DT            | Event Date               |       | DATE      | -       | Y        | EVT_DT            |
| PRINCIPAL_AMT     | Principal Amount         |       | NUMBER    | (36,18) | Y        | PRINCIPAL_AMT     |
| INTEREST          | Interest Amount          |       | NUMBER    | (36,18) | N        | INTEREST          |
| FEE_CHARGES       | Fee Charge Amount        |       | NUMBER    | (36,18) | N        | FEE_CHARGES       |
| INSURANCE         | Insurance Amount         |       | NUMBER    | (36,18) | N        | INSURANCE         |
| EXPENSES          | Expense Amount           |       | NUMBER    | (36,18) | N        | EXPENSES          |
| RECOVERY_INTEREST | Recovery Interest Amount |       | NUMBER    | 36,18   | N        | RECOVERY_INTEREST |
| EVENT_CODE        | Event Code               |       | VARCHAR2  | 1       | Y        | EVENT_CODE        |
| MKT_VAL_PROPERTY  | Market Value Property    |       | NUMBER    | 36,18   | N        | MKT_VAL_PROPERTY  |

## 4.2 Interfacing Tables

This section provides details about the Interfacing tables.

### 4.2.1 Agency or Vendor Upload

This section provides interfacing tables related to Agency or Vendor Upload (C1-VNDUP).

#### 4.2.1.1 Upload Followup Table Details

**Table Name:** Vendor/Agency Upload Follow up Table (CI\_VNDR\_UPLD\_FOLLOWUP)

**Description:** This table holds Follow up Upload data.

**Table 4–26 Upload Followup Table**

| Field Name             | Description               | Value | Data Type | Length | Required | Column Name        |
|------------------------|---------------------------|-------|-----------|--------|----------|--------------------|
| Vendor Upload Followup | Vendor Upload Followup Id |       | CHAR      | 10     | Y        | VNDR_UPLD_FLWUP_ID |
| Vendor ID              | Vendor ID                 |       | CHAR      | 10     | Y        | VENDOR_ID          |
| Account Number         | Account Number            |       | VARCHAR2  | 40     | N        | HOST_ACCT_NBR      |

## 4.2 Interfacing Tables

| Field Name               | Description             | Value | Data Type | Length | Required | Column Name     |
|--------------------------|-------------------------|-------|-----------|--------|----------|-----------------|
| Case ID                  | Case ID                 |       | CHAR      | 10     | Y        | CASE_ID         |
| Customer Number          | Customer Number         |       | VARCHAR2  | 40     | N        | HOST_CUST_NBR   |
| Source Host ID           | Source Host ID          |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID     |
| Account Type Code        | Account Type Code       |       | CHAR      | 12     | Y        | ACTION_TYPE_CD  |
| Action Date              | Action Date             |       | DATE      |        | Y        | ACTION_DTTM     |
| Collector Comments       | Collector Comments      |       | VARCHAR2  | 2000   | N        | COL_COMMENTS    |
| Next Action Type Code    | Next Action Type Code   |       | CHAR      | 12     | N        | NXT_ACTN_TYP_CD |
| Next Action Date         | Next Action Date        |       | DATE      |        | N        | NXT_ACTN_DTTM   |
| User ID                  | User ID                 |       | CHAR      | 255    | Y        | USER_ID         |
| 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            |
| UDF10                    | User Defined Fields     |       | VARCHAR2  | 60     | N        | UDF10           |
| User Defined Field 1Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_1      |
| User Defined Field 2Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_2      |
| User Defined Field 3Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_3      |
| User Defined Field 4Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_4      |
| User Defined Field 5Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_5      |

| Field Name              | Description             | Value | Data Type | Length | Required | Column Name     |
|-------------------------|-------------------------|-------|-----------|--------|----------|-----------------|
| User Defined Flag 1     | User Defined Flag       |       | CHAR      | 1      | N        | UDF_FLAG1       |
| User Defined Flag 2     | User Defined Flag       |       | CHAR      | 1      | N        | UDF_FLAG2       |
| User Defined Flag 3     | User Defined Flag       |       | CHAR      | 1      | N        | UDF_FLAG3       |
| User Defined Flag 4     | User Defined Flag       |       | CHAR      | 1      | N        | UDF_FLAG4       |
| User Defined Flag 5     | User Defined Flag       |       | CHAR      | 1      | N        | UDF_FLAG5       |
| Process Status          | Process Status          |       | VARCHAR2  | 1      | N        | PROCESS_STATUS  |
| Message Category Number | Message Category Number |       | NUMBER    | 5      | N        | MESSAGE_CAT_NBR |
| Message Number          | Message Number          |       | NUMBER    | 5      | N        | MESSAGE_NBR     |
| Batch Run Date          | Batch Run Date          |       | DATE      |        | N        | BATCH_RUN_DTTM  |

#### 4.2.1.2 Upload Result Table Details

**Table Name:** Vendor/Agency Upload Result Table (CI\_VNDR\_UPLD\_RESULT)

**Description:** This table holds Result Upload data.

**Table 4–27 Upload Result Table**

| Field Name                | Description               | Value | Data Type | Length | Required | Column Name         |
|---------------------------|---------------------------|-------|-----------|--------|----------|---------------------|
| Vendor Upload Result ID   | Vendor Upload Result ID   |       | CHAR      | 10     | Y        | VNDR_UPLD_RESULT_ID |
| Vendor Upload Followup ID | Vendor Upload Followup ID |       | CHAR      | 10     | Y        | VNDR_UPLD_FLWUP_ID  |
| Result Type Code          | Result Type Code          |       | CHAR      | 12     | Y        | RESULT_TYPE_CD      |
| Collector Comments        | Collector Comments        |       | VARCHAR2  | 2000   | N        | RESULT_COL_COMMENTS |
| Result Date               | Result Date               |       | DATE      |        | Y        | RESULT_DTTM         |
| Primary Result Switch     | Primary Result Switch     |       | CHAR      | 1      | Y        | PRIM_RESULT_SW      |

### 4.2.1.3 Upload PTP Table Details

**Table Name:** Vendor/Agency Upload Promise To Pay(PTP) Table (CI\_VNDR\_UPLD\_PTP)

**Description:** This table holds PTP Upload data.

**Table 4–28 PTP Upload data**

| Field Name              | Description             | Value | Data Type | Length | Required | Column Name        |
|-------------------------|-------------------------|-------|-----------|--------|----------|--------------------|
| Vendor Upload Id        | Vendor Upload Id        |       | CHAR      | 10     | Y        | VNDR_UPLD_FLWUP_ID |
| Vendor Id               | Vendor Id               |       | CHAR      | 10     | Y        | VENDOR_ID          |
| Account Number          | Account Number          |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR      |
| Case ID                 | Case ID                 |       | CHAR      | 10     | Y        | CASE_ID            |
| Source Host ID          | Source Host ID          |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID        |
| PTP Type Code           | PTP Type Code           |       | CHAR      | 12     | Y        | PP_TYPE_CD         |
| Pay Method Code         | Pay Method Code         |       | CHAR      | 10     | Y        | PAY_METH_CD        |
| Record Creation Date    | Record Creation Date    |       | DATE      |        | N        | CRE_DTTM           |
| PTP Start Date          | PTP Start Date          |       | DATE      |        | Y        | PTP_START_DT       |
| User ID                 | User ID                 |       | CHAR      | 255    | Y        | USER_ID            |
| 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               |
| UDF10                   | User Defined Fields     |       | VARCHAR2  | 60     | N        | UDF10              |
| User Defined Field Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_1         |
| User Defined Field Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_2         |

| Field Name              | Description             | Value | Data Type | Length | Required | Column Name     |
|-------------------------|-------------------------|-------|-----------|--------|----------|-----------------|
| User Defined Field Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_3      |
| User Defined Field Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_4      |
| User Defined Field Date | User Defined Field Date |       | DATE      |        | N        | UDF_DTTM_5      |
| User Defined Flag       | User Defined Field Date |       | CHAR      | 1      | N        | UDF_FLAG1       |
| User Defined Flag       | User Defined Field Date |       | CHAR      | 1      | N        | UDF_FLAG2       |
| User Defined Flag       | User Defined Field Date |       | CHAR      | 1      | N        | UDF_FLAG3       |
| User Defined Flag       | User Defined Field Date |       | CHAR      | 1      | N        | UDF_FLAG4       |
| User Defined Flag       | User Defined Field Date |       | CHAR      | 1      | N        | UDF_FLAG5       |
| Process Status          | Process Status          |       | VARCHAR2  | 1      | N        | PROCESS_STATUS  |
| Message Category Number | Message Category Number |       | NUMBER    | 5      | N        | MESSAGE_CAT_NBR |
| Message Number          | Message Number          |       | NUMBER    | 5      | N        | MESSAGE_NBR     |
| Batch Run Date          | Batch Run Date          |       | DATE      |        | N        | BATCH_RUN_DTTM  |

#### 4.2.1.4 Upload PTP Schedule Table Details

**Table Name:** Vendor/Agency Upload Promise To Pay(PTP) Schedule Table (CI\_VNDR\_UPLD\_PTP\_SCHED)

**Description:** This table holds PTP Schedule Upload data.

**Table 4–29 PTP Schedule Upload data**

| Field Name                    | Description                   | Value | Data Type | Length | Required | Column Name            |
|-------------------------------|-------------------------------|-------|-----------|--------|----------|------------------------|
| Vendor Upload PTP Schedule ID | Vendor Upload PTP Schedule ID |       |           | 10     | Y        | VNDR_UPLD_PTP_SCHED_ID |
| Vendor Upload ID              | Vendor Upload ID              |       | CHAR      | 10     | Y        | VNDR_UPLD_PTP_ID       |
| PTP                           | PTP Schedule Date             |       | DATE      |        | Y        | PP_SCHED_              |



## 4.2 Interfacing Tables

| Field Name          | Description         | Value | Data Type | Length | Required | Column Name  |
|---------------------|---------------------|-------|-----------|--------|----------|--------------|
| Schedule Date       |                     |       |           |        |          | DT           |
| PTP Schedule Amount | PTP Schedule Amount |       | NUMBER    | 36,18  | Y        | PP_SCHED_AMT |
| Currency Code       | Currency Code       |       | CHAR      | 3      | Y        | CURRENCY_CD  |
| Pay Clear ID        | Pay Clear ID        |       | CHAR      | 12     | N        | APAY_CLR_ID  |

### 4.2.1.5 Upload Settlement Table Details

**Table Name:** Vendor/Agency Upload Settlement Table (CI\_VNDR\_UPLD\_SETTLEMENT)

**Description:** This table holds Settlement Upload data.

**Table 4–30 Settlement Table data**

| Field Name              | Description             | Value | Data Type | Length | Required | Column Name     |
|-------------------------|-------------------------|-------|-----------|--------|----------|-----------------|
| Vendor ID               | Vendor ID               |       | CHAR      | 10     | Y        | VENDOR_ID       |
| Account Number          | Account Number          |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR   |
| Source Host ID          | Source Host ID          |       | VARCHAR2  | 10     | Y        | SRC_HOST_ID     |
| Case ID                 | Case ID                 |       | CHAR      | 10     | Y        | CASE_ID         |
| User ID                 | User ID                 |       | CHAR      | 255    | N        | USER_ID         |
| Process Status          | Process Status          |       | VARCHAR2  | 1      | N        | PROCESS_STATUS  |
| Message Category Number | Message Category Number |       | NUMBER    | 5      | N        | MESSAGE_CAT_NBR |
| Message Number          | Message Number          |       | NUMBER    | 5      | N        | MESSAGE_NBR     |
| Batch Run Date          | Batch Run Date          |       | DATE      |        | N        | BATCH_RUN_DTTM  |
| Settlement Amount       | Settlement Amount       |       | NUMBER    | 36,18  | Y        | SETTLEMENT_AMT  |
| Settlement Term         | Settlement Term         |       | NUMBER    | 3      | Y        | SETTLEMENT_TERM |
| Comments                | Comments                |       | VARCHAR2  | 4000   | N        | COMMENTS        |

### 4.2.2 Dialer Results Upload

This section provides interfacing tables related to Dialer Result Upload (C1-DLRRS).

### 4.2.2.1 Upload Dialer Result Table Details

**Table Name:** Dialer Result Upload Batch (CI\_DIALER\_RESULTS\_UPLOAD)

**Description:** This table holds Dialer Result Upload data.

**Table 4–31 Dialer Result Upload data**

| Field Name                 | Description                | Value | Data Type | Length | Required | Column Name         |
|----------------------------|----------------------------|-------|-----------|--------|----------|---------------------|
| Stagging ID                | Stagging ID                |       | CHAR      | 10     | Y        | STAGING_ID          |
| Dialer Extract ID          | Dialer Extract ID          |       | CHAR      | 10     | Y        | DIALER_EXTRACT_ID   |
| Channel Type               | Channel Type               |       | CHAR2     | 4      | Y        | CHANNEL_TYPE        |
| Account Number             | Account Number             |       | VARCHAR2  | 40     | N        | HOST_ACCNT_NBR      |
| Customer Number            | Customer Number            |       | VARCHAR2  | 40     | Y        | HOST_CUST_NBR       |
| Source Host ID             | Source Host ID             |       | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID      |
| Termination code           | Termination code           |       | CHAR      | 12     | Y        | TERMINATION_CD      |
| Status Code                | Status Code                |       | CHAR      | 12     | Y        | STATUS_CD           |
| Enterprise Customer Number | Enterprise Customer Number |       | VARCHAR2  | 60     | N        | ENTERPRISE_CUST_NBR |
| Attempts                   | Attempts                   |       | NUMBER    | 5      | Y        | ATTEMPTS            |
| Call Date                  | Call Date                  |       | DATE      |        | N        | CALL_DTTM           |
| Campaign ID                | Campaign ID                |       | CHAR      | 10     | Y        | CAMPAIGN_ID         |
| Dialer Contact ID          | Dialer Contact ID          |       | CHAR      | 10     | Y        | DIALER_CONTACT_ID   |
| Contact Number             | Contact Number             |       | NUMBER    | 20     | N        | CONTACT_NUMBER      |
| Record Creation Date       | Record Creation Date       |       | DATE      |        | N        | CRE_DTTM            |
| UDF1                       | User Defined Fields        |       | VARCHAR2  | 60     | Y        | UDF1                |
| UDF2                       | User Defined Fields        |       | VARCHAR2  | 60     | Y        | UDF2                |
| UDF3                       | User Defined Fields        |       | VARCHAR2  | 60     | Y        | UDF3                |
| UDF4                       | User Defined Fields        |       | VARCHAR2  | 60     | Y        | UDF4                |

## 4.2 Interfacing Tables

| Field Name | Description         | Value | Data Type | Length | Required | Column Name |
|------------|---------------------|-------|-----------|--------|----------|-------------|
| UDF5       | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF5        |
| UDF6       | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF6        |
| UDF7       | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF7        |
| UDF8       | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF8        |
| UDF9       | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF9        |
| UDF10      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF10       |
| UDF11      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF11       |
| UDF12      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF12       |
| UDF13      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF13       |
| UDF14      | User Defined Fields |       | VARCHAR2  | 60     | N        | UDF14       |
| UDF15      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF15       |
| UDF16      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF16       |
| UDF17      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF17       |
| UDF18      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF18       |
| UDF19      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF19       |
| UDF20      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF20       |
| UDF21      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF21       |
| UDF22      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF22       |
| UDF23      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF23       |
| UDF24      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF24       |
| UDF25      | User Defined        |       | VARCHAR2  | 60     | Y        | UDF25       |

| Field Name | Description         | Value | Data Type | Length | Required | Column Name |
|------------|---------------------|-------|-----------|--------|----------|-------------|
|            | Fields              |       |           |        |          |             |
| UDF26      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF26       |
| UDF27      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF27       |
| UDF28      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF28       |
| UDF29      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF29       |
| UDF30      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF30       |
| UDF31      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF31       |
| UDF32      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF32       |
| UDF33      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF33       |
| UDF34      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF34       |
| UDF35      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF35       |
| UDF36      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF36       |
| UDF37      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF37       |
| UDF38      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF38       |
| UDF39      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF39       |
| UDF40      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF40       |
| UDF41      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF41       |
| UDF42      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF42       |
| UDF43      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF43       |
| UDF44      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF44       |
| UDF45      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF45       |

| Field Name | Description         | Value | Data Type | Length | Required | Column Name |
|------------|---------------------|-------|-----------|--------|----------|-------------|
| UDF46      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF46       |
| UDF47      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF47       |
| UDF48      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF48       |
| UDF49      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF49       |
| UDF50      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF50       |
| UDF51      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF51       |
| UDF52      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF52       |
| UDF53      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF53       |
| UDF54      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF54       |
| UDF55      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF55       |
| UDF56      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF56       |
| UDF57      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF57       |
| UDF58      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF58       |
| UDF59      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF59       |
| UDF60      | User Defined Fields |       | VARCHAR2  | 60     | Y        | UDF60       |
| VERSION    | Version             |       | NUMBER    | 5      | Y        | VERSION     |

### 4.2.3 Account Dialer Extract

This section provides interfacing tables related to Account Dialer Extract (C1-DLEXT).

#### 4.2.3.1 Dialer Extract Account Table Details

**Table Name:** Account Dialer Extract Table (CI\_ACCT\_DILR\_EXTRCT)

**Description:** This table holds Account Dialer Extract data.

Table 4–32 Account Dialer Extract data.

| Field Name                      | Description                     | Value | Data Type | Length | Required | Column Name              |
|---------------------------------|---------------------------------|-------|-----------|--------|----------|--------------------------|
| Customer Number                 | Customer Number                 |       | VARCHAR2  | 40     | Y        | HOST_CUST_NBR            |
| Account Number                  | Account Number                  |       | VARCHAR2  | 40     | Y        | HOST_ACCT_NBR            |
| Source Host ID                  | Source Host ID                  |       | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID           |
| Case ID                         | Case ID                         |       | CHAR      | 10     | Y        | CASE_ID                  |
| Account Relation Type Code      | Account Relation Type Code      |       | VARCHAR2  | 10     | Y        | ACCT_REL_TYPE_CD         |
| Dialer Extract Channel Type     | Dialer Extract Channel Type     |       | VARCHAR2  | 40     | Y        | DILREXTCT_CHANNEL_TYPE   |
| Dialer Extract filter ID        | Dialer Extract filter ID        |       | VARCHAR2  | 40     | Y        | DILREXTCT_FILTER_ID      |
| Campaign Description            | Campaign Description            |       | VARCHAR2  | 40     | N        | CAMPAIGN_DESCR           |
| Campaign Priority               | Campaign Priority               |       | VARCHAR2  | 40     | N        | CAMPAIGN_PRIORITY        |
| Dialer Extract Status           | Dialer Extract Status           |       | VARCHAR2  | 20     | N        | DILREXTCT_STATUS         |
| Dialer Extract Termination Code | Dialer Extract Termination Code |       | VARCHAR2  | 20     | N        | DILREXTCT_TERMINATION_CD |
| Exclude Reason Code             | Exclude Reason Code             |       | VARCHAR2  | 20     | N        | EXCLUDE_REASON_CODE      |
| Extract Date                    | Extract Date                    |       | DATE      |        | N        | EXTRACT_DTTM             |
| Next Display Date               | Next Display Date               |       | DATE      |        | N        | NEXT_DISPLAY_DATE        |
| Dialer Extract File Name        | Dialer Extract File Name        |       | VARCHAR2  | 400    | N        | DILREXTCT_FILE_NAME      |
| Queue Code                      | Queue Code                      |       | CHAR      | 10     | N        | QUEUE_CD                 |
| Host Product Group Code         | Host Product Group Code         |       | VARCHAR2  | 30     | N        | HOST_PROD_GRP_CD         |
| Host Product Code               | Host Product Code               |       | VARCHAR2  | 30     | N        | HOST_PRD_CD              |
| Overdue Amount                  | Overdue Amount                  |       | NUMBER    | 36,18  | Y        | OVERDUE_AMT              |
| Outstanding                     | Outstanding                     |       | NUMBER    | 36,18  | Y        | OUTSTANDING_             |

## 4.2 Interfacing Tables

| Field Name           | Description          | Value | Data Type | Length | Required | Column Name      |
|----------------------|----------------------|-------|-----------|--------|----------|------------------|
| Amount               | Amount               |       |           |        |          | AMT              |
| Days Past Due        | Days Past Due        |       | NUMBER    | 4      | Y        | DAYS_PAST_DUE    |
| Record Creation Date | Record Creation Date |       | DATE      |        | Y        | CREATION_DTTM    |
| Version              | Version              |       | NUMBER    | 5      | Y        | VERSION          |
| Suspended Switch     | Suspended Switch     |       | CHAR      | 1      | Y        | SUSPEND_SW       |
| Full Name            | Full Name            |       | VARCHAR2  | 400    | N        | FULL_NAME        |
| Customer Prefix      | Customer Prefix      |       | VARCHAR2  | 40     | N        | CUST_PREFIX      |
| First Name           | First Name           |       | VARCHAR2  | 200    | N        | FIRST_NAME       |
| Last Name            | Last Name            |       | VARCHAR2  | 200    | N        | LAST_NAME        |
| Customer Suffix      | Customer Suffix      |       | VARCHAR2  | 40     | N        | CUST_SUFFIX      |
| Address Type Code    | Address Type Code    |       | VARCHAR2  | 20     | N        | ADDR_TYPE_CD     |
| Address Line1        | Address Line1        |       | VARCHAR2  | 400    | N        | ADDRESS_LN1      |
| Address Line2        | Address Line2        |       | VARCHAR2  | 400    | N        | ADDRESS_LN2      |
| Address Line3        | Address Line3        |       | VARCHAR2  | 400    | N        | ADDRESS_LN3      |
| Address Line4        | Address Line4        |       | VARCHAR2  | 400    | N        | ADDRESS_LN4      |
| City                 | City                 |       | VARCHAR2  | 40     | N        | CITY             |
| Country              | Country              |       | VARCHAR2  | 40     | N        | COUNTRY          |
| State                | State                |       | VARCHAR2  | 40     | N        | STATE            |
| Postal Code          | Postal Code          |       | VARCHAR2  | 40     | N        | POSTAL           |
| Birthdate            | Birthdate            |       | DATE      |        | N        | BIRTH_DT         |
| Next Action Time     | Next Action Time     |       | VARCHAR2  | 8      | N        | NEXT_ACTION_TIME |

### 4.2.3.2 Dialer Extract Contact Table Details

**Table Name:** Dialer Extract Contact Table (CI\_DIALER\_EXTRACTS\_CONTACT)

**Description:** This table holds Dialer Extract Contact data.

**Table 4–33 Dialer Extract Contact data.**

| Field Name                     | Description                    | Value | Data Type | Length | Required | Column Name                   |
|--------------------------------|--------------------------------|-------|-----------|--------|----------|-------------------------------|
| Customer Number                | Customer Number                |       | VARCHAR2  | 40     | Y        | HOST_CUST_NBR                 |
| Source Host ID                 | Source Host ID                 |       | VARCHAR2  | 10     | Y        | SOURCE_HOST_ID                |
| Determinant Value              | Determinant Value              |       | VARCHAR2  | 60     | Y        | DETERMINANT_VALUE             |
| Contact Point Type             | Contact Point Type             |       | VARCHAR2  | 10     | Y        | CONTACT_POINT_TYPE            |
| Contact Value                  | Contact Value                  |       | VARCHAR2  | 400    | Y        | CONTACT_VALUE                 |
| Contact Preferred Type         | Contact Preferred Type         |       | VARCHAR2  | 10     | Y        | CONTACT_PREF_TYPE             |
| Weekday From Time              | Weekday From Time              |       | NUMBER    | 4      | N        | WKDAY_FROM_TM                 |
| Weekday To Time                | Weekday To Time                |       | NUMBER    | 4      | N        | WKDAY_TO_TM                   |
| Weekend From Time              | Weekend From Time              |       | NUMBER    | 4      | N        | WKEND_FROM_TM                 |
| Weekend To Time                | Weekend To Time                |       | NUMBER    | 4      | N        | WKEND_TO_TM                   |
| Do Not Disturb Start Date      | Do Not Disturb Start Date      |       | DATE      |        | N        | DND_START                     |
| Do Not Disturb End Date        | Do Not Disturb End Date        |       | DATE      |        | N        | DND_END                       |
| Time Zone                      | Time Zone                      |       | VARCHAR2  | 50     | N        | TIME_ZONE                     |
| Acceptance Start Date          | Acceptance Start Date          |       | DATE      |        | N        | ACCEPT_START_DTTM             |
| Acceptance End Date            | Acceptance End Date            |       | DATE      |        | N        | ACCEPT_END_DTTM               |
| Do Not Disturb Flag            | Do Not Disturb Flag            |       | CHAR      | 1      | N        | DND_FLAG                      |
| Preferred Contact Switch       | Preferred Contact Switch       |       | CHAR      | 1      | N        | IS_PREFERRED_SW               |
| Preferred For SMS Alert Switch | Preferred For SMS Alert Switch |       | CHAR      | 1      | N        | IS_PREFERRED_FOR_ALERT_SMS_SW |
| Preferred for Call Switch      | Preferred for Call Switch      |       | CHAR      | 1      | N        | IS_PERMISSION_CALL_SW         |



| Field Name                              | Description                             | Value | Data Type | Length | Required | Column Name                   |
|---|---|-------|-----------|--------|----------|-------------------------------|
| Preferred for Email Switch              | Preferred for Email Switch              |       | CHAR      | 1      | N        | IS_PREFERRED_EMAIL_SW         |
| Switch for permission to record calls   | Switch for permission to record calls   |       | CHAR      | 1      | N        | IS_PERMISSION_RECORD_CALLS_SW |
| Electronic Communication Consent Switch | Electronic Communication Consent Switch |       | CHAR      | 1      | N        | ELEC_COMM_CONSENT_SW          |

## 4.3 OBP Views

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

### 4.3.1 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\_COL\_ACCT\_WARN\_IND\_DATA\_XF
- FLX\_DD\_COL\_BILL\_DATA\_XF\_VW
- FLX\_AC\_COL\_FD\_ACCT\_PAY\_TRACK\_VW

### 4.3.2 Account Updateable Views

The account updateable views are as follows:

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

### 4.3.3 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

### 4.3.4 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

### 4.3.5 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
- FLX\_LM\_COL\_FD\_COL\_ADDR\_VW
- FLX\_LM\_COL\_FD\_COL\_AUTO\_VW

## 4.4 OBP Collections Views

Collections system uses view 'CI\_SETTLEMENT\_OFFER\_VNDR\_VW' which includes the Settlement Offer related details created by the vendor, the Vendor Id and the full name of the Vendor.



# 5 Algorithms

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

## 5.1 Stop Contract: C1-CURENTITY

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

**Table 5–1 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. |

## 5.2 Cure Account: C1-FINCOLL

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

**Table 5–2 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> | <p>This algorithm performs the following activities:</p> <ul style="list-style-type: none"> <li>- Invoke OBP service to set the incollection flag in host as "N".</li> <li>- Mark incollection flag as "N" in collections.</li> <li>- Set end date in CI_PARTY_COLLECT as posting date.</li> <li>- Update number of times account is self cured (used for statistics).</li> <li>- Remove strategy review date.</li> </ul> |
| <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  |
| <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</p>   |

|                        |   |
|------------------------|---|
|                        | contact methods. For example, SMS, EM, and so on.<br>This value is used to calculate the number of self cured statistic.  |
| <b>Detailed Design</b> | 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 final collection process.<br>It also deletes the account review date from CI_ADM_RVW_SCH table, and updates the number of times an account is self-cured. |

**Table 5–3 Cure Account: Sample Algorithm**

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

## 5.3 Queue Allocation: C1-ALLOCQUEU

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

**Table 5–4 Queue Allocation: C1-ALLOCQUEU**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Allocation Group Queue Allocation.  |
| <b>Detailed Description</b> | This Algorithm type is used to allocate the entities such as cases to queues.<br>For parameter "queueAllocationView" ci_allocation_monitor_vw view is shipped from product to filter cases.<br>For parameter "queueAllocationTable" ci_allocation_monitor table is shipped from product for improvising performance of batch. This is optional parameter.                           |
| <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>           | <b>Name:</b> queueAllocationView (soft parameter)<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> View for allocation<br><br><b>Name:</b> queueAllocationTable<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Table for allocation  |
| <b>Detailed Design</b>      | This algorithm receives input as Allocation Group code from the batch.<br>The view used to filter cases is accepted as an algorithm soft parameter. Product will ship CI_ALLOCATION_MONITOR_VW view.<br>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). |

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

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

## 5.4 Update Customer Switch: C1-CUSTSW

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

**Table 5–6 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.<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.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 5–7 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 |

|  |  |
|--|--|
|  | <b>Name:</b> Switch Value<br><b>Value:</b> Y |
|--|--|

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

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

**Table 5–8 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> | <p>Legal Repo Switch Name: Specify the Legal or Repo case switch column name of account extension</p> <p>For example, LEGAL_CASE_EXISTS_SW or REPO_CASE_EXISTS_SW, and so on.</p> <p>Switch Value: Please enter the switch value as Y or N.</p>  |
| <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. The soft parameter is used to identify the fields that should be updated. 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 5–9 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> |

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

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

**Table 5–10 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 on the basis of capacity set during configuration on queue admin. OverFlow cases will get assigned to Exception User. 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>■ C1 = B1 - A1 = 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>■ C2 = B2 - A2 = 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> |



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

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

**Table 5–11 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> | <p>This algorithm allocates cases to user or teams in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC).</p> <p>User Allocation Percentage based algorithm type allocates cases to users on the basis of percentage allocations set during configuration on queue admin.</p> <p>OverFlow cases will get assigned to Exception User.</p>   |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| <b>Algorithm Entity</b>     | User Allocation  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| <b>Program Type</b>         | Java   |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPerBasedAllocRoundOff  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| <b>Parameters</b>           | NA   |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| <b>Detailed Design</b>      | <ul style="list-style-type: none"> <li>■ Open unallocated cases will be allocated to valid queue users based on corresponding percentage distribution value mentioned in the Queue Details screen.</li> <li>■ Algorithm will round off the decimal percent allocated case count to the nearest whole integer value.</li> </ul> <p>For example:<br/>User allocation percentage of Queue Q1 is as follows:</p> <p><b>Table 5–12 User allocation percentage of Queue Q1</b></p> <table border="1"> <thead> <tr> <th>User</th> <th>Allocation Percentage</th> </tr> </thead> <tbody> <tr> <td>U1</td> <td>33%</td> </tr> <tr> <td>U2</td> <td>33%</td> </tr> <tr> <td>U3</td> <td>34%</td> </tr> </tbody> </table> <p>Total unallocated cases = 10<br/>Then, cases will be allocated as per following calculations:</p> <p><b>Table 5–13 Calculations for allocating cases</b></p> <table border="1"> <thead> <tr> <th>User</th> <th>Allocation Percentage</th> <th>Calculated Case Allocation</th> <th>Actual Case Allocation</th> </tr> </thead> <tbody> <tr> <td>U1</td> <td>33%</td> <td>3.3</td> <td>3</td> </tr> </tbody> </table> | User                       | Allocation Percentage  | U1 | 33% | U2 | 33% | U3 | 34% | User | Allocation Percentage | Calculated Case Allocation | Actual Case Allocation | U1 | 33% | 3.3 | 3 |
| User                        | Allocation Percentage  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| U1                          | 33%  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| U2                          | 33%  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| U3                          | 34%  |                            |                        |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| User                        | Allocation Percentage  | Calculated Case Allocation | Actual Case Allocation |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |
| U1                          | 33%  | 3.3                        | 3                      |    |     |    |     |    |     |      |                       |                            |                        |    |     |     |   |

|    |     |     |   |
|----|-----|-----|---|
| U2 | 33% | 3.3 | 3 |
| U3 | 34% | 3.4 | 3 |

However, based on the rounding-off calculations, 1 case remains unallocated. The remainder (or unallocated) cases will be assigned to the User with highest allocation percentage. This distribution will be based on the calculations of available user capacity and maximum user allocation capacity.

Thus, the final distribution looks like below:

**Table 5–14 Final Distribution**

| User | Allocation Percentage | Calculated Case Allocation | Actual Case Allocation |
|------|-----------------------|----------------------------|------------------------|
| U1   | 33%                   | 3.3                        | 3                      |
| U2   | 33%                   | 3.3                        | 3                      |
| U3   | 34%                   | 3.4                        | 4                      |

- In any case, if the user with highest allocation percentage has exhausted his/her available capacity, unallocated cases will be assigned to the user with next highest allocation percentage.

For example:

User allocation percentage of Queue Q2 is as follows:

**Table 5–15 User allocation percentage of Queue Q2**

| User | Allocation Percentage |
|------|-----------------------|
| U1   | 33%                   |
| U2   | 33%                   |
| U3   | 34%                   |

Total unallocated cases = 10

Then, cases will be allocated as per following calculations:

**Table 5–16 Calculations for allocating cases**

| User | Allocation Percentage | Calculated Case Allocation | Actual Case Allocation |
|------|-----------------------|----------------------------|------------------------|
| U1   | 33%                   | 3.3                        | 3                      |
| U2   | 33%                   | 3.3                        | 3                      |
| U3   | 34%                   | 3.4                        | 3                      |

|      | <p>User UC has exhausted the available allocation capacity. Therefore, the remaining 1 unallocated case will be assigned to any one of the UA/UB users (since they have same allocation percentages).</p> <p>The final distribution in this case will be as follows:</p> <p><b>Table 5–17 Final distribution in cases</b></p> <table border="1"> <thead> <tr> <th>User</th> <th>Allocation Percentage</th> <th>Calculated Case Allocation</th> <th>Actual Case Allocation</th> </tr> </thead> <tbody> <tr> <td>U1</td> <td>33%</td> <td>3.3</td> <td>3</td> </tr> <tr> <td>U2</td> <td>33%</td> <td>3.3</td> <td>4</td> </tr> <tr> <td>U3</td> <td>34%</td> <td>3.4</td> <td>3</td> </tr> </tbody> </table> <ul style="list-style-type: none"> <li>■ If capacities of all queue users are exhausted, unallocated cases will be assigned to the Queue Exception User.</li> </ul> | User                       | Allocation Percentage  | Calculated Case Allocation | Actual Case Allocation | U1 | 33% | 3.3 | 3 | U2 | 33% | 3.3 | 4 | U3 | 34% | 3.4 | 3 |
|------|---|----------------------------|------------------------|----------------------------|------------------------|----|-----|-----|---|----|-----|-----|---|----|-----|-----|---|
| User | Allocation Percentage   | Calculated Case Allocation | Actual Case Allocation |                            |                        |    |     |     |   |    |     |     |   |    |     |     |   |
| U1   | 33%   | 3.3                        | 3                      |                            |                        |    |     |     |   |    |     |     |   |    |     |     |   |
| U2   | 33%   | 3.3                        | 4                      |                            |                        |    |     |     |   |    |     |     |   |    |     |     |   |
| U3   | 34%   | 3.4                        | 3                      |                            |                        |    |     |     |   |    |     |     |   |    |     |     |   |

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

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

**Table 5–18 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).<br>OverFlow cases will get assigned to Exception User of the queue.   |
| <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  |
| <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</li> </ul> |

|  |  |
|--|--|
|  | <p>C1 and C2.</p> <ul style="list-style-type: none"> <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>■ <math>C3 = B3 - A3</math> = 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 the 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 <math>&lt; A</math> and <math>B</math> respectively. If value for <math>B</math> 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 = <math>A</math> or <math>B</math> 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> |
|--|--|

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

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

**Table 5–19 Vendor Allocation - % Based: C1-VENALCRR**

|                             |  |
|-----------------------------|--|
| <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).<br>OverFlow cases will get assigned to Exception User of the queue.   |
| <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   |
| <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>■ 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</li> </ul> |

|  |  |
|--|--|
|  | <p>calculation) and then in decreasing order of capacity.</p> <ul style="list-style-type: none"> <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> |
|--|--|

## 5.10 Bulk Contact Creation: C1-BLKNTCRE

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

**Table 5–20 Bulk Contact Creation: C1-BLKNTCRE**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Bulk Contact Creation Algorithm  |
| <b>Detailed Description</b> | This algorithm type is called from Bulk Contact Creation Batch.<br>It invokes business service 'C1-GenMultipleCorrespondence' which creates a customer contact for the accounts filtered by the condition builder attached to the process codes in bulk contact admin.   |
| <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>      | This algorithm will be invoked from bulk contact creation batch from where the hard parameter values are set.<br>The algorithm will call business service 'C1-GenerateCorrespondence'.<br>addMultiple() method of 'C1-GenerateCorrespondence' will be called which in turn adds customer contact to CI_CC via add () method of the same service. |

## 5.11 Cross Strategy Action Matrix: C1-CSAM

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

**Table 5–21 Cross Strategy Action Matrix: C1-CSAM**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used for Cross Strategy Action Matrix.  |
| <b>Detailed Description</b> | <p>Cross Strategy Action Matrix Algorithm Type is used by Strategy Monitor and case association process in order to take actions on existing strategies and recommended strategies based on CSAM Matrix.</p> <p>Parameters :</p> <p>Check Status- It checks the status with which the matrix has to be dealt with. Possible values are "Y" or "N"</p>   |
| <b>Algorithm Entity</b>     | Case Type- CSAM   |
| <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<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Y - Case types with Status<br/> 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 5–22 Cross Strategy Action Matrix: Sample Algorithm**

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

## 5.12 Last Payment for Account: C1-PAYDTAMTU

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

**Table 5–23 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 for written off accounts.             |
| <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. |

## 5.13 Association Review Check: C1-ASORVCHK

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

**Table 5–24 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 Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckAssociationReview   |
| <b>Parameters</b>           | <p><b>Name:</b> NextStatus<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Next Status</p> <p><b>Name:</b> AssociationReviewRequired<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> Association Review Required</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 5–25 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> |

## 5.14 Validate Expired Default Notice: C1-DEFNOEXP

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

**Table 5–26 Validate Expired Default Notice: C1-DEFNOEXP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to validate expired default notices.   |
| <b>Detailed Description</b> | <p>System should check that for associated accounts default notice has expired. This check can be for primary account or for all associated delinquent account based on parameter.</p> <ol style="list-style-type: none"> <li>1. Association Type={P,A}. P=Primary Type Association,A= Primary as well as Secondary type association</li> <li>2. To Do Type= To Do will be created if validation failure option is N.</li> <li>3. To Do Role= To Do Role for the specified To Do Type.</li> <li>4. Validationfailure Option= {Y,N}. If it is Y then case transition will be failed else a To Do will be created.</li> </ol> |
| <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.DefaultNoticeExpiryCheck  |
| <b>Parameters</b>           | <p><b>Name:</b> associationType<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Association Type</p> <p><b>Name:</b> validationfailureOption<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Validation Failure Option</p> <p><b>Name:</b> toDoType<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type</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 5–27 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> |

## 5.15 Associate Related Entity: C1-ASSOENTY

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

Table 5–28 Associate Related Entity: C1-ASSOENTY

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to associate related entities with the case.   |
| <b>Detailed Description</b> | <p>The algorithm checks the for accounts associated to the primary account. The association of the primary account is done on the basis of the persons attached to the account and their financially responsible status. If the account has the same set of financially responsible persons attached as in the case for the primary account, the account is associated. The algorithm parameter are as follows:</p> <ol style="list-style-type: none"> <li>1. To Do Role: Specifies the role for the To Do Type created in case of any exception arising in association of accounts.</li> <li>2. To Do Type: Specifies the To Do Type created in case of any exception arising in association of accounts.</li> <li>3. Host Id: Specifies the host Id.</li> </ol> |
| <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.AssociatedAccountsList  |
| <b>Parameters</b>           | <p><b>Name:</b> hostId<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Host Id</p> <p><b>Name:</b> toDoType<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> To Do Type</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 5–29 Associate Related Entity: Sample Algorithm**

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

## 5.16 Validate Legal Case Exists: C1-CHKLGL

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

**Table 5–30 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> | <p>The algorithm checks if there is already open legal case for the primary account/Associated accounts linked to the case. The algorithm takes the parameters as follows:</p> <ol style="list-style-type: none"> <li>1. To Do Role: Specifies the Role for the To Do Type.</li> <li>2. To Do Type: Specifies the to do type created when the legal case has been created from batch mode and there is open legal case for the Primary Account/Associated Accounts.</li> <li>3. Case Category: Specifies the case category for the case(LEGL is for Legal Case)</li> </ol> |
| <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.CheckLegalCase   |
| <b>Parameters</b>           | <b>Name:</b> Case Category<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> Case Category<br><br><b>Name:</b> toDoType<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> To Do Type   |
| <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 5–31 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 |

|  |   |
|--|---|
|  | <b>Name:</b> ToDoType<br><b>Value:</b> C1-TD-CL |
|--|---|

## 5.17 Assign New LSP: C1-ASGNLSP

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

**Table 5–32 Assign New LSP: C1-ASGNLSP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | This algorithm is used to assign LSP to the case.   |
| <b>Detailed Description</b> | <p>This algorithm will assign a new LSP to the current case. LSP is a external vendor which is mapped LEGAL service Type. If manual review is not required then case will automatically transition to next status mentioned in soft parameter. Below are the soft parameter example</p> <ol style="list-style-type: none"> <li>1. Next Status: value can be possible next status example{PREPLGLDOC etc.}</li> <li>2. Prv Allocation Check: Possible values {Y, N}. If this switch is Y system will check if a legal case was created for any of the accounts associated with the current legal case in past.</li> <li>3. Reset Doc Sub Date Sw = Possible values {Y, N}. Value N means document submission date from previous assignment will be copied to new assignment.</li> <li>4. Change Allocation Option= Possible values {AUTO_WITH_REVIEW, AUTO_WITHOUT_REVIEW, MANUAL}.<br/> AUTO_WITH_REVIEW= System allocation with review option.<br/> AUTO_WITHOUT_REVIEW=System allocation without review option.<br/> MANUAL=Manual allocation. System will not allocate LSP.</li> <li>5. New Allocation And Review Option= Possible values {AUTO_WITH_REVIEW,AUTO_WITH_REVIEW_PRVALLOC,AUTO_WITHOUT_REVIEW,MANUAL}<br/> AUTO_WITH_REVIEW= System allocation with review option.<br/> AUTO_WITH_REVIEW_PRVALLOC=System allocation and review will be required if previous allocation was retained.<br/> AUTO_WITHOUT_REVIEW=System allocation without review option.<br/> MANUAL=Manual allocation. System will not allocate LSP.</li> </ol> |
| <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.AssignNewLSP  |
| <b>Parameters</b>           | <p><b>Name:</b> New Allocation And Review Option<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> New Allocation and Review Option</p> <p><b>Name:</b> Change Allocation Option</p>  |

|                        |  |
|------------------------|--|
|                        | <p><b>Required (Yes/No):</b> N<br/> <b>Description:</b> Change Allocation Option</p> <p><b>Name:</b> Reset Document Submission Date Sw<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Reset Document Submission Date Switch</p> <p><b>Name:</b> Previous Allocation Check<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Previous Allocation Check</p> <p><b>Name:</b> Next Status<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Next Status</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 5–33 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> |

## 5.18 Check Approval Requirement: C1-APPRCHK

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

Table 5–34 Check Approval Requirement: C1-APPRCHK

|                    |  |
|--------------------|--|
| <b>Description</b> | This algorithm is used to check the need of approval.                            |
| <b>Detailed</b>    | This algorithm creates approval request if required based on certain conditions. |

|                         |  |
|-------------------------|--|
| <b>Description</b>      | <p>This process will check if LSP assignment needs to be approved, if LSP assignment status = "Pending Approval"</p> <p>Approval would be required if either of below is true:</p> <ul style="list-style-type: none"> <li>■ System allocation override by user i.e. user has changed the LSP assigned by the system. Set Approval Reason as "Allocation override".</li> <li>■ Exposure i.e. sum of balances for all accounts associated with the case is more than a specified threshold. However if no threshold has been specified this parameter should be ignored. Set Approval Reason as "High Exposure".</li> <li>■ In case approval is required for both the reason, concatenate the approval reasons before sending for approval.</li> </ul> <p>If approval is required:</p> <ul style="list-style-type: none"> <li>■ Transition the case to a specified status defined as the parameter.</li> </ul> |
| <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.CreateApprovalRequest  |
| <b>Parameters</b>       | <p><b>Name:</b> Exposure Threshold<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Exposure Threshold</p> <p><b>Name:</b> Approval Request Status<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Approval Request Status</p> <p><b>Name:</b> Approved Status<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Approved Status</p> <p><b>Name:</b> Reject Request Status<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Reject Request Status</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 5–35 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> Approved Status<br/><b>Value:</b> WTFRLSPACK</p> <p><b>Name:</b> Reject Request Status<br/><b>Value:</b> ASSNEWLSP</p> |
|--|---|

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

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

**Table 5–36 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>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.ResumeStatusLSP   |
| <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. |

## 5.20 Check Submission Date: CI\_CHKSUBDT1

This section provides details of the Check Submission Date: CI\_CHKSUBDT1 algorithm.

**Table 5–37 Check Submission Date: CI\_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   |
| <b>Program Type</b>         | Java  |

|                        |   |
|------------------------|---|
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDate   |
| <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 5–38 Check Submission Date: Sample Algorithm

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | CI_CHKSUBDT1C1   |
| <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> |

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

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

Table 5–39 Update LSP (CLOS): C1-LSPSTATUS

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Legal Proceedings - Update Status   |
| <b>Detailed Description</b> | <p>This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either closed or cancelled.</p> <p>Set LSP assignment status to value provided in the parameter. This should be done only for Latest LSP assignment and if it was done by current legal case.</p> <p>If Status = Closed or Cancelled set Assignment End date = Business Date</p> <p>Status possible values {CLOS,REJ,CAN,PNAP}</p> <p>CLOS=Closed<br/>REJ=Rejected<br/>PNAP=Pending for Approval.</p> |
| <b>Algorithm Entity</b>     | Case Type-Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program</b>              | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssign   |



|                        |   |
|------------------------|---|
| <b>Name</b>            | nment   |
| <b>Parameters</b>      | <b>Name:</b> Lsp Assignment Status<br><b>Required (Yes/No):</b> Y<br><b>Description:</b> LSP Assignment Status  |
| <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 5–40 Update LSP (CLOS): Sample Algorithm

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-LSPSTATUS   |
| <b>Parameters</b>     | <b>Name:</b> : Lsp Assignment Status<br><b>Value:</b> CLOS |

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

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

Table 5–41 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.UpdateLSPAssignment   |
| <b>Parameters</b>           | <b>Name:</b> Lsp Assignment Status<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 5–42 Update LSP (CANCEL): Sample Algorithm

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-LSPSTACAN  |
| <b>Parameters</b>     | <b>Name:</b> : Lsp Assignment Status<br><b>Value:</b> CAN |

## 5.23 Collateral Verification: C1-VRFYCOLS

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

Table 5–43 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 no 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<br/> <b>Required (Yes/No):</b> N<br/> <b>Description:</b> Collateral Type</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 5–44 Collateral Verification: Sample Algorithm

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

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

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

Table 5–45 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> </ul> |

|                         |   |
|-------------------------|---|
|                         | <ul style="list-style-type: none"> <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  |
| <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.  |

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

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

**Table 5–46 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. |
|------------------------|---|

## 5.26 Update Collateral Property: C1-UPCOLPROP

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

**Table 5–47 Update Collateral Property: C1-UPCOLPROP**

|                             |  |
|-----------------------------|--|
| <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<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 the Asset Repossession Process case. It updates the collateral Properties like IS_LEGAL_SW, IS_REPO_SW depending on user inputs.  |

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

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

**Table 5–48 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>           | <b>Name:</b> To Do Type1  |

|                        |   |
|------------------------|---|
|                        | <p><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type 1</p> <p><b>Name:</b> To Do Type2<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type 2</p> <p><b>Name:</b> To Do Type3<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type 3</p> <p><b>Name:</b> To Do Type4<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type 4</p> <p><b>Name:</b> To Do Type5<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> To Do Type 5</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 5–49 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> |

## 5.28 Validations for Mandatory Characteristics: C1-CHARVALS

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

Table 5–50 Validations for Mandatory Characteristics:C1-CHARVALS

|                             |   |
|-----------------------------|---|
| <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> Reference Characteristics Value</p> <p><b>Name:</b> ReferenceCharacteristic<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Reference Characteristic Type</p> <p><b>Name:</b> CaseCharacteristics1<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Case Characteristics</p> <p><b>Name:</b> CaseCharacteristics2<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Case Characteristics</p> <p><b>Name:</b> CaseCharacteristics3<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Case Characteristics</p> <p><b>Name:</b> CaseCharacteristics4<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Case Characteristics</p> <p><b>Name:</b> CaseCharacteristics5<br/><b>Required (Yes/No):</b> N<br/><b>Description:</b> Case Characteristics</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 5–51 Validations for Mandatory Characteristics: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-CHARVALS   |
| <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> |

## 5.29 Validations for Mandatory Characteristics: C1-CHARVALA

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

**Table 5–52 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           |
| <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 5–53 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</p> |



|  |   |
|--|---|
|  | <b>Value:</b><br><br><b>Name:</b> CaseCharacteristics4<br><b>Value:</b><br><br><b>Name:</b> CaseCharacteristics5<br><b>Value:</b> |
|--|---|

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

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

**Table 5–54 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  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Type<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> To Do Type</p> <p><b>Name:</b> Collateral Status<br/> <b>Required (Yes/No):</b> Y<br/> <b>Description:</b> Collateral Status</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 5–55 Update Collateral Status in the Host: Sample Algorithm**

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-UPCOLLSTX   |
| <b>Parameters</b>     | <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> |

## 5.31 Initiate Collateral Valuation: C1-COLLVALX

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

**Table 5–56 Initiate Collateral Valuation: C1-COLLVALX**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Initiate collateral valuation  |
| <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-Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Type<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> To Do Type</p> <p><b>Name:</b> Days Since Closure Of Last To Do<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Days Since Closure Of Last To Do</p> <p><b>Name:</b> Assessment Expiry Days<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Assessment Expiry Days</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 5–57 Initiate Collateral Valuation: Sample Algorithm**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-COLLVALX   |
| <b>Parameters</b>     | <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> |
|--|---|

## 5.32 Validation Settlement: C1-VALSET

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

**Table 5–58 Validation Settlement: C1-VALSET**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validation Settlement   |
| <b>Detailed Description</b> | <p>This algorithm will perform following actions:<br/>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 will fail if above validations fail.</p> |
| <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.ValidateCollateralSettlementStatus  |
| <b>Parameters</b>           | <p><b>Name:</b> Realization Status<br/><b>Required (Yes/No):</b> Y<br/><b>Description:</b> Realization Status</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 5–59 Validation Settlement: Sample Algorithm**

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

## 5.33 Initiate LMI Process: C1-INITLMI

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

Table 5–60 Initiate LMI Process: C1-INITLMI

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Initiate LMI Process   |
| <b>Detailed Description</b> | <p>Parameters to the algorithm must be as follows:</p> <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.InitiateLMI P  |
| <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 5–61 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> |

## 5.34 PTP Active Algorithm: C1-PTPACTIVE

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

**Table 5–62 PTP Active Algorithm: C1-PTPACTIVE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | PTP Active Algorithm  |
| <b>Detailed Description</b> | <p>This algorithm is used to perform additional processing when the status of a PTP becomes Active.</p> <p>Customer contacts can be generated via this algorithm. Contact class, method and type have to be specified.</p>  |
| <b>Algorithm Entity</b>     | PTP Active Algorithm  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> contactTypeForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Required (Yes/No):</b> No</p> |

|                        |   |
|------------------------|---|
|                        | <p><b>Description:</b> Contact Method for Letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Type for SMS</p> <p><b>Name:</b> contactClassForSMS<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for SMS</p> <p><b>Name:</b> contactMethodForSMS<br/><b>Required (Yes/No):</b> No<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 5–63 PTP Active Algorithm: Sample Algorithm

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-PTPKEPT   |
| <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> |

## 5.35 PTP Kept Algorithm: C1-PTPKEPT

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

Table 5–64 PTP Kept Algorithm: C1-PTPKEPT

|                    |                    |
|--------------------|--------------------|
| <b>Description</b> | PTP Kept Algorithm |
|--------------------|--------------------|

|                             |  |
|-----------------------------|--|
| <b>Detailed Description</b> | This algorithm is used to perform additional processing when the status of a PTP becomes Kept.<br>Customer Contacts can be generated via this algorithm. Contact Class, method and type have to be specified.  |
| <b>Algorithm Entity</b>     | PTP Kept Algorithm   |
| <b>Program Type</b>         | Java   |
| <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> No<br/><b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Method for Letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Type for SMS</p> <p><b>Name:</b> contactClassForSMS<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for SMS</p> <p><b>Name:</b> contactMethodForSMS<br/><b>Required (Yes/No):</b> No<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 5–65 PTP Active Algorithm: Sample Algorithm

|                       |  |
|-----------------------|--|
| <b>Algorithm Name</b> | C1-PTPKEPT   |
| <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> |
|--|---|

## 5.36 PTP Broken Algorithm: C1-BRKPTPNGP

This section provides details of the PTP Broken Algorithm: C1-BRKPTPNGP algorithm.

*Table 5–66 PTP Broken Algorithm: C1-BRKPTPNGP*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | PTP Broken Algorithm   |
| <b>Detailed Description</b> | This algorithm is used to perform additional processing when the status of a PTP is set to Broken.<br>Customer Contacts can be generated via this algorithm.   |
| <b>Algorithm Entity</b>     | PTP Broken Algorithm   |
| <b>Program Type</b>         | Java   |
| <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> No<br/><b>Description:</b> Contact Type for Letter generation</p> <p><b>Name:</b> contactClassForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for letter generation</p> <p><b>Name:</b> contactMethodForLetter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Method for letter generation</p> <p><b>Name:</b> contactTypeForSMS<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Class for SMS generation</p> |



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

## 5.37 Rule facts populating algorithm: C1-BRLSR

This section provides details of the Rule Facts Populating Algorithm: C1\_BRLSR algorithm.

*Table 5–67 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</p> |

|                        |   |
|------------------------|---|
|                        | <p><b>Required (Yes/No):</b> No<br/> <b>Description:</b> Primary Key name of defined BO.</p>  |
| 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).<br/> 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.</p> <p>These populated facts will act as input to defined rule through soft parameter.</p> |
|--|--|

### Sample Algorithm

*Table 5–68 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> |

|  |   |
|--|---|
|  | <b>Name:</b> Pre Populated Rule Facts Algorithm Code<br><b>Value:</b> |
|--|---|

## 5.38 Borrower Centric Case Lifecycle

This table provides details of the Borrower Level: C1-ASSODELAC algorithm.

**Table 5–69 Borrower Level: C1-ASSODELAC**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Associate new delinquent account of the customer  |
| <b>Detailed Description</b> | Associate delinquent accounts where the customer is the main customer to the case.  |
| <b>Algorithm Entity</b>     | Case Type -Enter Status   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssociateDelinquentAccount  |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | It is invoked in Pending status of borrower centric case. Transition to Borrower Centricity happens only if a customer has multiple delinquent accounts where he is the main customer only. |

This table provides details of the Borrower Level : C1-BRWRSW\_Y algorithm.

**Table 5–70 Borrower Level : C1-BRWRSW\_Y**

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

This table provides details of the Borrower Level : C1-BRWTRNDF algorithm.

**Table 5–71 Borrower Level : C1-BRWTRNDF**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-BRWTRNDF                               |
| <b>Parameters</b>     | <b>Name:</b> Wait Days<br><b>Value:</b> 0 |

This table provides details of the Borrower Level : C1-BRWRSW\_N algorithm.

**Table 5–72 Borrower Level : C1-BRWRSW\_N**

|                       |   |
|-----------------------|---|
| <b>Algorithm Name</b> | C1-BRWRSW_N   |
| <b>Parameters</b>     | <b>Name:</b> Customer Level Switch Name<br><b>Value:</b> BRRWR_SW<br><br><b>Name:</b> Switch Value<br><b>Value:</b> N |

## 5.39 Update Collection Address on Borrower Panel

This table provides details of the Person Address Update -Pre-Processing: C1-PADDPRE algorithm.

**Table 5–73 Person Address Update -Pre-Processing: C1-PADDPRE**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Person Address Update - Pre Processing   |
| <b>Detailed Description</b> | Person Address PreProcessing algorithm. Attached on BO pre processing spot. This is a hook provided to customization. This can be utilized to validate the address data.   |
| <b>Algorithm Entity</b>     | Business Object -Pre-Processing  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.address.PersonCollectionAddressPreProcess  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | This algorithm is hooked in PreprocessBusinessObjectRequestAlgorithmSpot. Business object Name: C1-PERADDRCO. Currently there is no logic inside this algorithm. Implementation team can write their own algorithm in this spot and they can attach this in C1-PERADDRCO |

This table provides details of the Collection Address Post Processing: C1-PERADDPP algorithm.

**Table 5–74 Collection Address Post Processing: C1-PERADDPP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Person Address Update - Post Processing   |
| <b>Detailed Description</b> | This is a reference implementation of Post processing algorithm. Customization team can utilize this hook. This is a sample algorithm without having any logic. |
| <b>Algorithm Entity</b>     | Collection Person Address - Post Process  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.address.CollectionPersonAddressPostProcessing   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | This is a reference implementation of Post processing algorithm. Customization team can utilize this hook. This is a sample algorithm without having any logic. |

## 5.40 Update Collection Contact Point

This table provides details of Person Contact Point Update - Pre Processing: C1-PCONTPRE algorithm.

**Table 5–75 Person Contact Point Update - Pre Processing: C1-PCONTPRE**

|                                |  |
|--------------------------------|--|
| <b>Description</b>             | Person Contact Point Update - Pre Processing   |
| <b>Description Description</b> | Contact Point PreProcessing algorithm is attached on BO pre processing spot. This hook is provided for customization and can be utilized to validate the contact point data. |
| <b>Algorithm Entity</b>        | Business Object - Pre Processing   |
| <b>Program Type</b>            | Java   |
| <b>Program Name</b>            | com.splwg.ccb.domain.collection.address.ContactPreferencePreProcess  |
| <b>Parameters</b>              | NA   |
| <b>Detailed Design</b>         | Contact Point PreProcessing algorithm is attached on BO pre processing spot. This hook is provided for customization and can be utilized to validate the contact point data. |

## 5.41 Bankruptcy Process

This table provides details of the Check if Special Case Already exist on the Customer- Enter Processing: C1-CKSPLCASE algorithm.

**Table 5–76 Check if Special Case Already Exist on the Customer- Enter Processing: C1-CKSPLCASE**

|                              |  |
|------------------------------|--|
| <b>Descripti on</b>          | Check if any active case is present of a given case category or case type on the customer - Enter Processing   |
| <b>Detailed Descripti on</b> | <p>Check if any active case is present of a given case category or case type on the customer. Processing steps are as below:</p> <ol style="list-style-type: none"> <li>1. If only Case Category is specified check if any active case is running on the customer whose <ol style="list-style-type: none"> <li>a. Case category is same as the parameter set for the algorithm</li> </ol> </li> <li>2. If Case Type is specified check if any active case is running on the customer whose <ol style="list-style-type: none"> <li>a. Case type is same as the parameter set for the algorithm</li> </ol> </li> <li>3. If yes validation should fail</li> <li>4. If Consider Enterprise Id = Y perform the check for all the parties with same Enterprise Id.<br/>Consider Enterprise Id value should be "YES" or "NO"</li> </ol> |
| <b>Algorith m Entity</b>     | Case Type -Enter Status  |
| <b>Program Type</b>          | Java   |
| <b>Program Name</b>          | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.CheckBankruptcyCaseExist  |
| <b>Paramete rs</b>           | <p><b>Name:</b> Case Category<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Case Category</p>  |

|                        |  |
|------------------------|--|
|                        | <p><b>Name:</b> Case Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Case Type</p> <p><b>Name:</b> Consider Enterprise Id<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Enterprise Id</p> |
| <b>Detailed Design</b> | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook.  |

This table provides details of the Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullINDAcc algorithm.

**Table 5–77 Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullINDAcc**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Pull all the non delinquent accounts of the customer into collections- Enter Processing   |
| <b>Detailed Description</b> | <p>Processing steps are as below:</p> <ul style="list-style-type: none"> <li>■ Pull all Not in Collections accounts into OB Collections (from OBP) where the associated customer is one of the borrower.</li> <li>■ If Account Relationships = MC consider only the accounts where the customer is primary owner.<br/> If Account Relationships = FO consider all accounts where the customer is a financial owner.<br/> If Account Relationship = All consider all accounts where the customer is a financial or non-financial owner.</li> <li>■ If Consider Enterprise Id = Yes; Determine the Enterprise Id corresponding the party id; then determine the party id corresponding to OBP host and then proceed to pull the accounts.<br/> Possible Values of Account Relationships MC, FO, ALL<br/> Possible Values for Consider Enterprise Id Yes/No</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.bankruptcy.BankruptcyPullNonDelinquentAcc   |
| <b>Parameters</b>           | <p><b>Name:</b> Account Relationships<br/> (MC,FO,ALL)<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Account Relationships<br/> <b>Name:</b> Consider Enterprise Id (Yes/No)</p>  |

|                        |   |
|------------------------|---|
|                        | <b>Required (Yes/No):</b> Yes<br><b>Description:</b> Enterprise Id  |
| <b>Detailed Design</b> | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook. |

This table provides details of the Associate all accounts to the case where customer is a primary borrower- Enter Processing: C1-ASSCTEACC algorithm.

**Table 5–78 Associate all accounts to the case where customer is a primary borrower- Enter Processing: C1-ASSCTEACC**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Associate all accounts to the case where customer is a primary borrower.   |
| <b>Detailed Description</b> | Associate all accounts to the case where customer is a primary borrower<br>For the primary customer associated with the case: <ul style="list-style-type: none"> <li>■ Get all accounts where this customer is primary owner and the accounts are In Collections. (Fetch accounts based on Enterprise Id if Consider Enterprise ID = Y).</li> <li>■ Shortlist the accounts that are not yet associated with the case.</li> <li>■ Associate the shortlisted accounts with the case.<br/>Consider Enterprise Id value should be "YES" or "NO"</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.bankruptcy.BankruptcyAssociateAcc  |
| <b>Parameters</b>           | <b>Name:</b> Consider Enterprise Id<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Consider Enterprise Id   |
| <b>Detailed Design</b>      | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook.  |

This table provides details of the Exclude all the associated accounts from Dialer- Enter Processing: C1-ExcAccDir algorithm.

**Table 5–79 Exclude all the associated accounts from Dialer- Enter Processing: C1-ExcAccDir**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Exclude all the associated accounts from Dialer- Enter Processing   |
| <b>Detailed Description</b> | This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Exclude all the associated accounts from Dialer.<br>For all the accounts associated with the case: <ul style="list-style-type: none"> <li>■ Call the Dialer Exclusion Service to exclude the accounts from feed to Dialer</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.bankruptcy.BankruptcyExcludeAccDir          |
| <b>Parameters</b>       | NA  |
| <b>Detailed Design</b>  | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook. |

This table provides details of the Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing: C1-IniClVal algorithm.

**Table 5–80 Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing: C1-IniClVal**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing  |
| <b>Detailed Description</b> | For each collateral on the associated account if last valuation was done 'X' days before than create a Collateral Valuation Task.<br>Enter the Collateral Code; Collateral Type and Collateral Description as Remarks<br>Exclude Collaterals with Collateral Types specified in parameter.<br>Also Exclude Collaterals that have been already Repossessed or Sold.<br>Values of Validation Date: POSTING DATE, SYSTEM DATE       |
| <b>Algorithm Entity</b>     | Case Type -Enter Status  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyInitiateCollateralValuation   |
| <b>Parameters</b>           | <p><b>Name:</b> Valuation Expiry days<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Valuation Expiry days</p> <p><b>Name:</b> Collateral Valuation Task<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Collateral Valuation Task</p> <p><b>Name:</b> Administration Queue<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Administration Queue</p> <p><b>Name:</b> Exclude Collateral Types</p> |

|                        |   |
|------------------------|---|
|                        | <p><b>Required (Yes/No):</b> No<br/> <b>Description:</b> Exclude Collateral Types</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b> | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook.   |

This table provides details of the Monitor if any of the associated account need to be charged off and monitor delinquency- Monitoring: C1-MTRCRGDQY algorithm.

**Table 5–81 Monitor if any of the associated account need to be charged off and monitor delinquency- Monitoring: C1-MTRCRGDQY**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Monitor if any of the associated account need to be charged off and monitor delinquency- Monitoring   |
| <b>Detailed Description</b> | <p>Monitor Delinquency:<br/> If any of the associated account has delinquency Start Date = Today's posting date Create Bankruptcy Notification as:<br/> 'Account &lt;Account Number&gt; has become Delinquent'<br/> Set Display Date of the case to current business date.</p> <p>Monitor Charge Off:<br/> If any of the associated account has DPD= Charge Off Threshold Create Bankruptcy Notification as<br/> 'Account &lt;Account Number&gt; can be Charged Off'<br/> Set Display Date of the case to current business date.</p> <p>If Secured Accounts = Yes than associated accounts with Secured Switch = Y should also be considered.</p> <p>Monitor Delinquency = "Y" or "N"<br/> Monitor Charge Off = "Y" or "N"<br/> Secured Accounts = "Y" or "N"<br/> Values of Validation Date: POSTING DATE, SYSTEM DATE</p> |
| <b>Algorithm Entity</b>     | Case Type -Auto Transition  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorChargeOffDelinquency  |
| <b>Parameters</b>           | <p><b>Name:</b> Monitor Delinquency<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Monitor Delinquency</p> <p><b>Name:</b> Monitor Charge Off<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Monitor Charge Off</p>   |

|                        |   |
|------------------------|---|
|                        | <p><b>Name:</b> Charge Off Threshold D P D<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Charge Off Threshold D P D</p> <p><b>Name:</b> Secured Accounts<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Secured Accounts</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b> | This is a reference implementation of Monitoring algorithm. Customization team can utilize this hook.   |

This table provides details of the Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG algorithm.

**Table 5–82 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Notify the Bankruptcy Specialist on Hearing Dates- Monitoring   |
| <b>Detailed Description</b> | <p>If 341 Hearing Date has been captured and is in future<br/> Create a notification for the Bankruptcy Specialist when the 341 Hearing date has been passed. i.e. when Business Date = 341 Hearing Date + 1<br/> Notification:<br/> "Capture details of 341 Hearing"<br/> Set Display Date of the case to current Business Date</p> <p>If Objection Hearing Date has been captured and is in future<br/> Create a notification for the Bankruptcy Specialist when the Objection Hearing date has been passed. i.e. when Business Date = Objection Hearing Date + 1<br/> Notification:<br/> "Capture details of Objection Hearing for Debtors Proposed Plan"<br/> Set Display Date of the case to current Business Date<br/> Values of Validation Date: POSTING DATE, SYSTEM DATE</p> |
| <b>Algorithm Entity</b>     | Case Type - Auto Transitions  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitor341Hearing  |
| <b>Parameters</b>           | <p><b>Name:</b>Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p>  |
| <b>Detailed Design</b>      | This is a reference implementation of Monitoring algorithm. Customization team can utilize this hook.   |

This table provides details of the Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring: C1-MTRPYMPLN algorithm.

**Table 5–83 Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring: C1-MTRPYMPLN**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring   |
| <b>Detailed Description</b> | <p>If for any of the associated account on the case the days since the last PTP Broken reaches X days a notification should be created on the case.</p> <p>The PTP Type specified in the parameter should be considered</p> <p>Notification: &lt;PTP Type&gt; broken for account &lt;Account Number&gt;. Days since plan broken &lt;Days Since PTP Broken&gt;.</p> <p>Set Display Date of the case to current business date.</p> <p>Values of Validation Date: POSTING DATE, SYSTEM DATE</p> |
| <b>Algorithm Entity</b>     | Case Type -Auto Transitions  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorPaymentPlan  |
| <b>Parameters</b>           | <p><b>Name:</b>P T P Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> P T P Type</p> <p><b>Name:</b>Days Since P T P Broken<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Days Since P T P Broken</p> <p><b>Name:</b>Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p>   |
| <b>Detailed Design</b>      | This is a reference implementation of Monitoring algorithm. Customization team can utilize this hook.  |

This table provides details of the Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD algorithm.

**Table 5–84 Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Notify the Bankruptcy Specialist if the Liquidation reaches a specific status.  |
| <b>Detailed Description</b> | <p>Notify the Bankruptcy Specialist if the Liquidation reaches a specific status.</p> <p>If for any of the associated account if the liquidation case reaches a specific status than create a notification for the Bankruptcy Specialist. Notification:</p> <p>"Liquidation for Account &lt;Account Number&gt;; Collateral &lt;Collateral Code&gt; has reached status</p> |

|                         |  |
|-------------------------|--|
|                         | <Case Status><br>Set Display Date of the Bankruptcy Case to Business Date                                    |
| <b>Algorithm Entity</b> | Case Type -Enter Status  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorAssetLiquidation |
| <b>Parameters</b>       | <b>Name:</b> Validation Date<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Validation Date         |
| <b>Detailed Design</b>  | This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook.  |

This table provides details of the Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring: C1-MTRHRNGDT algorithm.

**Table 5–85 Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring: C1-MTRHRNGDT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring   |
| <b>Detailed Description</b> | If for any of the associated account on the case if the RFS Hearing Date is reached<br>Create Notification:<br>"Capture details for RFS Hearing for Account <Account Number><br>When Business date = Hearing Date + 1<br>Set Display Date of the case to current Business Date<br>Values of Validation Date: POSTING DATE, SYSTEM DATE |
| <b>Algorithm Entity</b>     | Case Type -Auto Transition   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorHearingDate  |
| <b>Parameters</b>           | <b>Name:</b> Validation Date<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Validation Date   |
| <b>Detailed Design</b>      | This is a reference implementation of Monitoring algorithm. Customization team can utilize this hook.  |

This table provides details of the Determine in which status the case should proceed for Bankruptcy Treatment- Post Processing C1-DTMBKTRTM algorithm.

**Table 5–86 Determine in which status the case should proceed for Bankruptcy Treatment- Post Processing C1-DTMBKTRTM**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Determine in which status the case should proceed for Bankruptcy Treatment - Post Processing  |
| <b>Detailed Description</b> | <p>This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Determine in which status the case should proceed for Bankruptcy Treatment. Bankruptcy Chapter Field should be passed as a Filing Information Chapter(FC) or Converted to Chapter(CC) as an input parameter</p> <p>If Bankruptcy Chapter = Chapter 7<br/>Then Transition to Manage Chapter 7 Bankruptcy Status</p> <p>If Bankruptcy Chapter = Chapter 13<br/>Then Transition to Manage Chapter 13 Bankruptcy Status</p> <p>If Bankruptcy Chapter = Chapter other than 7 or 13<br/>Then Transition to Other Bankruptcy Status</p> <p>Bankruptcy Chapter Field = "FC" or "CC"<br/>Where "FC" = Filing Chapter and "CC"=Convert to chapter</p> |
| <b>Algorithm Entity</b>     | Result Type -Post Processing  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.DetermineBankruptcyTreatment   |
| <b>Parameters</b>           | <p><b>Name:</b>Bankruptcy Chapter Field<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Bankruptcy Chapter Field</p> <p><b>Name:</b>Manage Chapter7 Bankruptcy Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Manage Chapter7 Bankruptcy Status</p> <p><b>Name:</b>Manage Chapter13 Bankruptcy Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Manage Chapter13 Bankruptcy Status</p> <p><b>Name:</b>Other Bankruptcy Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Other Bankruptcy Status</p>   |
| <b>Detailed Design</b>      | This is a reference implementation of Result type Post processing algorithm. Customization team can utilize this hook.  |

This table provides details of the Validate if appropriate Case Details have been entered by the user- Post Processing C1-VLDBCDATA algorithm.

**Table 5–87 Validate if appropriate Case Details have been entered by the user- Post Processing C1-VLBCDATA**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate if appropriate Case Details have been entered by the user- Post Processing   |
| <b>Detailed Description</b> | <p>Validate if the Dynamic Panel Fields mentioned for the corresponding Dynamic panels have some values for the case.<br/>         If yes the Follow Up is saved successfully.<br/>         If no system should throw an error message for the first blank field that it will encounter.<br/>         Error Message: "&lt;Field Name&gt; cannot be blank"<br/>         Possible values for Panel Names and Panel fields belonging to that Panel are as follows:<br/>         Panel Name : bankruptcyTrusteeInfoPanel<br/>         Corresponding Panel Fields:</p> <ul style="list-style-type: none"> <li>■ ENTITY_NAME,PHONE,EMAIL,FAX_NUMBER,CONTACT_POINT_NAME,CONTACT_POINT_PHONE_NUM,CONTACT_POINT_EMAIL,CONTACT_POINT_FAX</li> </ul> <p>Panel Name : bankruptcyProcessingInfoPanel<br/>         Corresponding Panel Fields :</p> <ul style="list-style-type: none"> <li>■ HEARING_DATE,HEARING_LOCATION,LENDER_COLL_VAL_DATE,LENDER_COLL_VAL,DISCHARGE_DATE,DISMISSED_DATE,CHAPTER_CODE,COVERSION_REMARKS,CONVERSION_DATE,HEARING_ADD_INFO</li> </ul> <p>Panel Name : bankruptcyDebtorAttorneyPanel<br/>         Corresponding Panel Fields:</p> <ul style="list-style-type: none"> <li>■ FIRM_NAME,PHONE,ENTITY_NAME,DEBTOR_ADDRESS</li> </ul> <p>Panel Name : bankruptcyFilingInfoPanel<br/>         Corresponding Panel Fields :</p> <ul style="list-style-type: none"> <li>■ DATE_OF_BNKPT_CASE_FILE,BNKPT_CASE_NUM,COURT,CHAPTER</li> </ul> <p>Panel Name : bankruptcyConfirmPlanInformationPanel<br/>         Corresponding Panel Fields :</p> <ul style="list-style-type: none"> <li>■ RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT</li> </ul> <p>Panel Name : bankruptcyDebtorProposedPlanInfoPanel<br/>         Corresponding Panel Fields :</p> <ul style="list-style-type: none"> <li>■ RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT,OBJECTION_DATE,OBJECTION_OUTCOME,HEARING_DATE</li> </ul> <p>Panel Name : bankruptcyLegalCounselInfoPanel<br/>         Corresponding Panel Fields :</p> <ul style="list-style-type: none"> <li>■ ASSIGNED_DATE,COUNSEL_NAME,CONTACT_POINT_NAME,EMAIL,PHONE,ALTERNATE_PHONE</li> </ul> |
| <b>Algorithm</b>            | Result Type -Post Processing  |

|                           |   |
|---------------------------|---|
| <b>m Entity</b>           |   |
| <b>Program Type</b>       | Java  |
| <b>Program Name</b>       | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.ValidateBankruptcyCaseData   |
| <b>Parameters</b>         | <p><b>Name:</b>Dynamic Panel 1<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 1</p> <p><b>Name:</b>Dynamic Panel 2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 2</p> <p><b>Name:</b>Dynamic Panel 3<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 3</p> <p><b>Name:</b>Dynamic Panel 4<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 4</p> <p><b>Name:</b>Dynamic Panel 5<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 5</p> <p><b>Name:</b>Dynamic Panel 1 Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 1 Fields</p> <p><b>Name:</b>Dynamic Panel 2 Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 2 Fields</p> |
| <b>Parameters (Cont.)</b> | <p><b>Name:</b>Dynamic Panel 3 Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 3 Fields</p> <p><b>Name:</b>Dynamic Panel 4 Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel 4 Fields</p> <p><b>Name:</b>Dynamic Panel 5 Fields</p>   |



|                        |  |
|------------------------|--|
|                        | <b>Required (Yes/No):</b> No<br><b>Description:</b> Dynamic Panel 5 Fields<br><br><b>Name:</b> Case Characteristics<br><b>Required (Yes/No):</b><br><b>Description:</b> Case Characteristics |
| <b>Detailed Design</b> | This is a reference implementation of Result type Post processing algorithm. Customization team can utilize this hook.   |

This table provides details of the Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK algorithm.

**Table 5–88 Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Notify Bankruptcy Specialist when a Payment Plan status becomes Kept   |
| <b>Detailed Description</b> | Create Notification<br>Notification: <PTP Type> Kept for account <Account Number>.<br>Set Display Date of the case to current business date. |
| <b>Algorithm Entity</b>     | Business Object -Enter Status  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyPaymentPlanKept                                   |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | This is a reference implementation of Business Object Enter Processing algorithm. Customization team can utilize this hook.                  |

This table provides details of the Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP algorithm.

**Table 5–89 Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Notify Bankruptcy Specialist of Task Completion - Post Processing   |
| <b>Detailed Description</b> | Create Notification<br>Notification: <Task Id> - <Task Name> complete for <Account Number>.<br>Set Display Date of the case to current business date. |
| <b>Algorithm Entity</b>     | TO DO Type-Post Processing  |
| <b>Program Type</b>         | Java  |

|                        |  |
|------------------------|--|
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyTaskCompletion          |
| <b>Parameters</b>      | NA   |
| <b>Detailed Design</b> | This is a reference implementation TO DO Type-Post Processing algorithm. Customization team can utilize this hook. |

**Table 5–90 Joint Bankruptcy - Associate other customers to the Bankruptcy case C1-ASSCUSTJB**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Joint Bankruptcy - Associate other customers to the Bankruptcy case   |
| <b>Detailed Description</b> | Associate additional customers specified on the UI that exist in OB Collections.<br>(Assumption - If the party does not exist in OB Collection assumption is the party is pulled in OB Collections from OBP through UI or through pull non delinquent accounts) |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.JointBankruptcyAssociateCust   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | Associate additional customers specified on the UI that exist in OB Collections.<br>(Assumption - If the party does not exist in OB Collection assumption is the party is pulled in OB Collections from OBP through UI or through pull non delinquent accounts) |

This table provides details of the Pay Plan for a Bankruptcy Case- Enter Processing: C1-CRTATP algorithm.

**Table 5–91 Create Pay Plan for a Bankruptcy Case - Enter Processing: C1-CRTATP**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Algorithm to create Pay Plan for a Bankruptcy Case  |
| <b>Detailed Description</b> | This algorithm will create a dummy pay plan for all accounts associated with a bankruptcy case. The pay plan is created with pending status in the following tables :<br><ol style="list-style-type: none"> <li>1. CI_BKPTCY_PAY_PLAN_INFO</li> <li>2. CI_BKPTCY_PAY_PLAN_DTLS</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BankruptcyPayPlanCreation  |
| <b>Parameters</b>           | NA  |

|                        |  |
|------------------------|--|
| <b>Detailed Design</b> | This algorithm will create a dummy pay plan for all accounts associated with a bankruptcy case. The pay plan is created with pending status. |
|------------------------|--|

This table provides details of the Pay Plan for a Bankruptcy Case- Enter Processing: C1-CLDATP algorithm.

**Table 5–92 Close Pay Plan for a Bankruptcy Case - Exit Processing: C1-CLDATP**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm to close Arrearage Pay Plan for a Bankruptcy Case  |
| <b>Detailed Description</b> | <p>This algorithm will close the pay plan for all accounts associated with a bankruptcy case. The pay plan is marked with close status in the following tables :</p> <ol style="list-style-type: none"> <li>1. CI_BKPTCY_PAY_PLAN_INFO</li> <li>2. CI_BKPTCY_PAY_PLAN_DTLS</li> <li>3. CI_BKPTCY_PAY_PLAN_SCHED</li> </ol> <p>Associate additional customers specified on the UI that exist in OB Collections.<br/>(Assumption - If the party does not exist in OB Collection assumption is the party is pulled in OB Collections from OBP through UI or through pull non delinquent accounts)</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.bankruptcy.arrearage.BkptcyPayPlanClosure  |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | This algorithm will close the pay plan for all accounts associated with a bankruptcy case. The pay plan is marked with closed status.  |

This table provides details of the Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days - Monitoring: C1-MTRARPLNT algorithm.

**Table 5–93 Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days - Monitoring: C1-MTRARPLNT**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days  |
| <b>Detailed Description</b> | <p>Algorithm to notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days of an account, if these values are above the threshold values provided as parameters. The required parameters are :</p> <ul style="list-style-type: none"> <li>Arrearage Plan Threshold Days</li> <li>Arrearage Plan Threshold Amount</li> <li>Confirmed Plan Threshold Days</li> <li>Confirmed Plan Threshold Amount</li> <li>Notification Date Type</li> </ul> <p>Notification is generated as -&gt;</p> |

|                         |   |
|-------------------------|---|
|                         | <Arrearage/Confirmed Plan> amount for Account Number <account no> of <currency symbol> <overdue amount> is overdue by <overdue no of days> Days   |
| <b>Algorithm Entity</b> | Case Type - Auto Transition   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Banckruptcy MonitorArrearagePlanNotification   |
| <b>Parameters</b>       | <p><b>Name:</b> Arrearage Plan Threshold Days<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Arrearage Plan Threshold Days</p> <p><b>Name:</b> Arrearage Plan Threshold Amount<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Arrearage Plan Threshold Amount</p> <p><b>Name:</b> Confirmed Plan Threshold Days<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Confirmed Plan Threshold Days</p> <p><b>Name:</b> Confirmed Plan Threshold Amount<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Confirmed Plan Threshold Amount</p> <p><b>Name:</b> Notification Date Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Notification Date Type</p> |
| <b>Detailed Design</b>  | <p>Algorithm to notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days of an account, if these values are above the threshold values provided as parameters. The required parameters are :</p> <ol style="list-style-type: none"> <li>1. Arrearage Plan Threshold Days</li> <li>2. Arrearage Plan Threshold Amount</li> <li>3. Confirmed Plan Threshold Days</li> <li>4. Confirmed Plan Threshold Amount</li> <li>5. Notification Date Type</li> </ol> <p>Notification is generated as -&gt;<br/> &lt;Arrearage/Confirmed Plan&gt; amount for Account Number &lt;account no&gt; of &lt;currency symbol&gt; &lt;overdue amount&gt; is overdue by &lt;overdue no of days&gt; Days</p>           |

**Table 5–94 Set or Reset Account level Warning Indicator for Bankruptcy - Enter Processing: C1-SETWI**

|                    |   |
|--------------------|---|
| <b>Description</b> | Set or Reset Account level Warning Indicator for Bankruptcy                                     |
| <b>Detail</b>      | This Algorithm Set or Reset the Account level Warning Indicators of all the associated accounts |

|                             |   |
|-----------------------------|---|
| <b>d<br/>Description</b>    | of Bankruptcy.<br>This will exclude the Charge-Off Accounts. ( Based on RECOVERY_SW in CI_ACCT_EXTN table).<br>Risk Indicator Codes should be comma separated.<br>Values:<br>Risk Indicator = SET or RESET<br>Risk Indicator Code = <Risk Indicator Code1,Risk Indicator Code2,...>   |
| <b>Algorithm<br/>Entity</b> | Case Type -Enter Status   |
| <b>Program<br/>Type</b>     | Java  |
| <b>Program<br/>Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.algorithm.RiskIndicatorSetResetEnterProcessing   |
| <b>Parameters</b>           | <b>Name:</b> Risk Indicator<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Risk Indicator<br><br><b>Name:</b> Risk Indicator Code<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Risk Indicator Code  |
| <b>Detailed<br/>Design</b>  | This Algorithm Set or Reset the Account level Warning Indicators of all the associated accounts of Bankruptcy.<br>Note this will exclude the Charge-Off Accounts. ( Based on RECOVERY_SW in CI_ACCT_EXTN table)<br>Risk Indicator Codes should be comma separated.<br>Values:<br>Risk Indicator = SET or RESET<br>Risk Indicator Code = <Risk Indicator Code1,Risk Indicator Code2,...> |

## 5.42 Task - Automatic Allocation of tasks to Vendors

This table provides details of the Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing C1-TSKVNDR algorithm.

**Table 5–95 Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing C1-TSKVNDR**

|                                 |   |
|---------------------------------|---|
| <b>Description</b>              | Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing   |
| <b>Detailed<br/>Description</b> | On creation of task check if task is already allocated to a member.<br>If Yes no action required. If No allocate the case to the member with lowest number of tasks of that task type in the queue. |
| <b>Algorithm</b>                | TO DO Type - Post Processing  |

|                        |   |
|------------------------|---|
| <b>Entity</b>          |   |
| <b>Program Type</b>    | Java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.vendor.VendorManagementAutomaticTaskAllocation  |
| <b>Parameters</b>      | NA  |
| <b>Detailed Design</b> | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook. |

## 5.43 Hardship - Associate Accounts of Main Customer

This table provides details of the Hardship - Associate Accounts of Main Customer - Enter Processing C1-HARASOPND algorithm.

**Table 5–96 Hardship - Associate Accounts of Main Customer - Enter Processing C1-HARASOPND**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Hardship Entity Association Pending State - Enter Processing  |
| <b>Detailed Description</b> | This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which has logic for Hardship Entity Association. |
| <b>Algorithm Entity</b>     | Case Type -Enter Processing   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipAssociation   |
| <b>Parameters</b>           | NA  |
| <b>Detailed Design</b>      | This is a reference implementation Enter Processing algorithm. Customization team can utilize this hook.  |

## 5.44 Early Collection

This table provides details of the Transition to Contact Statuses - Monitoring C1-ECIC

**Table 5–97 Transition to Contact Statuses - Monitoring C1-ECIC**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to Contact Statuses - Monitoring  |
| <b>Detailed Description</b> | <p>This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which transition to contact state if First Contact Date has reached.</p> <p>Transition to contact state if First Contact Date has reached:</p> <p>If First Contact Date has reached (based on the parameters below) or</p> <p>Account is Direct Debit and Immediate Transition if Direct Debit = Yes/No</p> <p>Transition to Contact RM status if Relationship Manager exists and Contact RM status has been specified .</p> <p>Transition to Contact Alternate status if Contact Alternate Flag = Y and Contact</p> |

|                         |   |
|-------------------------|---|
|                         | <p>Alternate Status has been specified<br/>Else<br/>Transition to Contact Status<br/>Set Re-Allocation Switch = Y for the case post case transition<br/>Possible Values<br/>First Contact Calculation Parameter: DPD, DIA, Days Since Case Start<br/>Immediate Transition if Direct Debit: Y,N<br/>Validation Date : POSTINGDATE, SYSTEMDATE</p>  |
| <b>Algorithm Entity</b> | Case Type-Auto Transition   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateContact   |
| <b>Parameters</b>       | <p><b>Name:</b> First Contact Calculation Parameter<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> First Contact Calculation Parameter</p> <p><b>Name:</b> Number Of Days For First Contact<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Number Of Days For First Contact</p> <p><b>Name:</b> Contact RM Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact RM Status</p> <p><b>Name:</b> Contact Alternate Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Alternate Status</p> <p><b>Name:</b> Contact Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Status</p> <p><b>Name:</b> Immediate Transition if Direct Debit: Yes/No<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Immediate Transition if Direct Debit: Yes/No</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>  | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

**Table 5–98 Park Small Balance Accounts - Monitoring C1-ECPSBA**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Park accounts with small balances to a separate status - Monitoring  |
| <b>Detailed Description</b> | <p>This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which has logic for Park Small Balance Accounts.</p> <p>Park accounts with small balances to a separate status so that no contacts are initiated for the account.</p> <p>If Net Arrear Amount &lt;= Small Balance Threshold<br/>And Net Arrear Amount &gt; 0<br/>Then transition to small balance status.<br/>Net Arrear Amount = (Overdue Amount - Unclear Amount)<br/>If Use Overdue Amount = Yes then use Overdue Amount instead of Net Arrear Amount in the calculations.<br/>Set Re-Allocation Switch = Y for the case post case transition.<br/>Possible Values :<br/>Use Overdue Amount : Y,N</p> |
| <b>Algorithm Entity</b>     | Case Type-Auto Transitions   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.ParkSmallBalanceAccounts   |
| <b>Parameters</b>           | <p><b>Name:</b> Small Balance Threshold<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Small Balance Threshold</p> <p><b>Name:</b> Small Balance Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Small Balance Status</p> <p><b>Name:</b> Use Overdue Amount<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Use Overdue Amount</p>   |
| <b>Detailed Design</b>      | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.   |

**Table 5–99 Initiate Skip Tracking - No Telephone Number- Enter Processing C1-ECISTNTN**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to skip tracking status if no telephone number exists for any of the account holder - Enter Processing  |
| <b>Detailed Description</b> | <p>If no contact points exists then move the case to Skip Tracing status</p> <p>Check if one of the Contact Points as specified in the parameters exists for any of the account holder.</p> <p>If no contact point exists than move the case to Skip Tracing Status.</p> |



|                         |   |
|-------------------------|---|
|                         | Set Re-Allocation Switch = Y for the case post case transition.   |
| <b>Algorithm Entity</b> | Case Type-Enter Status  |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracing   |
| <b>Parameters</b>       | <p><b>Name:</b> Contact Points<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Points</p> <p><b>Name:</b> Skip Tracing Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Skip Tracing Status</p> |
| <b>Detailed Design</b>  | This is a reference implementation Enter Processing algorithm. Customization team can utilize this hook.  |

**Table 5–100 Initiate Skip Tracking - No Telephone Number- Monitoring C1-ECTTSS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to Suspended status based on Account and Party Risk Indicators - Monitoring   |
| <b>Detailed Description</b> | <p>If the Account has one of the Account Risk Indicators specified in the parameter Transition to Suspended status.<br/> Create a task if Task Type has been mentioned and assign it to the Specified Queue<br/> Set Re-Allocation Switch = Y for the case post case transition.<br/> Set Suspend Reason = Risk Indicator<br/> Exit.</p> <p>If either of the financial owners have one of the Party Indicators mentioned in the parameter than transition to Suspended status.<br/> Create a task if Task Type has been mentioned and assign it to the Specified Queue<br/> Set Re-Allocation Switch = Y for the case post case transition.<br/> Set Suspend Reason = Risk Indicator<br/> Exit.</p> <p>If there is at least one financial owner with no Risk indicators mentioned in the parameter 'Party Risk Indicators - Contact Alternate' than transition the case to the Contact Alternate Status.<br/> Create a task if Task Type has been mentioned and assign it to the Specified Queue.<br/> Set Re-Allocation Switch = Y for the case post case transition.<br/> Set Alternate Contact Flag = Y<br/> Set Alternate Contact Reason = Risk Indicator<br/> If case already in Contact Alternate status don't initiate any activities.<br/> Exit.</p> |
| <b>Algorithm Entity</b>     | Case Type-Auto Transition  |
| <b>Program Type</b>         | Java   |

|                           |   |
|---------------------------|---|
| <b>Program Name</b>       | com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToSuspendedStatus   |
| <b>Parameters</b>         | <p><b>Name:</b> Account Warning Indicators<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Account Warning Indicators</p> <p><b>Name:</b> Party Warning Indicators<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Party Warning Indicators</p> <p><b>Name:</b> Party Level Risk Code (ConatctAlternate)<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Party Level Risk Code (ConatctAlternate)</p> <p><b>Name:</b> Contact Alternate Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Alternate Status</p> <p><b>Name:</b> Suspended Status<br/><b>Required (Yes/No):</b><br/><b>Description:</b> Suspended Status</p> <p><b>Name:</b> Task Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Queue</p> |
| <b>Parameters (Cont.)</b> | <p><b>Name:</b> Characteristics Type Suspend Reason<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Characteristics Type Suspend Reason</p> <p><b>Name:</b> Characteristics Type Alternate contact Reason<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Characteristics Type Alternate contact Reason</p>   |
| <b>Detailed Design</b>    | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

Table 5–101 Validate Contact Cap- Monitoring C1-ECVCC

|                    |   |
|--------------------|---|
| <b>Description</b> | Check if the contact cap has reached for the case<br>If case is not already on Hold and Display Date <= Business Date |
|--------------------|---|

|                             |   |
|-----------------------------|---|
|                             | <p>And the number of successful contacts linked to the case in last X number of days &gt;= Contact Cap</p> <p>Hold the case for Y number of days with the given Hold Reason.</p> <p>Logic for considering successful contacts: All contacts with given contact methods that have Authentication Status = Green</p> <p>Possible Values for Validation Date {SYSTEMDATE,POSTINGDATE}</p>  |
| <b>Detailed Description</b> | This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which have logic for Validate Contact Cap.   |
| <b>Algorithm Entity</b>     | Case Type-Auto Transitions  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.ValidateContactCap  |
| <b>Parameters</b>           | <p><b>Name:</b> Contact Cap<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Cap</p> <p><b>Name:</b> Contact Cap Duration (X)<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Cap Duration (X)</p> <p><b>Name:</b> Contact Hold Days (Y)<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Hold Days (Y)</p> <p><b>Name:</b> Contact Methods<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Methods</p> <p><b>Name:</b> Hold Reason<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Hold Reason</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>      | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

Table 5–102 Schedule Contact - Monitoring C1-ECSC

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Schedule Contact - Monitoring  |
| <b>Detailed Description</b> | Schedule Contact for the case as per intensity<br>If case is not on Hold |

|                         |  |
|-------------------------|--|
|                         | <p>And Display Date &lt;= Business Date or Display Date is Blank<br/> Set Display Date = Max((Last Successful Contact Date + Contact Intensity), Business Date)<br/> Consider Contact Intensity from Algorithm parameter if specified else picks up Contact Intensity from case level field.<br/> Logic for considering successful contacts: Last contact with given contact methods that have Authentication Status = Green<br/> Validation Date can be POSTINGDATE or SYSTEMDATE</p> |
| <b>Algorithm Entity</b> | Case Type-Auto Transition  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.ScheduleContact  |
| <b>Parameters</b>       | <p><b>Name:</b> Contact Intensity<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Intensity</p> <p><b>Name:</b> Contact Methods<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Methods</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Validation Date</p>   |
| <b>Detailed Design</b>  | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.   |

**Table 5–103 Initiate Skip Tracing - Wrong Telephone Number- Monitoring C1-ECISTITN**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Initiate Skip Tracing - Wrong Telephone Number- Monitoring   |
| <b>Detailed Description</b> | <p>Transition to skip review if 'X' number of consecutive failed contacts</p> <ul style="list-style-type: none"> <li>■ If last X number of consecutive contacts has been unsuccessful, transition to Skip Tracing Status.</li> </ul> <p>Logic for considering unsuccessful contacts: If last X consecutive contacts with given contact methods have Authentication Status other than 'Green'<br/> Set Re-Allocation Switch = Y for the case post case transition<br/> Possible Values for Validation Date are POSTINGDATE and SYSTEMDATE</p> |
| <b>Algorithm Entity</b>     | Case Type-Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracingInvalidTelNumber  |

|                        |   |
|------------------------|---|
| <b>Parameter<br/>s</b> | <b>Name:</b> Consecutive Failed Contacts (X)<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Consecutive Failed Contacts (X) |
|                        | <b>Name:</b> Skip Tracing Status<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Skip Tracing Status                         |
|                        | <b>Name:</b> Contact Methods<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Contact Methods                                 |
|                        | <b>Name:</b> Validation Date<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Validation Date                                |
| <b>Detailed Design</b> | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.                                  |

**Table 5–104 Transition to Under Resolution Status- Monitoring C1-ECTURS**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Transition to under resolution status.  |
| <b>Detailed Description</b> | <p>Transition to under resolution status if Net Arrear Amount &lt;=0</p> <ul style="list-style-type: none"> <li>■ Transition the case to Under Resolution Status if Net Arrear Amount &lt;= 0 or PTP is running on the account.</li> <li>■ Set Re-Allocation Switch = Y for the case post case transition</li> </ul> <p>Net Arrear Amount = (Overdue Amount - Unclear Amount)<br/> If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the calculations.<br/> Possible values:<br/> Use Overdue Amount: Y,N</p> |
| <b>Algorithm Entity</b>     | Case Type-Auto Transitions  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToUnderResolution Status  |
| <b>Parameter<br/>s</b>      | <b>Name:</b> Under Resolution Status<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Under Resolution Status<br><br><b>Name:</b> Use Overdue Amount  |

|                        |  |
|------------------------|--|
|                        | <b>Required (Yes/No):</b> No<br><b>Description:</b> Use Overdue Amount                             |
| <b>Detailed Design</b> | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook. |

**Table 5–105 Resume Contact From Under Resolution- Monitoring C1-RCFR**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Resume Contact From Under Resolution- Monitoring   |
| <b>Detailed Description</b> | <ol style="list-style-type: none"> <li>1. If Active Settlement Status='N'<br/>If there is no active PTP on the account and<br/>If Last PTP Type &lt;&gt; Final Settlement Type and<br/>If the Net Outstanding Amount &gt; 0<br/>Then transition the case to Contact Alternate Status If Contact Alternate Flag = Y<br/>Else move to Contact Status<br/><br/>If Last PTP Type = Final Settlement Type and<br/>If Last PTP Status = 'Kept' and<br/>If the Net Outstanding Amount &gt; 0<br/>Then transition the case to Supervisor Review Status<br/>Else transition the case to Contact Alternate Status If Contact Alternate Flag = Y<br/>Else move to Contact Status</li> <li>2. If Active Settlement Status='Y'<br/>If Last PTP Type='Final Settlement Type' and if Last PTP status='Kept' then<br/>transition the Case to 'Settled'. Mark account for Write off as 'Y' and set Write Off<br/>reason code as given in the soft parameter.<br/>Else if Last PTP status='Broken'<br/>then transition the case to 'Contact/Contact Alternate' based on contact alternate flag<br/>Change Settlement status to Settlement Broken<br/>set Active Settlement Status = N<br/>If Use Outstanding Amount = Yes then use Outstanding Amount instead of Net<br/>Outstanding Amount in the calculations.<br/>Net Outstanding Amount = (Outstanding Amount - Unclear Amount)<br/>Set Re-Allocation Switch = Y for the case post case transition</li> </ol> <p>Possible Values of parameters are as given below:</p> <ol style="list-style-type: none"> <li>1. Contact Status - Value will be Contact Status Code</li> <li>2. Contact Alternate status - Value will be Contact Alternate status code</li> <li>3. Supervisor Review status - Value will be Supervisor Review status code</li> <li>4. Use Outstanding Amount(Y/N) - Value will be Y/N</li> <li>5. Settled Status -Value of Settled Status Code</li> <li>6. Write Off Reason Code -Value will be Write Off Reason Code</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type-Auto Transition  |

|                        |   |
|------------------------|---|
| <b>Program Type</b>    | Java  |
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactFromUnderResolution  |
| <b>Parameters</b>      | <p><b>Name:</b> Contact Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Status</p> <p><b>Name:</b> Contact Alternate Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Alternate Status</p> <p><b>Name:</b> Supervisor Review Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Supervisor Review Status</p> <p><b>Name:</b> Use Outstanding Amount<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Use Outstanding Amount</p> <p><b>Name:</b> Settled Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Settled Status</p> <p><b>Name:</b> Write Off Reason Code<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Write Off Reason Code</p> |
| <b>Detailed Design</b> | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

**Table 5–106 Resume Contact from Small Balance- Monitoring C1-ECRCB**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Resume Contact from Small Balance- Monitoring  |
| <b>Detailed Description</b> | <p>.This algorithm is used to resume contact from small balance status.<br/>If Net Arrear Amount &gt; Small Balance Threshold<br/>Then transition the case to<br/>Contact RM Status if RM exists and Contact RM status has been configured<br/>Contact Alternate Status If Contact Alternate Flag = Y<br/>Else Contact Status<br/>Set Re-Allocation Switch = Y for the case post case transition<br/>If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the calculations.</p> |

|                         |   |
|-------------------------|---|
|                         | Net Arrear Amount = (Overdue Amount - Unclear Amount)<br>Possible Value:<br>Overdue Amount : Y,N  |
| <b>Algorithm Entity</b> | Case Type-Auto Transition   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactfromSmallBalance   |
| <b>Parameters</b>       | <p><b>Name:</b> Contact Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Status</p> <p><b>Name:</b> Contact RM Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact RM Status</p> <p><b>Name:</b> Contact Alternate Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Alternate Status</p> <p><b>Name:</b> Use Overdue Amount<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Use Overdue Amount</p> <p><b>Name:</b> Small Balance Threshold<br/><b>(Yes/No):</b> No<br/><b>Description:</b> Small Balance Threshold</p> |
| <b>Detailed Design</b>  | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

**Table 5–107 Determine Contact Intensity - Monitoring C1-ECDCI**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Determine Contact Intensity and Contact Intensity Review Date -Monitoring  |
| <b>Detailed Description</b> | This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which have logic for Determine Contact Intensity. |
| <b>Algorithm Entity</b>     | Case Type-Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.DetermineContactIntensity  |



|                        |   |
|------------------------|---|
| <b>Parameters</b>      | <p><b>Name:</b> Contact Intensity Rule<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Intensity Rule</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b> | This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.  |

**Table 5–108 Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing C1-CTRANTCRE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing   |
| <b>Detailed Description</b> | <p>Generic Result Post Processing Algorithm for Case Transition and Task Creation.</p> <p>Transition the case to given Case Status if Case Status is configured and the current status is present in one of the Valid Current Statuses. Display an error 'The selected result &lt;Result Type&gt; is not allowed in current Status.' If the current status is not present in one of the valid status.</p> <ul style="list-style-type: none"> <li>■ Create Task of given Task Type and assign it to the give Task Queue if Task Type is configured.</li> <li>■ Map the created task with the Follow up Id of the Follow Up that created the task.</li> <li>■ Set Re-Allocation Switch = Y if Re-Allocate = Y</li> <li>■ Copy the common characteristics of result into the case. (here the char codes need to be maintained at both the result type and case type level)</li> <li>■ Task Creation Logic:<br/>       If Task For = Account<br/>       Create Task on the primary associated account on the case<br/>       If Task For = Customer<br/>       Create Task on the primary associated customer of the case<br/>       If Task For = Case<br/>       Create Task on the case<br/>       If Task For = Admin<br/>       Create Admin level Task<br/>       Note: Task For is the mandatory characteristic at Task Level<br/>       Task For = Customer is an invalid configuration for Account level Case and vice versa</li> </ul> <p>Possible Values of Re-Allocate Switch and Copy Characteristics to Case are : Y/N<br/>       "Event Name" and "Action Flag" fields are introduced to update Cease_Desist\Contact_Alternate\Dispute Flags, where:-<br/>       "Event Name" will be provided depending on the FLAG which you need to update.<br/>       So, it can have one of the values:-<br/>       Event Name :- "CEASE_DESIST"<br/>       Event Name :- "CONTACT_ALT"</p> |

|                         |  |
|-------------------------|--|
|                         | <p>Event Name :- "DISPUTE"<br/> And<br/> "Action Flag" value will be SET\RESET.<br/> To set Cease_Desist\Contact_Alternate\Dispute Flags to "Y", provide Action Flag :- "SET".<br/> To set Cease_Desist\Contact_Alternate\Dispute Flags "N", provide Action Flag :- "RESET".<br/> To transit case to other status from contact alternate status and remove contact alternate details set Contact Alternate Status to contact alternate case status.<br/> Contact Alternate Case Status soft parameter usage is as follow :<br/> When user want to stop contact alternate details on an account in that scenario :</p> <ul style="list-style-type: none"> <li>■ If current case status is other than status provided in soft parameter "Contact Alternate Case Status" then Do not transition case and set contact alternate flag to No and delete contact alternate details on the account</li> <li>■ And if current case status is same as the case status code provided in soft parameter "Contact Alternate Case Status" then perform case transition to the status provided in soft parameter "Case Status" and set contact alternate flag to No and delete contact alternate details on the account.</li> </ul> |
| <b>Algorithm Entity</b> | Result Type - Post Processing  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseTransitionandTraskCreationPostProcessingAlgo   |
| <b>Parameters</b>       | <p><b>Name:</b> Case Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Case Status</p> <p><b>Name:</b> Valid Current Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Valid Current Status</p> <p><b>Name:</b> Task Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Task Type</p> <p><b>Name:</b> Queue<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Queue</p> <p><b>Name:</b> Re-Allocate Switch<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Re-Allocate Switch</p>   |

|                           |   |
|---------------------------|---|
|                           | <p><b>Name:</b> Copy Characteristics to Case<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Copy Characteristics to Case</p> <p><b>Name:</b> Event Name<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Event Name</p>     |
| <b>Parameters (Cont.)</b> | <p><b>Name:</b> Action Flag<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Action Flag</p> <p><b>Name:</b> Contact Alternate Case Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Alternate Case Status</p> |
| <b>Detailed Design</b>    | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.   |

**Table 5–109 Refer to Supervisor- Result Type - Post Processing C1-ECRTS**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Refer to Supervisor - Result Type - Post Processing   |
| <b>Detailed Description</b> | <p>Supervisor Referral Algorithm</p> <ul style="list-style-type: none"> <li>■ If case is present in one of the status's specified in 'Valid Current Status' than<br/> Proceed with further actions<br/> Else<br/> Display an error 'The selected result &lt;Result Type&gt; is not allowed in current Status.'<br/> And don't proceed with further actions.</li> <li>■ Transition the case to given Case Status</li> <li>■ Create Task of given Task Type and assign it to the Supervisor Queue (Queue of Task) of the Case Queue</li> <li>■ Map the created task with the Follow up Id of the Follow Up that created the task.</li> <li>■ Set Re-Allocation Switch = Y if Re-Allocate = Y<br/> Re-Allocate can be Y/N</li> </ul> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         |   |
| <b>Parameters</b>           | <p><b>Name:</b> Valid Current Status<br/> <b>Required (Yes/No):</b> No</p>  |

|                        |  |
|------------------------|--|
|                        | <p><b>Description:</b> Valid Current Status</p> <p><b>Name:</b> Case Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Case Status</p> <p><b>Name:</b> Task Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Re-Allocate<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Re-Allocate</p> |
| <b>Detailed Design</b> | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.  |

**Table 5–110 Resume Collections- Result Type - Post Processing C1-RESCOLL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Resume Collections- Result Type - Post Processing  |
| <b>Detailed Description</b> | <p>This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which have logic for Resume Collections.</p> <p>Transition the case to</p> <p>Contact RM Status if RM exists and Contact RM status has been configured</p> <p>Contact Alternate Status If Contact Alternate Flag = Y</p> <p>Else Contact Status</p> <p>Set Re-Allocation Switch = Yes if Re-Allocate = Y</p> <p>Re-Allocate can be Y/N</p> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeCollectionsPostProcessingAlgo  |
| <b>Parameters</b>           | <p><b>Name:</b> Contact Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact Status</p> <p><b>Name:</b> Contact RM Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Contact RM Status</p> <p><b>Name:</b> Contact Alternate Status</p>   |

|                        |   |
|------------------------|---|
|                        | <b>Required (Yes/No):</b> No<br><b>Description:</b> Contact Alternate Status<br><br><b>Name:</b> Re-Allocate<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Re-Allocate |
| <b>Detailed Design</b> | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.   |

**Table 5–111 Create case on Follow up- Result Type - Post Processing C1-CRETCSFL**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Create case on Follow up - Post Processing  |
| <b>Detailed Description</b> | <p>Create Required Case on Follow Up</p> <p>If Account Level Case Type creates case on account, If Customer level Case Type creates case on the main customer of the account.</p> <p>Queue to which the case should be allocated if provided else the case should remain unallocated with Re-Allocation Switch as Y</p> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseCreationonFollowupPostProcessingAlgo  |
| <b>Parameters</b>           | <b>Name:</b> Case Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Type<br><br><b>Name:</b> Queue Type<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Queue Type   |
| <b>Detailed Design</b>      | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.   |

**Table 5–112 Hold Case- Result Type - Post Processing C1-HOLDCASE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Hold Case - Post Processing   |
| <b>Detailed Description</b> | <p>Hold Case for Days as provided in Characteristic Type provided in Hold Period or if that is blank Hold Period should be referred from Hold Period parameter.</p> <p>And</p> <p>Hold Reason should be set as provided in characteristic type provided in Hold Reason or if that is blank Hold Reason should be referred from Hold Reason parameter.</p> <p>Validation Date can be SYSTEMDATE or POSTINGDATE</p> |

|                         |   |
|-------------------------|---|
| <b>Algorithm Entity</b> | Result Type - Post Processing   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.HoldCasePostProcessingAlgo  |
| <b>Parameters</b>       | <p><b>Name:</b> Hold Period Characteristic Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Hold Period Characteristic Type</p> <p><b>Name:</b> Hold Period<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Hold Period</p> <p><b>Name:</b> Hold Reason Characteristic Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Hold Reason Characteristic Type</p> <p><b>Name:</b> Hold Reason<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Hold Reason</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>  | This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.   |

**Table 5–113 Set Case Data- Result Type - Enter Processing C1-ECUPCASE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Case Level Data when a case enters a new status - Enter Processing   |
| <b>Detailed Description</b> | <p>Set Case Characteristics to specific values provided in algorithm parameters. On entering the value the corresponding characteristic validation algorithm should be triggered.</p> <p>If type is mentioned but value is not than the char type needs to be made blank.</p> |
| <b>Algorithm Entity</b>     | Case Status - Enter Processing  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateCaseData  |
| <b>Parameters</b>           | <p><b>Name:</b> Char Type - 1<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Char Type - 1</p>  |

|                        |  |
|------------------------|--|
|                        | <p><b>Name:</b> Char Type - 2<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Type - 2</p> <p><b>Name:</b> Char Type - 3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Type - 3</p> <p><b>Name:</b> Char Type - 4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Type - 4</p> <p><b>Name:</b> Char Type - 5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Type - 5</p> <p><b>Name:</b> Char Value - 1<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Value - 1</p> <p><b>Name:</b> Char Value - 2<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Value - 2</p> <p><b>Name:</b> Char Value - 3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Value - 3</p> <p><b>Name:</b> Char Value - 4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Value - 4</p> <p><b>Name:</b> Char Value - 5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Char Value - 5</p> |
| <b>Detailed Design</b> | This is a reference implementation of Pre Processing algorithm. Customization team can utilize this hook.  |

**Table 5–114** This algorithm will transition the case status to the Suspension status if Cease and Desist = Y C1-CSETRANS

|                    |   |
|--------------------|---|
| <b>Description</b> | This algorithm will transition the case status to the Suspension status if Cease and Desist = Y |
|--------------------|---|

|                             |   |
|-----------------------------|---|
| <b>Detailed Description</b> | Additional algorithm in Pending Status: Enter Processing to transition to Suspend Status if Cease and Desist = Y. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.CeaseDesistAccountSuspension                            |
| <b>Parameters</b>           | <b>Name:</b> Suspension State<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Suspension State            |
| <b>Detailed Design</b>      | Additional algorithm in Pending Status: Enter Processing to transition to Suspend Status if Cease and Desist = Y. |
| <b>Detailed Design</b>      | This is a reference implementation of Pre Processing algorithm. Customization team can utilize this hook.         |

**Table 5–115 Algorithm is used for scheduling call C1-SCHCALL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm is used for scheduling call  |
| <b>Detailed Description</b> | <p>This algorithm is used to fulfil request by customer to collector for calling at specific time.</p> <ul style="list-style-type: none"> <li>■ The Call Back Time will get saved as the Next Action Time on the case. If NA is selected the value will go as blank.</li> <li>■ If the Next Action Date is same as Current date and Online Dialer Inclusion = Yes then add/update the record in the Dialer extract using the Dialer Inclusion Service. The Dialer Extract Status will be set as 10.</li> </ul> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.algorithms.ScheduleCallPostProcessingAlgorithm   |
| <b>Parameters</b>           | <b>Name:</b> Online Dialer Inclusion<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Online Dialer Inclusion<br><br><b>Name:</b> Preferred Time Char<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Preferred Time Char<br><br><b>Name:</b> Validation Date<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Validation Date  |



|                        |  |
|------------------------|--|
| <b>Detailed Design</b> | <p>This algorithm is used to fulfil request by customer to collector for calling at specific time.</p> <ul style="list-style-type: none"> <li>■ The Call Back Time will get saved as the Next Action Time on the case. If NA is selected the value will go as blank.</li> <li>■ If the Next Action Date is same as Current date and Online Dialer Inclusion = Yes then add/update the record in the Dialer extract using the Dialer Inclusion Service. The Dialer Extract Status will be set as 10.</li> </ul> |
|------------------------|--|

Table 5–116 Reset WI in the host C1-RESETWISHCALL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Reset WI in the host   |
| <b>Detailed Description</b> | <p>.This algorithm resets WI in the host.</p> <ul style="list-style-type: none"> <li>■ Call the Host Account Warning Indicator Service to set the WI mentioned in the parameter</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type – Exit Status  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collectionevt.ResetCaseWarningIndOnHost   |
| <b>Parameters</b>           | <p><b>Name:</b> Account Warning Indicator<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Account Warning Indicator</p>  |
| <b>Detailed Design</b>      | <p>This algorithm resets WI in the host.</p> <ul style="list-style-type: none"> <li>■ Call the Host Account Warning Indicator Service to set the WI mentioned in the parameter</li> </ul>  |

## 5.45 Asset Repossession

Table 5–117 Validate Collateral - Enter Validation C1-VALDCOLL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validate Collateral - Enter Status Validation  |
| <b>Detailed Description</b> | <p>The input collateral is associated with the account on which the repossession case is being created.</p> <p>The collateral belongs to the collateral type and collateral category specified in the parameters. If collateral type and collateral category are not mentioned no validation will be done.</p> <p>The collateral status is not 'Sold'. Date of Sale is blank.</p> <p>Error Message: "Repossession cannot be initiated on the collateral. Please check if The collateral type is supported in the repossession process.</p> |

|                         |   |
|-------------------------|---|
|                         | The collateral in not already sold or under repossession."<br>Collateral Type: It should be allow any collateral type as a parameter. (For eg: AUTOMOBILE,PROPERTY etc)<br>Collateral Category :Category of collateral(For eg: Vehicle)<br>Both parameter accept comma separated values. CI_COLLATERAL table have both columns. |
| <b>Algorithm Entity</b> | Case Status - Enter Status Validation   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | om.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateral   |
| <b>Parameters</b>       | <p><b>Name:</b> Collateral Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Collateral Type</p> <p><b>Name:</b> Collateral Category<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Collateral Category</p>   |
| <b>Detailed Design</b>  | Verify that the collateral code provided as input is associated with the account and has not been sold already. The collateral belongs to one of the collateral type supported by the process.  |

**Table 5–118 Validate Demand Letter and Acceleration Letter - Enter Validation C1-VALIDDLAL**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate if Demand Letter and Acceleration letter have been sent - Enter Status Validation  |
| <b>Detailed Description</b> | <p>If DL Template Code has been mentioned validate if Demand Letter has been sent in last X days.</p> <p>If AL Template Code has been mentioned validate if Acceleration Letter has been sent in last X days.</p> <p>If X Days is not specified just check if the letters have been sent on the account.</p> <p>Checks will be done for all associated accounts unless 'Only Primary Account = Yes' in which case the check will be only on primary associated account.</p> <p>Parameter Description as follows :</p> <ol style="list-style-type: none"> <li>1. Demand Letter Template Code - Demand Letter Template Code</li> <li>2. Acceleration Letter Template Code - Acceleration Letter Template Code</li> <li>3. Number Of Days in which Demand Letter or Acceleration Letter send - number of days</li> <li>4. Only PrimaryAccount Switch - This switch determines weather letters defined in 1 and 2 should be fetched from primary customer only or all associated customers. Possible values are "Y", "N", true, false</li> <li>5. Validation Date - This parameter determines letter sent date should be calculated with reference to posting date or system date. Possible values are "SYSTEM DATE" and</li> </ol> |

|                         |  |
|-------------------------|--|
|                         | "POSTING DATE"   |
| <b>Algorithm Entity</b> | Case Status - Enter Status Validation  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ValidateDemandLetterandAccelerationLetter   |
| <b>Parameters</b>       | <p><b>Name:</b> Demand Letter Template Code<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Demand Letter Template Code</p> <p><b>CodeName:</b> Acceleration Letter Template Code<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Acceleration Letter Template Code</p> <p><b>Name:</b> Number Of Days in which Demand Letter or Acceleration Letter send<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Number Of Days in which Demand Letter or Acceleration Letter send</p> <p><b>Name:</b> Only PrimaryAccount Switch<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Only PrimaryAccount Switch</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>  | Validate if Demand Letter and Acceleration letter have been sent   |

**Table 5–119 Associate Customers in Repossession Case - Enter Validation C1-ASSOCUST**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Associate Customers in Repossession Case - Enter Status                              |
| <b>Detailed Description</b> | Associate all financial owners on the associated accounts to the Repossession case.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program</b>              | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AssociateC |

|                        |   |
|------------------------|---|
| <b>Name</b>            | ustAssRepo  |
| <b>Parameters</b>      | No Parameters   |
| <b>Detailed Design</b> | Associate all financial owners on the associated accounts to the Repossession case. |

**Table 5–120 Bankruptcy Check on Associate Customers - Enter Status C1-CHKBKPTCY**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Verify if any of the customer associated with the case has claimed Bankruptcy - Enter Status  |
| <b>Detailed Description</b> | <p>If Repossession Reason &lt;&gt; Bankruptcy<br/> For each customer associated with the case<br/> Check if the Bankruptcy_Switch = Y. If yes Case Creation will be rolled back and below error message will be displayed.<br/> "One or more of the collateral owners have claimed Bankruptcy. Repossession process should be initiated from Bankruptcy process"<br/> Repossession Reason for Bankruptcy: Possible reason for bankruptcy, Comma separated values can be pass. (Repo reasons available into REPO_REASON look up)</p> |
| <b>Algorithm Entity</b>     | Case Status - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ChkBkpcyOnAssociateCust   |
| <b>Parameters</b>           | Name: Repossession Reason for Bankruptcy<br>Required (Yes/No): Yes<br>Description: Repossession Reason for Bankruptcy   |
| <b>Detailed Design</b>      | Verify if any of the customer associated with the case has claimed Bankruptcy.  |

**Table 5–121 Monitor if Demand letter and Acceleration letter have been sent on the account. C1-MNTRDLAL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Monitor if Demand letter and Acceleration letter have been sent on the account.  |
| <b>Detailed Description</b> | <p>If DL Template Code has been mentioned validate if Demand letter has been sent and current date &gt; Demand Letter Expiry Date.<br/> If AL Template Code has been mentioned validate if Acceleration letter has been sent and the current date &gt; Acceleration letter Expiry Date.<br/> If Only Primary Account = Yes then the above checks need to be done only on Primary account else the checks should be done on all associated accounts.<br/> If both are true transition the case to Repossession Referred Status.<br/> Parameter Description as follows :</p> |

|                         |   |
|-------------------------|---|
|                         | <ol style="list-style-type: none"> <li>1. Demand Letter Template Code - Demand Letter Template Code</li> <li>2. Acceleration Letter Template Code - Acceleration Letter Template Code</li> <li>3. Repossession Referred Status - Repossession referred status code</li> <li>4. Primary Account Switch - This switch determines weather letters defined in 1 and 2 should be fetched from primary customer only or all associated customers. Possible values are Y,N,true,false</li> <li>5. Validation Date - This parameter determines letter sent date should be calculated with reference to posting date or system date. Possible values are SYSTEM DATE and POSTING DATE</li> </ol> |
| <b>Algorithm Entity</b> | Case Type - Auto Transition   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.MonitorDemandLetterandAccelerationLetterExpiry   |
| <b>Parameters</b>       | <p><b>Name:</b> Demand Letter Template Code<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Demand Letter Template Code</p> <p><b>Name:</b> Acceleration Template Code<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Acceleration Template Code</p> <p><b>Name:</b> Repossession Referred Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Repossession Referred Status</p> <p><b>Name:</b> Primary Account Sw<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Primary Account Sw</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Validation Date</p>  |
| <b>Detailed Design</b>  | Monitor if Demand letter and Acceleration letter have been sent on the account.   |

Table 5–122 Auto Approval Check for Repossession C1-REPOAPRV

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Auto Approval Check for Repossession  |
| <b>Detailed Description</b> | <p>If the Auto- Approval Rule returns true the case will be transitioned to the Approved status.<br/> If the Auto Approval Rule returns false the case will remain in the Repossession Referred Status and a Task is created for the given Task Type and is assigned to the supervisor of the queue.</p> <p>Below facts are used:</p> <ul style="list-style-type: none"> <li>■ Collateral Type</li> <li>■ Collateral Category</li> <li>■ Repossession Reason</li> <li>■ Outstanding Amount</li> <li>■ Overdue Amount</li> <li>■ Days Past Due</li> <li>■ Last Payment Date</li> <li>■ Last Payment Amount</li> <li>■ Estimated Realization Amount</li> <li>■ Deficiency Balance</li> <li>■ Number of accounts associated with the collateral</li> </ul> |
| <b>Algorithm Entity</b>     | Case Status - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutoApprovalCheckforRepossession   |
| <b>Parameters</b>           | <p><b>Name:</b> Auto Approval Rule<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Auto Approval Rule</p> <p><b>Name:</b> Approved Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Approved Status</p> <p><b>Name:</b> Task Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Task Type</p> <p><b>Name:</b> Queue</p>   |

|                        |  |
|------------------------|--|
|                        | <b>Required (Yes/No):</b> Yes<br><b>Description:</b> Queue   |
| <b>Detailed Design</b> | <p>If the Auto- Approval Rule returns true the case will be transitioned to the Approved status.</p> <p>If the Auto Approval Rule returns false the case will remain in the Repossession Referred Status and a Task is created for the given Task Type and is assigned to the supervisor of the queue.</p> |

**Table 5–123 Repossession Setup Complete C1-RSTUPCMPL**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Repossession Setup Complete   |
| <b>Detailed Description</b> | If Repossession Reason = Voluntary Repossession transition to Repossession In Progress - Voluntary Surrender else transition to Repossession in Progress  |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepossessionTransition  |
| <b>Parameters</b>           | <p><b>Name:</b> Voluntary Repossession Reason<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Voluntary Repossession Reason</p> <p><b>Name:</b> Voluntary Repossession Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Voluntary Repossession Status</p> <p><b>Name:</b> Normal Repossession Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Normal Repossession Status</p> |
| <b>Detailed Design</b>      | If Repossession Reason = Voluntary Repossession transition to Repossession In Progress - Voluntary Surrender else transition to Repossession in Progress  |

**Table 5–124 Automatic task creation for vendors C1-AUTOTASKC**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Automatic task creation for vendors   |
| <b>Detailed Description</b> | Create a Task of given Task Type and assign it to the queue code specified in the parameter. Additionally assign the task to the vendor defined against the service type for the case. If the vendor is not allocated to the Queue code or if there is no vendor assigned to the service type in the case give error message. Task cannot be allocated for service type: <Service Type>. Please contact system administrator. Case Transition will be rolled back in this case. |
| <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.AutoTaskCreationForVendor   |
| <b>Parameters</b>      | <p><b>Name:</b> Service Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Service Type</p> <p><b>Name:</b> Task Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Queue</p> |
| <b>Detailed Design</b> | Create a Task of given Task Type and assign it to the queue code specified in the parameter. Additionally assign the task to the vendor defined against the service type for the case.  |

**Table 5–125 Notify Repossession Specialist on Task Completion C1-NOTRSTSK**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Notify Repossession Specialist on Task Completion   |
| <b>Detailed Description</b> | <p>Create Notification<br/>Notification: &lt;Task Id&gt; - &lt;Task Name&gt; complete for &lt;Collateral Code&gt; &lt;Collateral Description&gt;.</p> <p>Set Display Date of the case to current business date.<br/>Notification should be created on the case associated to the task.<br/>This algorithm can be attached to any case level task on the Repossession case to alert the repossession specialist.</p> |
| <b>Algorithm Entity</b>     | To Do Type - To Do Post Processing  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.NotifyOnTaskCompletion  |
| <b>Parameters</b>           | <p><b>Name:</b> Display Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Display Date</p>   |
| <b>Detailed Design</b>      | Create Notification.  |



Table 5–126 Automatic sending of Redemption letters C1-REDEMPLTR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Automatic sending of Redemption letters   |
| <b>Detailed Description</b> | <p>For each of the accounts associated to the repossession case send the Redemption letter (create customer contact of given template code)</p> <p>If Only Primary Account = Yes then send letter only on the primary account.</p> <p>Parameter Description as follows :</p> <ol style="list-style-type: none"> <li>1. Contact Class - Contact class</li> <li>2. Contact Type - Contact type</li> <li>3. .Primary Account Switch - This switch determines weather contact should be generated for primary customer only or all associated customers. Possible values are Y,N,true,false</li> <li>4. Validation Date - Possible values are SYSTEM DATE and POSTING DATE</li> </ol> |
| <b>Algorithm Entity</b>     | Case Status - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutomaticSendingofRedemptionLetters  |
| <b>Parameters</b>           | <p><b>Name:</b> Contact Class<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Contact Class</p> <p><b>Name:</b> Contact Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Contact Type</p> <p><b>Name:</b> Primary Account Sw<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Primary Account Sw</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p>  |
| <b>Detailed Design</b>      | <p>For each of the accounts associated to the repossession case send the Redemption letter (create customer contact of given template code)</p> <p>If Only Primary Account = Yes then send letter only on the primary account.</p>  |

Table 5–127 Monitor for Redemption Proceeds C1-REDEPROC

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Monitor for Redemption Proceeds   |
| <b>Detailed Description</b> | When the outstanding amount of all the associated accounts becomes zero move the case to Closed Status. |
| <b>Algorithm Entity</b>     | Case Type - Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForRedemptionProc      |
| <b>Parameters</b>           | <b>Name:</b> Closed Status<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Closed Status        |
| <b>Detailed Design</b>      | When the outstanding amount of all the associated accounts becomes zero move the case to Closed Status. |

Table 5–128 Validate if appropriate Case Details have been entered by the user and transition C1-VALDATAPR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate if appropriate Case Details have been entered by the user and transition   |
| <b>Detailed Description</b> | <p>Validate if the Dynamic Panel Data Elements and Case Characteristics mentioned in the parameters have some values for the case.</p> <p>If yes the Follow Up is saved successfully and case is transitioned to the previous case status.</p> <p>If no system should throw an error message for the first blank field that it will encounter.<br/> Error Message: "&lt;Field Name&gt; cannot be blank"</p> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateRepoCaseData  |
| <b>Parameters</b>           | <b>Name:</b> Dynamic Panel One<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Dynamic Panel One<br><br><b>Name:</b> Dynamic Panel One Fields<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Dynamic Panel One Fields<br><br><b>Name:</b> Dynamic Panel Two Fields   |

|                    |  |
|--------------------|--|
|                    | <p><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Two Fields</p> <p><b>Name:</b> Dynamic Panel Three<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Three</p> <p><b>Name:</b> Dynamic Panel Three Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Three Fields</p> <p><b>Name:</b> Dynamic Panel Four<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Four</p> <p><b>Name:</b> Dynamic Panel Four Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Four Fields</p> |
| Parameters(Contd.) | <p><b>Name:</b> Dynamic Panel Five<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Five</p> <p><b>Name:</b> Dynamic Panel Five Fields<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Dynamic Panel Five Fields</p> <p><b>Name:</b> caseCharacteristics<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Case Characteristics</p> <p><b>Name:</b> Previous Status Transition<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Previous Status Transition</p>  |
| Detailed Design    | Validate if the Dynamic Panel Data Elements and Case Characteristics mentioned in the parameters have some values for the case.  |

Table 5–129 Monitor for Liquidation Setup Complete C1-LIQSETCMP

|                      |  |
|----------------------|--|
| Description          | Monitor for Liquidation Setup Complete   |
| Detailed Description | When Repo Title Received Date and Vehicle at Sale Location Date is available the case is moved to the next status. |

|                         |  |
|-------------------------|--|
| <b>Algorithm Entity</b> | Case Type - Auto Transition  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete       |
| <b>Parameters</b>       | <b>Name:</b> Next Status<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Next Status                       |
| <b>Detailed Design</b>  | When Repo Title Received Date and Vehicle at Sale Location Date is available the case is moved to the next status. |

**Table 5–130 Send Repossession Alert to Vendor C1-REPOASAL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Send Repossession Alert to Vendor  |
| <b>Detailed Description</b> | <p>Generate and send the email to the email id of the contact person associated to the service type mentioned in the parameter</p> <p>Email of specified template code will be sent.</p> <p>The algorithm will generate the contact as well as initiate contact processing</p>   |
| <b>Algorithm Entity</b>     | Case Status - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionAssignmentAlert   |
| <b>Parameters</b>           | <p><b>Name:</b> Contact Class<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Class</p> <p><b>Name:</b> Contact Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Type</p> <p><b>Name:</b> Service Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Service Type</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>      | <p>Generate and send the email to the email id of the contact person associated to the service type mentioned in the parameter.</p> <p>Email of specified template code will be sent.</p> <p>The algorithm will generate the contact as well as initiate contact processing</p>  |

**Table 5–131 Extract Algorithm Repossession Assignment C1-REPEMTEMP**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Extract Algorithm Repossession Assignment  |
| <b>Detailed Description</b> | Extract all the Collateral, Account and Customer Information and send it to Alert Module.<br>The contact person details of the Vendor will also be sent to the Alert Module to generate the alert. |
| <b>Algorithm Entity</b>     | Letter Template Letter Extraction Collection Algorithm   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ExtractRepossessionAssignmentAlgorithm  |
| <b>Parameters</b>           | <b>Name:</b> Event Id<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Event Id<br><b>Name:</b> Activity Id<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Activity Id               |
| <b>Detailed Design</b>      | Extract all the Collateral, Account and Customer Information and send it to Alert Module.<br>The contact person details of the Vendor will also be sent to the Alert Module to generate the alert. |

**Table 5–132 Monitor Redemption Clear Date C1-REDCLRDT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Monitor Redemption Clear Date  |
| <b>Detailed Description</b> | When the redemption clear date is reached transition the case to the Liquidation Setup Status.   |
| <b>Algorithm Entity</b>     | Case Type - Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorRedemptionClearDate   |
| <b>Parameters</b>           | <b>Name:</b> Next Status<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Next Status<br><br><b>Name:</b> Validate Date<br><b>Required (Yes/No):</b> No |

|                        |  |
|------------------------|--|
|                        | <b>Description:</b> Validate Date  |
| <b>Detailed Design</b> | When the redemption clear date is reached transition the case to the Liquidation Setup Status. |

**Table 5–133 Result Post Processing Algorithm for Approvals C1-RAPRVRSLT**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Result Post Processing Algorithm for Approvals  |
| <b>Detailed Description</b> | Transition the case to given Case Status if Case Status is configured.<br>Close the Approval Task Type present on the case if approval task type is configured.<br>Copy the comments in the result to the Approver remarks field  |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionApprovalResultPostProcessingAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> Case Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Case Status</p> <p><b>Name:</b> Approval Task Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Approval Task Type</p> <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p> |
| <b>Detailed Design</b>      | Transition the case to given Case Status if Case Status is configured.<br>Close the Approval Task Type present on the case if approval task type is configured.<br>Copy the comments in the result to the Approver remarks field  |

**Table 5–134 Adhoc Characteristic Value Validation Algorithm PASTDATE\_VAL**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Result Characteristic Value Date field Validation  |
| <b>Detailed Description</b> | This algorithm is used to validate format enter by user for result characteristics during follow up.<br>Validation Date: Validation Date will validate and compare the date with user provided date.<br>It's value can be system date or posting date. |

|                         |  |
|-------------------------|--|
|                         | <p>This is mandatory parameter.</p> <p>The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter</p> <p>The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS.</p> <p>Stored Date Format is a mandatory parameter whereas Date Format2 is not.</p> <p>Date Format2 is given for future requirement, if any.</p>   |
| <b>Algorithm Entity</b> | Characteristic Type - Adhoc Validation   |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDate  |
| <b>Parameters</b>       | <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Date</p> <p><b>Name:</b> Stored Date Format<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Stored Date Format</p> <p><b>Name:</b> Date Format2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Date Format2</p>   |
| <b>Detailed Design</b>  | <p>This algorithm is used to validate format enter by user for result characteristics during taking follow up.</p> <p>Validation Date: This Validation Date will validate and compare the date with user provided date. It's value can be system date or posting date.<br/>This is mandatory parameter.</p> <p>The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter</p> <p>The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS.</p> <p>Stored Date Format is a mandatory parameter whereas Date Format2 is not.</p> <p>Date Format2 is given for future requirement, if any.</p> |

**Table 5–135 Result Post Processing Algorithm for Redemption Clear Date C1-RDEEMDATE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Redemption Clear Date Value Date field Calculation  |
| <b>Detailed Description</b> | This algorithm is used to calculate the Redemption Clear Date.<br><br>By Default Redemption Clear Date will be calculated if REDEM_CLEAR_DT in CI_REPO_CLOSURE table is null and will be calculated as repossession Date + Redemption Clearing Days. Otherwise, Redemption Clear Date will be shown as per the date mentioned in REDEM_CLEAR_DT in CI_REPO_CLOSURE table. |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDateCal  |
| <b>Parameters</b>           | <b>Name:</b> Characteristic Type Code<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Characteristic Type Code   |
| <b>Detailed Design</b>      | This algorithm is used to calculate the Redemption Clear Date.<br><br>By Default Redemption Clear Date will be calculated if REDEM_CLEAR_DT in CI_REPO_CLOSURE table is null and will be calculated as repossession Date + Redemption Clearing Days. Otherwise, Redemption Clear Date will be shown as per the date mentioned in REDEM_CLEAR_DT in CI_REPO_CLOSURE table. |

## 5.46 Miscellaneous

**Table 5–136 Update Review Date for associated accounts C1-UPDRVWDT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Review Date for associated accounts   |
| <b>Detailed Description</b> | <p>For all accounts associated with the case this process will update the review date. Below parameters should be available for the process</p> <p>Update Type</p> <ul style="list-style-type: none"> <li>■ Set Review Date - This will set the Review Date for the account</li> <li>■ Remove Review Date - This will remove the Review date from the account</li> </ul> <p>Days Offset - Applicable only of Update Type = Set. System will set the review date as Current business days + Offset days.</p> <p>Override Flag</p> <ul style="list-style-type: none"> <li>■ Yes: System will update existing account review date i.e. in case a review date is already present, system will override the same</li> <li>■ No: System will not update existing account review date i.e. in case a review date is already present, system will not override the same</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.UpdateReviewDate  |
| <b>Parameters</b>       | <p><b>Name:</b> Override Flag Value<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Override Flag Value</p> <p><b>Name:</b> Days Offset<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Days Offset</p> <p><b>Name:</b> Update Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Update Type</p> |

Table 5–137 Case Monitoring CS-MONITOR

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Case Monitoring  |
| <b>Detailed Description</b> | <p>This algorithm determines if a case has been in its current status long enough to be automatically transitioned to another status or some other action needs to be taken on case.</p> <p>If the case has been in its current status for more than the given Number of days, it is allowed to do the following activity as per configuration:</p> <ol style="list-style-type: none"> <li>1. Create a To Do, for a given To Do type.</li> <li>2. Re-Allocate the case to a different Queue.</li> <li>3. Set Prompt Days.</li> <li>4. Transition to another Status.</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type - Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.CaseAutoTransition  |
| <b>Parameters</b>           | <p><b>Name:</b> Next Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Next Status</p> <p><b>Name:</b> Work List<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Work List</p> <p><b>Name:</b> Reallocate Switch</p>  |

|  |   |
|--|---|
|  | <p><b>Required (Yes/No):</b> No<br/><b>Description:</b> Reallocate Switch</p> <p><b>Name:</b> To Do Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> To Do Type</p> <p><b>Name:</b> No Of Days<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> No Of Days</p> |
|--|---|

**Table 5–138 Update warning indicator for the customer C1-UPDWARN**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update warning indicator for the customer  |
| <b>Detailed Description</b> | <p>This process will update the warning indicator for the customer</p> <ul style="list-style-type: none"> <li>■ Update activity i.e. set or remove the warning will also be defined as parameter to this process</li> <li>■ Warning indicator to be set or removed will be set as parameter to this process</li> <li>■ Additionally process will have a parameter to define if update needs to be done only for the customer associated as primary entity or for all customers associated to the case</li> </ul> <p>Call the service form host to update the warning indicator.<br/>Please give following values for the below parameters:<br/>Association Type : P (Primary) and A (Primary and Secondary)<br/>Update Type : S (Set) and R (Remove)</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.UpdateWarningIndicator   |
| <b>Parameters</b>           | <p><b>Name:</b> Association Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Override Association Type</p> <p><b>Name:</b> Warning Indicator Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Warning Indicator Type</p> <p><b>Name:</b> Update Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Update Type</p>   |

**Table 5–139 Transition to Default Next Status C1-TRAN-STAT**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to Default Next Status  |
| <b>Detailed Description</b> | <p>This is a common algorithm that will automatically transition the case to the next status.</p> <p>Following are the parameters :</p> <ol style="list-style-type: none"> <li>1. Next Status - The next status to which the case will be transitioned.</li> <li>2. Next Transition Condition - Mention the transition condition for the next status.</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type - Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.TransitionDefaultNextStatus   |
| <b>Parameters</b>           | <p><b>Name:</b> Next Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Status</p> <p><b>Name:</b> Next Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Transition Condition</p>  |

**Table 5–140 Set Account Warning Indicator C1-ACTCSWGID**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Set Account Warning Indicator  |
| <b>Detailed Description</b> | Set Account Warning Indicator for host accounts  |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collectionevt.SetCaseWarningIndOnHost   |
| <b>Parameters</b>           | <p><b>Name:</b> Account Warning Indicator<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Account Warning Indicator Code</p> |
| <b>Detailed Design</b>      | Set Account Warning Indicator for host accounts  |

## 5.47 Derived Field

**Table 5–141 Timezone derivation field update algorithm C1-TZDRFLD**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Timezone derivation field update algorithm                     |
| <b>Detailed Description</b> | This algorithm will update timezone of a person if it is blank |

|                         |   |
|-------------------------|---|
| <b>Algorithm Entity</b> | Timezone derivcation Algorithm Spot   |
| <b>Program Type</b>     | Java (Converted)  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.batch.algorithm.TimeZoneDerivationAlgorithm |
| <b>Parameters</b>       | NA  |
| <b>Detailed Design</b>  | This algorithm will update timezone of a person if it is blank              |

## 5.48 Task

*Table 5–142 Validate Task Completion (Case Closure) C1-VALTASKCM*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validate Task Completion (Case Closure)  |
| <b>Detailed Description</b> | Validate if given tasks have been completed before entering the status<br>For case level tasks check if any open tasks on the case id.<br>For account level tasks check if any open tasks on the accounts associated with the case.<br>For customer level tasks check if any open tasks on the customers associated with the case. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status Validation  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletionClosure   |
| <b>Parameters</b>           | <b>Name:</b> Task Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Task Type   |
| <b>Detailed Design</b>      | Validate if given tasks have been completed before entering the status   |

*Table 5–143 Validate Task Completion C1-VALTASKEX*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validate Task Completion   |
| <b>Detailed Description</b> | Validate if given tasks have been completed before exiting the status.<br>For case level tasks check if any open tasks on the case id.<br>For account level tasks check if any open tasks on the accounts associated with the case.<br>For customer level tasks check if any open tasks on the customers associated with the case. |
| <b>Algorithm Entity</b>     | Case Type - Exit Status Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | c.com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletion  |
| <b>Parameters</b>           | <b>Name:</b> Task Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Task Type   |
| <b>Detailed Design</b>      | Validate if given tasks have been completed before entering the status   |

**Table 5–144 Automatic Task Creation when case enters a particular status C1-CREATTASK**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Automatic Task Creation when case enters a particular status  |
| <b>Detailed Description</b> | <p>If case level task create a task on the case id.</p> <p>If account level task create a task each on all the accounts associated on the case.</p> <p>If customer level task create a task each on all the customers associated on the case.</p>   |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.tasks.algo.AutomaticTaskCreatiomn   |
| <b>Parameters</b>           | <p><b>Name:</b> Task Type1<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue1<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Queue</p> <p><b>Name:</b> Task Type2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Queue</p> <p><b>Name:</b> Task Type3<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue3<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Queue</p> <p><b>Name:</b> Task Type4<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Task Type</p> <p><b>Name:</b> Queue4<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Queue</p> |

|                           |  |
|---------------------------|--|
| <b>Parameters (Cont.)</b> | <b>Name:</b> Task Type5<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Task Type<br><br><b>Name:</b> Queue5<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Queue |
| <b>Detailed Design</b>    | Automatic Task Creation when case enters a particular status   |

## 5.49 Event Manager

This table provides details of the algorithm used for Event Manager spot.

*Table 5–145 Set Account Warning Indicator*

|                         |   |
|-------------------------|---|
| <b>Description</b>      | Set Account Warning Indicator   |
| <b>Algorithm Entity</b> | Generic Event Outcome Algorithm Spot  |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collectionevt.SetWarningIndOnHost  |
| <b>Parameters</b>       | <b>Name:</b> Account Warning Indicator<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Account Warning Indicator Code |
| <b>Detailed Design</b>  | Set Account Warning Indicator for host accounts   |

## 5.50 Legal Vendor Allocation C1-LGLVNDRAL

*Table 5–146 Legal vendor Allocation C1-LGLVNDRAL*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Legal vendor Allocation  |
| <b>Detailed Description</b> | Legal vendor Allocation  |
| <b>Algorithm Entity</b>     | Vendor Service Type Allocation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.LspVendorAllocationAlgorithm |
| <b>Parameters</b>           | NA   |

Table 5–147 Validate Extended Expiry Date C1-VAL-EXPDT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validate Extended Expiry Date  |
| <b>Algorithm Entity</b>     | Generic Event Outcome Algorithm Spot   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateExtendedExpiryDate   |
| <b>Parameters</b>           | <p>Hardship Expiry Date Characteristic<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Hardship Expiry Date Characteristic Type Code</p> <p><b>Name:</b> Extended Expiry Date Characteristic<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Extended Expiry Date Characteristic Type Code</p> |
| <b>Detailed Description</b> | Validate Extended Expiry Date  |

## 5.51 Extend Expiry Date C1-EXT-EXPDT

Table 5–148 Extend Expiry Date C1-EXT-EXPDT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Extend Expiry Date   |
| <b>Detailed Description</b> | This algorithm will invoke the host service to extend the hardship expiry date   |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ExtendedExpiryDate   |
| <b>Parameters</b>           | <p><b>Name:</b> Extended Expiry Date Char Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Extended Expiry Date Char Type Code</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exception Transition Condition</p> |

## 5.52 Capture Case Status Update Date/Time C1-CASE-STAT

Table 5–149 Capture Case Status Update Date/Time C1-CASE-STAT

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Capture Case Status Update Date/Time  |
| <b>Detailed Description</b> | This algorithm will store Case Status Update Date/Time for current status into the element specified by xpath in algorithm soft parameter.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureEnterStausUpdateDateTime   |
| <b>Parameters</b>           | <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exception Transition Condition Code</p> <p><b>Name:</b> Xpath to Date Element<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Xpath to Date Element Code</p> |

## 5.53 Create To Do C1-TO-DO

Table 5–150 Create To Do C1-TO-DO

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Create To Do   |
| <b>Detailed Description</b> | This common algorithm creates a To Do using the values from algorithm parameters.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.CreateToDo  |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> To Do Type Code</p> <p><b>Name:</b> Message Category<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Message Category Code</p> <p><b>Name:</b> Message Number<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Message Number Code</p> |



|  |   |
|--|---|
|  | <p><b>Name:</b> Characteristic Type For Log Entry<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Characteristic Type For Log Entry Code</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exception Transition Condition Code</p> |
|--|---|

## 5.54 Check customer eligibility C1-CHKCUST

Table 5–151 Check customer eligibility C1-CHKCUST

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check customer eligibility   |
| <b>Detailed Description</b> | <p>This process will check warning indicators for a customer. This check will be done by a call to rule engine for each customer. Processing logic will be as below:</p> <p>Primary entity for the case is account. Based on ownership type parameter for the process, system should consider the customers for eligibility check.</p> <ul style="list-style-type: none"> <li>■ If ownership type parameter is set to "financial owner" <ul style="list-style-type: none"> <li>• Get all financially responsible customers for the account.</li> <li>• For each customer, system should call the rule engine to check for customer eligibility.</li> </ul> </li> <li>■ If ownership type parameter is set to "primary" <ul style="list-style-type: none"> <li>• System should call the rule engine to check for primary customers eligibility.</li> </ul> </li> </ul> <p>Customers' facts should be used for rule engine decision.<br/> For each call</p> <ul style="list-style-type: none"> <li>■ Rule will return output as "Validation Status". Possible values can be "Success" OR "Failure".</li> <li>■ If validation status = Failed, process should return result as validation failed. <ul style="list-style-type: none"> <li>• Check Validation failure option <ul style="list-style-type: none"> <li>○ Validation failure option = Fail case creation/transition. Case should not get created or should not transition status.</li> <li>○ Validation failure option = Transition status. Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters).</li> </ul> </li> </ul> </li> <li>■ If validation status = Success, process should return result as validation successful.</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>■ Ownership Type - Ownership type can be FINANCIAL_OWNER(Financial Owner) or</li> </ul> |

|                         |   |
|-------------------------|---|
|                         | <p>PRIMARY(Primary Owner).</p> <ul style="list-style-type: none"> <li>■ Rule ID - Defined rule id to check customer eligibility. Rule should return output validation status in fact 'SuccessOrFailure', which can have value true or false.</li> <li>■ Validation Failure Option - This option is used to determine action to be taken in case of validation failure. Permissible values are FAIL_CASE_CREATION(fail case creation) and TRANSITION_STATUS(transition status).</li> <li>■ Validation Failure Transition Status - Case transition status in case of validation failure.</li> <li>■ Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status.</li> <li>■ Cancel Reason Char Value - Characteristic value for the defined characteristic type.</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.roso.CheckCustomerEligibility   |
| <b>Parameters</b>       | <p><b>Name:</b> Cancel Reason Char Value<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Cancel Reason Char Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Type</p> <p><b>Name:</b> Validation Failure Transition Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Validation Failure Transition Status</p> <p><b>Name:</b> Validation Failure Option<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Failure Option</p> <p><b>Name:</b> Rule Id<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Rule Id</p> <p><b>Name:</b> Ownership Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Ownership Type</p>  |

## 5.55 Capture Hardship Approval Date C1-HARAP-DT

Table 5–152 Capture Hardship Approval Date C1-HARAP-DT

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Capture Hardship Approval Date  |
| <b>Detailed Description</b> | This algorithm will store Case Status Update Date/Time for current status into the element specified by xpath in algorithm soft parameter.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureHardshipApprovalDate   |
| <b>Parameters</b>           | <p><b>Name:</b> Xpathto Date Element<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Xpathto Date Element</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exception Transition Condition</p> |

## 5.56 Algorithm that will interface with Rule Engine. C1-RULEADAPT

Table 5–153 Algorithm that will interface with Rule Engine. C1-RULEADAPT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm that will interface with Rule Engine.          |
| <b>Detailed Description</b> | Algorithm that will interface with Rule Engine.          |
| <b>Algorithm Entity</b>     | Case Type - Enter Status Validation                      |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseCreation.AdapterTest |
| <b>Parameters</b>           | NA   |

## 5.57 Perform Validation for Collateral C1-VRFYCOL

Table 5–154 Perform Validation for Collateral C1-VRFYCOL

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Perform Validation for Collateral   |
| <b>Detailed Description</b> | This algorithm type will perform below validations for the collateral with the case |

|                         |  |
|-------------------------|--|
| <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>       | <b>Name:</b> CaseCategory<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Category Code |

## 5.58 Check target account eligibility C1-CHKTRGT

Table 5–155 Check target account eligibility C1-CHKTRGT

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Check target account eligibility  |
| <b>Detailed Description</b> | <ul style="list-style-type: none"> <li>■ System should call the rule engine for eligibility check. Output of rule engine will be "Validation Status" <ul style="list-style-type: none"> <li>• If validation status is "Success" <ul style="list-style-type: none"> <li>◦ Set set-off status as "Pending"</li> <li>◦ Compute maximum amount allowed to Debit = Target account Balance - Minimum residual amount</li> <li>◦ Clear the values in the "Exclude Target Till Date" and "Exclude Reason" fields, if populated</li> </ul> </li> <li>• If validation status is "Fail" <ul style="list-style-type: none"> <li>◦ Set set-off status for target account as "Not eligible"</li> <li>◦ "Exclude Reason" should be set as "Not Eligible"</li> <li>◦ Get offset days for exclude reason from the look-up</li> <li>◦ Set "Exclude target till" date for the target account to current business days + offset day.</li> <li>◦ If no offset days are returned, "Exclude target till" date should not be updated</li> </ul> </li> </ul> </li> <li>■ Once all target accounts have been processed and for this case, if none of the target accounts has set-off status as "Pending". <ul style="list-style-type: none"> <li>• Case should be created and transitioned to the status specified in parameters.</li> <li>• Set given char value for the given char type (as defined in parameters)</li> </ul> </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.roso.CheckTargetAccountEligibility  |
| <b>Parameters</b>   | <p><b>Name:</b> Cancel Reason Char Value<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Cancel Reason Char Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Type Code</p> <p><b>Name:</b> Validation Failure Transition Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Failure Transition Status Code</p> <p><b>Name:</b> Minimum Residual Amount<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Minimum Residual Amount</p> <p><b>Name:</b> Rule Id<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Rule Id</p> |

## 5.59 Approval check for set-off transaction C1-ROSOAPPR

Table 5–156 Approval check for set-off transaction C1-ROSOAPPR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Approval check for set-off transaction  |
| <b>Detailed Description</b> | <p>This process will check if approval is required for a set-off transaction. Approval will be required if</p> <ul style="list-style-type: none"> <li>■ Asset classification = Value set as parameter for the process</li> <li>■ Accrual status = Value set as parameter for the process</li> <li>■ Sum of Debit Amounts for all target accounts &gt;= Specified threshold</li> </ul> <p>Based on whether approval is required or not, transition the case to a status as set in the parameters of the process.</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.roso.RosoApprovalCheck  |
| <b>Parameters</b>           | <b>Name:</b> Total Debit Amount Threshold   |

|  |  |
|--|--|
|  | <p><b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Total Debit Amount Threshold</p> <p><b>Name:</b> No Approval Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> No Approval Status</p> <p><b>Name:</b> Approval Required Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Approval Required Status</p> <p><b>Name:</b> Accrual Status Flag<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Accrual Status Flag</p> <p><b>Name:</b> Asset Classification Code<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Asset Classification Code</p> |
|--|--|

## 5.60 Get target accounts C1-GETTRGT

Table 5–157 Get target accounts C1-GETTRGT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Get target accounts  |
| <b>Detailed Description</b> | <p>This algorithm gets all savings accounts and term deposit accounts having the same set of owners as owners of the delinquent account processing logic for this will be as below:</p> <ul style="list-style-type: none"> <li>■ Get all savings accounts and term deposit accounts having the same set of owners as owners of the delinquent account (that is, primary account associated with the case). Ownership types can however be different.</li> <li>■ Same owners: Indicates that all owners of delinquent accounts are one the savings account / term deposit and there is no additional owner</li> <li>■ If no such accounts are found, Case should be created and transitioned to the status specified in parameters. Set given char value for the given char type (as defined in parameters).</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>■ Validation Failure Transition Status - Case transition status in case of validation failure.</li> <li>■ Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status.</li> <li>■ Cancel Reason Char Value - Characteristic value for the defined characteristic type.</li> <li>■ Casa Account Type Identifier List - Comma separated savings account (CASA)</li> </ul> |

|                         |   |
|-------------------------|---|
|                         | <p>identifiers.</p> <ul style="list-style-type: none"> <li>■ Td Account Type Identifier List - Comma separated term Deposit account (TD) identifiers.</li> <li>■ Casa Account Exclude Status List - Comma separated savings account (CASA) status to be excluded while fetching account data from host.</li> <li>■ Td Account Exclude Status List - Comma separated term Deposit account (TD) status to be excluded while fetching account data from host.</li> <li>■ Exclude Blocked Td Account - Flag to exclude blocked Term Deposit account (Y or N).</li> <li>■ Exclude Blocked Deposit - Flag to exclude blocked Deposit (Y or N).</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.roso.FetchTargetAccounts  |
| <b>Parameters</b>       | <p><b>Name:</b> Validation Failure Transition Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Failure Transition Status</p> <p><b>Name:</b> Cancel Reason Char Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Type Code</p> <p><b>Name:</b> Cancel Reason Char Value<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Casa Account Type Identifier List<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Casa Account Type Identifier List</p> <p><b>Name:</b> Casa Account Exclude Status List<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Casa Account Exclude Status List</p> <p><b>Name:</b> Td Account Type Identifier List<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> To Do Account Type Identifier List</p> <p><b>Name:</b> Td Account Exclude Status List<br/> <b>Required (Yes/No):</b> Yes</p> |

|  |   |
|--|---|
|  | <p><b>Description:</b> Td Account Exclude Status List</p> <p><b>Name:</b> Exclude Blocked Td Account<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exclude Blocked Td Account</p> <p><b>Name:</b> Exclude Blocked Deposit<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exclude Blocked Deposit</p> |
|--|---|

## 5.61 Update status of relief to Expired in Hardship C1-UPDHDSTAT

Table 5–158 Update status of relief to Expired in Hardship C1-UPDHDSTAT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update status of relief to Expired in Hardship   |
| <b>Detailed Description</b> | Update status of relief to Expired in Hardship details table.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateHardshipStatusToExpire                     |
| <b>Parameters</b>           | <p><b>Name:</b> Hardship Expire Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Hardship Expire Status Code</p> |

## 5.62 To Do Completion for case C1-TO-DO-END

Table 5–159 To Do Completion for case C1-TO-DO-END

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | To Do Completion for case   |
| <b>Detailed Description</b> | This common algorithm will complete all To Do's with Drill Keys = Current Case Id and To Do's To Do Type is not excluded from auto completion.                                    |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.GenericToDoCompletionForCase   |
| <b>Parameters</b>           | <p><b>Name:</b> Do Not Complete To Do Type Characteristic Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Do Not Complete To Do Type Characteristic Type Code</p> |



|  |   |
|--|---|
|  | <b>Name:</b> Do Not Complete To Do Type Characteristic Value<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Do Not Complete To Do Type Characteristic Value |
|--|---|

## 5.63 Update Marketing Consent flag C1-MKT-FLG

Table 5–160 Update Marketing Consent flag C1-MKT-FLG

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Marketing Consent flag   |
| <b>Detailed Description</b> | This is a generic algorithm that will make a service call to host to update the Marketing Consent flag.   |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateMarketingConsentFlag  |
| <b>Parameters</b>           | <b>Name:</b> Marketing Consent Flag Value<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Marketing Consent Flag Value<br><br><b>Name:</b> Exception Transition Condition<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Exception Transition Condition |

## 5.64 Check Default Notice for Voluntary possession C1-CHKDFLT

Table 5–161 Check Default Notice for Voluntary possession C1-CHKDFLT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check Default Notice for Voluntary possession  |
| <b>Detailed Description</b> | Check Default Notice for Voluntary possession  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program</b>              | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CheckDefaultNoticeForVoluntaryPossession |

|                   |  |
|-------------------|--|
| <b>Name</b>       |  |
| <b>Parameters</b> | <b>Name:</b> Check Expiry Status<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Check Expiry Status |

## 5.65 Check Submission Date C1-CHKSUBDT2

Table 5–162 Check Submission Date C1-CHKSUBDT2

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Check Submission Date   |
| <b>Detailed Description</b> | Check Submission Date   |
| <b>Algorithm Entity</b>     | Case Type - Exit Status   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDateExitProcessing |
| <b>Parameters</b>           | NA  |

## 5.66 Update Financial Hardship flag C1-FNHRD-FLG

Table 5–163 Update Financial Hardship flag C1-FNHRD-FLG

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Financial Hardship flag  |
| <b>Detailed Description</b> | This algorithm will make a service call to host to update the Financial Hardships flag for Primary Customer and corresponding joint account holders.  |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateFinancialHardshipFlag   |
| <b>Parameters</b>           | <b>Name:</b> Financial Hardship Flag Value<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Financial Hardship Flag Value<br><br><b>Name:</b> Exception Transition Condition<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Financial Exception Transition Condition |

## 5.67 Result Type Case Transition Algorithm C1-RTCT

Table 5–164 Update Financial Hardship flag C1-FNHRD-FLG

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Result Type Case Transition Algorithm   |
| <b>Detailed Description</b> | <p>If specified on the Result Type, this algorithm will be invoked when the corresponding result is recorded for a Case (Action/Result UI)</p> <p>This can be used to transition the case from the current status to the next possible status as follows:</p> <ul style="list-style-type: none"> <li>■ This algorithm has a parameter Output Status i.e. next possible status, so for case transition, it will be checked whether Output Status is one of the next possible status. If YES, it will transition the case to that status.</li> <li>■ This algorithm has a parameter Input Status, which will be checked against the current status of the Case. This is an optional parameter. If specified, Case transition will happen only when the current status of the case matches with this parameter.</li> </ul> |
| <b>Algorithm Entity</b>     | Result Type – Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypeCaseTransitionAlgo  |
| <b>Parameters</b>           | <p><b>Name:</b> Output Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Output Status</p> <p><b>Name:</b> Input Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Input Status</p>  |

## 5.68 Algorithm to see if case is running before closing C1-CHKCASE

Table 5–165 Algorithm to see if case is running before closing. C1-CHKCASE

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Algorithm to see if case is running before closing  |
| <b>Detailed Description</b> | The algorithm sees if the case is running in the child case category before closing the case from the parent case category. |
| <b>Algorithm Entity</b>     | Case Type - Exit Status   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckActiveArs Case                                   |

|                   |  |
|-------------------|--|
| <b>Parameters</b> | <b>Name:</b> Case Category<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Category |
|-------------------|--|

## 5.69 Check Deceased status for the customer C1-CHKDCD

*Table 5–166 Check Deceased status for the customer .C1-CHKDCD*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check Deceased status for the customer   |
| <b>Detailed Description</b> | For the customer for whom the deceased case is being initiated check if,<br>Deceased warning indicator is already set OR<br>An active deceased case is present<br>If either of above is true, case creation should fail. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.CheckDeceasedStatusForCustomer  |
| <b>Parameters</b>           | <b>Name:</b> Case Category<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Category   |

## 5.70 Associated accounts with deceased customer case C1-DCDACCTS

*Table 5–167 Associated accounts with deceased customer case.C1-DCDACCTS*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Associated accounts with deceased customer case  |
| <b>Detailed Description</b> | For the primary customer associated with the case.<br>Get all accounts where this customer is primary owner and the accounts are in collections.<br>Associated those accounts with the case. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.AssociatedAccWithDeceasedCustomer   |
| <b>Parameters</b>           | NA   |

## 5.71 Execute Fund Transfer C1-FUNDTRFR

Table 5–168 Execute Fund Transfer C1-FUNDTRFR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Execute Fund Transfer   |
| <b>Detailed Description</b> | <p>This process will execute the fund transfer. This should follow below steps for "each" target account where debit amount specified is &gt; 0 and set-off status = "Pending"</p> <ul style="list-style-type: none"> <li>■ Execute a payment transfer transaction from Target account to the delinquent account.</li> <li>■ If transaction is successful, set set-off status = "Success" for this target account</li> <li>■ If transaction is not successful, set set-off status = "Fail" for this target account</li> </ul> <p>For any target account where set-off status was "Pending", but were not considered for set-off (because debit amount was specified as zero) update set-off status to "Cancelled"</p> <p>Once all target accounts have been processed, check if at least one payment transfer has status as "Success".</p> <ul style="list-style-type: none"> <li>■ If yes, transition the case to status as set in the parameter "Execution Success Status"</li> <li>■ If no, transition the case to status as set in the parameter "Execution Failure Status". Set the char value for the char type as specified in the parameters.</li> </ul> <p>Parameters:</p> <p>Execution Success Status - Case transition status if fund transfer is successful.</p> <p>Execution Failure Status - Case transition status if fund transfer fails.</p> <p>Cancel Reason Char Type - Characteristic type to set as case characteristic if fund transfer fails.</p> <p>Cancel Reason Char Value - Characteristic value for the defined characteristic type.</p> <p>Successful Fund Transfer Transaction Status - Transaction status code to identify successful fund transfer. This value is returned from host service.</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.roso.ExecuteFundTransfer  |
| <b>Parameters</b>           | <p><b>Name:</b> Execution Success Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Execution Success Status Code</p> <p><b>Name:</b> Execution Failure Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Execution Failure Status</p> <p><b>Name:</b> Cancel Reason Char Type</p>  |

|  |   |
|--|---|
|  | <p><b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Type</p> <p><b>Name:</b> Cancel Reason Char Value<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Successful Fund Transfer Transaction Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Successful Fund Transfer Transaction Status</p> |
|--|---|

## 5.72 Algorithm to save previous state's status code C1-SAVPRESTA

Table 5–169 Algorithm to save previous state's status code.C1-SAVPRESTA

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Algorithm to save previous state's status code.   |
| <b>Detailed Description</b> | Algorithm to save the case status in CI_LSP_DTLS table from where it has come to the current status. This algorithm is must when we are using C1-RESSTATUS. C1-RESSTATUS transition the case to the status which is saved by this (C1-SAVPRESTA) algorithm. |
| <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.SavePreviousStatus  |
| <b>Parameters</b>           | NA  |

## 5.73 Attach case type from feature config attach to BO C1-ATCHCS

Table 5–170 Attach case type from feature config attach to BO. C1-ATCHCS

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Attach case type from feature config attach to BO |
| <b>Detailed Description</b> | Attach case type from feature config attach to BO |
| <b>Algorithm Entity</b>     | Business Object – Pre-Processing                  |
| <b>Program Type</b>         | Java  |

|                     |  |
|---------------------|--|
| <b>Program Name</b> | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignCaseTypeFromFeatureConfig                        |
| <b>Parameters</b>   | <b>Name:</b> Hardship Case Type Feature Config<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Hardship Case Type Feature Config |

## 5.74 Update Collection Warning Indicator C1-UPD-WRIND

Table 5–171 Update Collection Warning Indicator. C1-UPD-WRIND

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Collection Warning Indicator  |
| <b>Detailed Description</b> | <p>This is a generic algorithm that will make a service call to the Host to update Party level warning indicators for the Main Customer.</p> <p>It has following parameters:</p> <ol style="list-style-type: none"> <li>1. Warning Indicator Type.</li> <li>2. Warning Indicator Value</li> <li>3. Rule Type Code</li> <li>4. Collection Column To Be Updated</li> <li>5. Set In Collections On Related Accounts</li> <li>6. Exception Transition Condition</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateCollectionPartyWarningIndicator  |
| <b>Parameters</b>           | <b>Name:</b> Warning Indicator Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Hardship Warning Indicator Type<br><br><b>Name:</b> Warning Indicator Value<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Warning Indicator Value<br><br><b>Name:</b> Rule Type Code<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Rule Type Code<br><br><b>Name:</b> Collection Column To Be Updated   |

|  |  |
|--|--|
|  | <p><b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Collection Column To Be Updated</p> <p><b>Name:</b> Set In Collections On Related Accounts<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Set In Collections On Related Accounts</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Exception Transition Condition</p> |
|--|--|

## 5.75 Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSO

*Table 5–172 Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSO*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Hardship Entity Association to nominated accounts and financial owners of account  |
| <b>Detailed Description</b> | This algorithm associates all the accounts nominated for hardship and also associates related financial Owners of the accounts selected. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipEntityAssociation                              |
| <b>Parameters</b>           | NA   |

## 5.76 Assign Applicable Relief Type C1-RELIF-TYP

*Table 5–173 Assign Applicable Relief Type C1-RELIF-TYP*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Assign Applicable Relief Type  |
| <b>Detailed Description</b> | This algorithm will invoke Rules Engine to determine Applicable Relief Type(s) for each nominated Account. |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program</b>              | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.Assign                   |



|                   |  |
|-------------------|--|
| <b>Name</b>       | ApplicableReliefTypes  |
| <b>Parameters</b> | <b>Name:</b> Rule Type Code<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Rule Type Code |

## 5.77 Create Customer Contact for Resultype Algo C1-CREATCC

Table 5–174 Create Customer Contact for Resultype Algo. C1-CREATCC

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Create Customer Contact for Resultype Algo   |
| <b>Detailed Description</b> | Create Customer Contact for Resultype Algo   |
| <b>Algorithm Entity</b>     | Result Type – Post Processing Algorithm  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.interaction.CreateCustomerContactAlgo  |
| <b>Parameters</b>           | <b>Name:</b> Customer Contact Class<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Customer Contact Class<br><br><b>Name:</b> Customer Contact Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Customer Contact Type<br><br><b>Name:</b> Preferred Contact Method<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Preferred Contact Method |

## 5.78 Calculate an expiry date when entering case status C1-CSEXPDT

Table 5–175 Calculate an expiry date when entering case status. C1-CSEXPDT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Calculate an expiry date when entering case status   |
| <b>Detailed Description</b> | This algorithm type accepts a parameter for a characteristic type which will be used to create a Case Characteristic which contains a date that is equal to case status change plus Number of Days parameter value |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.CalculateCaseStatusExpiryDate   |
| <b>Parameters</b>           | <b>Name:</b> Number of Days<br><b>Required (Yes/No):</b> Yes   |

|  |   |
|--|---|
|  | <p><b>Description:</b> Number of Days</p> <p><b>Name:</b> Expiry Date Characteristic Type</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Expiry Date Characteristic Type</p> |
|--|---|

## 5.79 Create Customer Contact C1-CUST-CONT

Table 5–176 Create Customer Contact. C1-CUST-CONT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Create Customer Contact  |
| <b>Detailed Description</b> | This common algorithm creates a customer contact for the given customer contact type   |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.CustomerContact   |
| <b>Parameters</b>           | <p><b>Name:</b> Customer Class<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Customer Class</p> <p><b>Name:</b> Customer Contact Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Customer Contact Type</p> <p><b>Name:</b> Char Type Cust Cont Log Entry<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Char Type Cust Cont Log Entry</p> <p><b>Name:</b> X Path Completion Flag<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> X Path Completion Flag</p> <p><b>Name:</b> Transition Condition<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Transition Condition</p> <p><b>Name:</b> Contact Method<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Contact Method</p> |

## 5.80 Transition to Next Status x days before expiry C1-NXT-BX-DY

Table 5–177 Transition to Next Status x days before expiry C1-NXT-BX-DY

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Transition to Next Status x days before expiry  |
| <b>Detailed Description</b> | Transition to Next Status x days before expiry  |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.TransitionToNextDaysOnBeforeExpiry   |
| <b>Parameters</b>           | <p><b>Name:</b> Days Before Expiry<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Days Before Expiry</p> <p><b>Name:</b> Xpath to Expiry Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Xpath to Expiry Date</p> <p><b>Name:</b> Next Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Status</p> <p><b>Name:</b> Next Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Transition Condition</p> |

## 5.81 Validate Hardship Expiry Date. C1-VAL-FHEXP

Table 5–178 Validate Hardship Expiry Date. C1-VAL-FHEXP

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Validate Hardship Expiry Date  |
| <b>Detailed Description</b> | This validates the Hardship Expiry Date. It validates if the expiry date is greater than Posting date and the Allowed Minimum maturity date. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipExpiryDate                                 |
| <b>Parameters</b>           | <p><b>Name:</b> Xpath To Hardship Expiry Date<br/> <b>Required (Yes/No):</b> No</p>  |

|  |   |
|--|---|
|  | <p><b>Description:</b> Xpath To Hardship Expiry Date</p> <p><b>Name:</b> Expiry Date Char Type</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Expiry Date Char Type</p> |
|--|---|

## 5.82 Update Account in collections flag C1-ACTINCOL

Table 5–179 Update Account in collections flag. C1-ACTINCOL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Account in collections flag   |
| <b>Detailed Description</b> | <p>Get all accounts for the customer from the host.</p> <p>Relationship type to be considered will be primary or financial ownership based on parameter set for the process.</p> <p>For the accounts retrieved, check if the account is setup in collections i.e. an active contract is present for the account</p> <ul style="list-style-type: none"> <li>■ If no, set in-collections flag to "N" for the account</li> <li>■ If yes. No updates should be done</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.UpdateAccountInCollectionFlag  |
| <b>Parameters</b>           | <p><b>Name:</b> relationshipType</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Relationship_Type</p> <p><b>Name:</b> sourceHostId</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Source Host Id</p>   |

## 5.83 Associate related entities with the case C1-ARSENTITY

Table 5–180 Associate related entities with the case. C1-ARSENTITY

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Associate related entities with the case |
| <b>Detailed Description</b> | Associate related entities with the case |
| <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.CaseAssociationForAssetRepossessionCase   |
| <b>Parameters</b>   | <p><b>Name:</b> Customer Association<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Customer Association</p> <p><b>Name:</b> Account Association<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Account Association</p> |

## 5.84 Revalidate target account C1-REVALTRGT

Table 5–181 Revalidate target account. C1-REVALTRGT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Revalidate target account  |
| <b>Detailed Description</b> | <p>This algorithm validates target account (savings and term deposit) balance and computes maximum amount to be debited.</p> <p>Processing logic should be as below:</p> <ul style="list-style-type: none"> <li>■ Validate that "Total Debit Amount" is greater than zero. Else transition into the status should fail and appropriate error message be displayed OR recorded in case log (if not executed manually).</li> <li>■ It is possible that target account balance got updated after user had entered the debit amounts. System should refresh balance from host.</li> <li>■ Re-compute maximum amount which can be debited for each target account For each of the target account with set-off status as "Pending".</li> <li>■ If maximum amount which can be debited is &lt; Debit amount specified by the user then <ul style="list-style-type: none"> <li>• Set set-off status and exclude reason as "Not eligible".</li> <li>• Skip rest of the processing and move to next target account</li> </ul> </li> <li>■ Call Rule engine to validate the account, which will output "Success" or "Failure". <ul style="list-style-type: none"> <li>• If for any of the account validation status = "Failure", <ul style="list-style-type: none"> <li>◦ Set set-off status and exclude reason as "Not eligible".</li> </ul> </li> </ul> </li> </ul> <p>Once all target accounts have been processed get sum of debit amounts for all target accounts with set-of status as "Pending". Three scenarios are possible</p> <ul style="list-style-type: none"> <li>■ There are no target accounts in pending status. Go to cancel set-off step</li> <li>■ Sum of Debit amounts of target account &gt; Overdue amount for delinquent account. In this case check the "Excess debit" option</li> </ul> |

|                  |  |
|------------------|--|
|                  | <ul style="list-style-type: none"> <li>• Adjust Debit Amounts - Proportionately reduce debit amounts from all target accounts. See example at bottom of section.</li> <li>• Cancel Set-off - Go to Cancel set-off step</li> </ul> <ul style="list-style-type: none"> <li>■ Sum of Debit amounts of target account <math>\leq</math> Overdue amount for delinquent account. In this case there is no exception and set-off process should proceed.</li> </ul> <p>Cancel Set-off</p> <ul style="list-style-type: none"> <li>■ Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters)</li> </ul> <p>Example of proportionate adjustment:</p> <p>Say A1 is delinquent account and has \$ 120 as arrear. Say debit amounts of \$ 60 and \$40 have been set from target accounts TA1 and TA2. So total amount to be debited is \$ 100</p> <p>Now during revalidation it is found that overdue has dropped to \$ 60. So now below computations should be done</p> <ul style="list-style-type: none"> <li>■ <math>X = (\text{Overdue amount}) / (\text{Sum of debit amounts})</math></li> <li>■ New Amount to debited from TA1 = Previous debit amount for TA1 * X</li> <li>■ New Amount to debited from TA2 = Previous debit amount for TA2 * X</li> </ul> <p>So in this case</p> <ul style="list-style-type: none"> <li>■ <math>X = \\$ 60 / (\\$60 + \\$ 40) = 0.6</math></li> <li>■ New Amount to debited from TA1 = <math>\\$ 60 * 0.6 = \\$ 36</math></li> <li>■ New Amount to debited from TA2 = <math>\\$ 40 * 0.6 = \\$ 24</math></li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>■ Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status.</li> <li>■ Cancel Reason Char Value - Characteristic value for the defined characteristic type.</li> <li>■ Validation Failure Transition Status - Case transition status in case of validation failure.</li> <li>■ Excess Debit Option - Can have value ADJUST_DEBIT_AMOUNTS(Adjust Debit Amounts) or CANCEL_SETOFF(Cancel Set-off).</li> <li>■ Minimum Residual Amount - Minimum amount that must be present in account after set-off</li> <li>■ Rule ID - Defined rule id to validate account. Rule should return output validation status in fact 'SuccessOrFailure', which can have value true or false.</li> <li>■ Casa Account Type Identifier List - Comma separated savings account(CASA) identifiers.</li> <li>■ Td Account Type Identifier List - Comma separated term Deposit account(TD) identifiers.</li> </ul> |
| <b>Algorithm</b> | Case Type - Enter Status   |

|                     |   |
|---------------------|---|
| <b>Entity</b>       |   |
| <b>Program Type</b> | Java  |
| <b>Program Name</b> | com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RevalidateTarget Account   |
| <b>Parameter s</b>  | <p><b>Name:</b> Cancel Reason Char Value<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Cancel Reason Char Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Type</p> <p><b>Name:</b> Validation Failure Transition Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Failure Transition Status</p> <p><b>Name:</b> Excess Debit Option<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Excess Debit Option</p> <p><b>Name:</b> Minimum Residual Amount<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Minimum Residual Amount</p> <p><b>Name:</b> Rule Id<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Rule Id</p> <p><b>Name:</b> Casa Account Type Identifier List<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Casa Account Type Identifier List</p> <p><b>Name:</b> Td Account Type Identifier List<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Td Account Type Identifier List</p> |

## 5.85 Initiate LMI C1-INITLMIS

Table 5–182 Initiate LMI. C1-INITLMIS

|                    |              |
|--------------------|--------------|
| <b>Description</b> | Initiate LMI |
| <b>Detailed</b>    | Initiate LMI |

|                         |   |
|-------------------------|---|
| <b>Description</b>      |   |
| <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.Initiate_LMIProcess   |
| <b>Parameters</b>       | <p><b>Name:</b> No LMI Option<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> No LMI Option</p> <p><b>Name:</b> LMI Insurer Code<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> LMI Insurer Code</p> <p><b>Name:</b> Initiate LMI Options<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Initiate LMI Options</p> <p><b>Name:</b> LMI Case Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> LMI Case Type</p> <p><b>Name:</b> Balance Threshold<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Balance Threshold</p> |

## 5.86 NGP Collection case creation algorithm C1-COLLCASE

Table 5–183 NGP Collection case creation algorithm. C1-COLLCASE

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | NGP Collection case creation algorithm  |
| <b>Detailed Description</b> | This is overdue monitor Rule algorithm used for NGP Collection Case creation. It will be invoked through the over due monitor batch process C1-ADMOV. |
| <b>Algorithm Entity</b>     | Collection Class Overdue Rules – Overdue Monitor Rule   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.CollectionCaseCreationOverdueMonitorRuleAlgo  |
| <b>Parameters</b>           | <b>Name:</b> New Case Creation Rule Id<br><b>Required (Yes/No):</b> Yes   |



|  |  |
|--|--|
|  | <p><b>Description:</b> New Case Creation Rule Id</p> <p><b>Name:</b> Exist Case Creation Rule Id</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Exist Case Creation Rule Id</p> |
|--|--|

## 5.87 Stop Contract Algorithm C1-CONTSTOP

Table 5–184 Stop Contract Algorithm. C1-CONTSTOP

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Stop Contract Algorithm   |
| <b>Detailed Description</b> | This algorithm will stop the contract linked to case in the CI_CASE_PARTY table.    |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.CaseEnterStatusContractStopAlgoComp |
| <b>Parameters</b>           | NA  |

## 5.88 Check for existing Hardship C1-CHKHRDSHP

Table 5–185 Check for existing Hardship. C1-CHKHRDSHP

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check for existing Hardship  |
| <b>Detailed Description</b> | <p>Before creating case in Pending state. This Algorithm checks, if there is any active case of Hardship case type (By Retrieving case type code from feature configuration). If yes, It Display message ' Party is already in Hardship'</p> <p>If no,</p> <p>It will proceed with case creation.</p> <p>This checks for an existing Hardship application for the party.</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.financialHardship.CheckExistingHardship  |
| <b>Parameters</b>           | <p><b>Name:</b> Hardship Case Type Feature Config</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Hardship Case Type Feature Config</p>  |

## 5.89 Algorithm for contact processing C1-CNTCT

Table 5–186 Algorithm for contact processing. C1-CNTCT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm for contact processing.                                |
| <b>Detailed Description</b> | Algorithm for contact processing                                 |
| <b>Algorithm Entity</b>     | Customer Contact – Action Algorithm                              |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.contacthistory.ContactProcessing |
| <b>Parameters</b>           | NA   |

## 5.90 Check application expiry date C1-CHKEXP

Table 5–187 Check application expiry date. C1-CHKEXP

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check application expiry date  |
| <b>Detailed Description</b> | Check application expiry date with allowed minimum date of nominated account and posting date.                                   |
| <b>Algorithm Entity</b>     | Business Object – Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckApplicationExpiryDate                     |
| <b>Parameters</b>           | <b>Name:</b> xpathForApplicationExpiryDate<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> xpathForApplicationExpiryDate |

## 5.91 New Customer Contact Creation Algorithm C1-CCCREATE

Table 5–188 New Customer Contact Creation Algorithm. C1-CCCREATE

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | New Customer Contact Creation Algorithm  |
| <b>Detailed Description</b> | This Algorithm Type is used to create Customer Contact on the basis of Customer Contact class, Customer Contact Type and Preferred Contact Method on a Customer Level case or an Account Level case. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.correspondence.CustomerContactCreation   |
| <b>Parameters</b>           | <b>Name:</b> Customer Contact Class  |

|  |  |
|--|--|
|  | <p><b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Customer Contact Class</p> <p><b>Name:</b> Customer Contact Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Customer Contact Type</p> <p><b>Name:</b> Preferred Contact Method<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Preferred Contact Method</p> |
|--|--|

## 5.92 Removes a case characteristic on case status exit C1-REMCSCH

*Table 5–189 Removes a case characteristic on case status exit. C1-REMCSCH*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Removes a case characteristic on case status exit.  |
| <b>Detailed Description</b> | This algorithm type removes a case characteristic with char type = parameter 10 value.                                  |
| <b>Algorithm Entity</b>     | Case Type – Exit Status   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.RemoveCaseCharacteristic   |
| <b>Parameters</b>           | <p><b>Name:</b> Characteristic Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Characteristic Type</p> |

## 5.93 Transition case on a date on a case characteristic C1-TRANSDT

*Table 5–190 Transition case on a date on a case characteristic. C1-TRANSDT*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Transition case on a date on a case characteristic  |
| <b>Detailed Description</b> | This algorithm type transitions the case on a date stored on a case characteristic (char type = parameter 10 value). If the case characteristic is not found, the case will be transitioned on the current date. This algorithm type accepts parameters Next Status or Next Transition Condition to determine the next status |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.TransitionToNextStatusOnDate   |
| <b>Parameters</b>           | <p><b>Name:</b> Characteristic Type<br/> <b>Required (Yes/No):</b> Yes</p>  |

|  |  |
|--|--|
|  | <p><b>Description:</b> Characteristic Type</p> <p><b>Name:</b> Next Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Status</p> <p><b>Name:</b> Next Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Transition Condition</p> |
|--|--|

## 5.94 Set Account Nxt Credit Review Date to current date C1-NXTRVWDT

Table 5–191 Set Account Nxt Credit Review Date to current date. C1-NXTRVWDT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Set Account Nxt Credit Review Date to current date                                   |
| <b>Detailed Description</b> | This algorithm sets the accounts next credit review date to current date.            |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.SetAccountNextCreditReviewDateToCurrentDate |
| <b>Parameters</b>           | NA   |

## 5.95 Mark accounts for strategy review C1-REVIW-ACT

Table 5–192 Mark accounts for strategy review. C1-REVIW-ACT

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Mark accounts for strategy review  |
| <b>Detailed Description</b> | This algorithm will mark all accounts that are "in-collections" for the customer in hardship for review.         |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | ccom.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.MarkAccountsForStrategyReview |
| <b>Parameters</b>           | NA   |

## 5.96 Wait Time Out (in days) C1-WAIT-DAYS

Table 5–193 Wait Time Out (in days). C1-WAIT-DAYS

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Wait Time Out (in days)   |
| <b>Detailed Description</b> | This algorithm times out when the Case has been on the state for too long.  |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.WaitTimeOut  |
| <b>Parameters</b>           | <p><b>Name:</b> Expiration Days<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Expiration Days</p> <p><b>Name:</b> Xpath To Expiration Days<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Xpath To Expiration Days</p> <p><b>Name:</b> Time Out To Do Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Time Out To Do Type</p> <p><b>Name:</b> Log Entry Char Type Fk To To Do<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Log Entry Char Type Fk To To Do</p> <p><b>Name:</b> Work Calendar<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Work Calendar</p> |

## 5.97 Validate Hardship Application inputs C1-V-FH-APP

Table 5–194 Validate Hardship Application inputs. C1-V-FH-APP

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate Hardship Application inputs  |
| <b>Detailed Description</b> | This algorithm validates that all the mandatory fields on the Hardship Application Form are populated.              |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipApplicationInputs |

|                   |    |
|-------------------|----|
| <b>Parameters</b> | NA |
|-------------------|----|

## 5.98 Check for Operational Relief Types C1-OP-RT

*Table 5–195 Check for Operational Relief Types .C1-OP-RT*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Check for Operational Relief Types   |
| <b>Detailed Description</b> | This algorithm checks if any of the identified stp relief types need to be operational.  |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckForOperationalReliefTypes   |
| <b>Parameters</b>           | <p><b>Name:</b> Operational Relief Type 1<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 1</p> <p><b>Name:</b> Operational Relief Type 2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 2</p> <p><b>Name:</b> Operational Relief Type 3<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 3</p> <p><b>Name:</b> Operational Relief Type 4<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 4</p> <p><b>Name:</b> Operational Relief Type 5<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 5</p> <p><b>Name:</b> Operational Relief Type 6<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Operational Relief Type 6</p> <p><b>Name:</b> Operational Relief Type 7<br/><b>Required (Yes/No):</b> No</p> |

|  |   |
|--|---|
|  | <p><b>Description:</b> Operational Relief Type 7</p> <p><b>Name:</b> Operational Relief Type 8</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Operational Relief Type 8</p> <p><b>Name:</b> Operational Relief Type 9</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Operational Relief Type 9</p> <p><b>Name:</b> Operational Relief Type 10</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Operational Relief Type 10</p> |
|--|---|

## 5.99 Auto-Approval Check C1-FH-AUTOAP

Table 5–196 Auto-Approval Check. C1-FH-AUTOAP

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Auto-Approval Check   |
| <b>Detailed Description</b> | This algorithm invokes an Application service which in turn invokes host service which determines if the Hardship application can be auto-approved. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AutoApprovalCheck   |
| <b>Parameters</b>           | <p><b>Name:</b> Rule Type Code</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Rule Type Code</p>                                   |

## 5.100 Apply Hardship Relief Types for accounts in Host C1-FH-EVAL

Table 5–197 Apply Hardship Relief Types for accounts in Host. C1-FH-EVAL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Apply Hardship Relief Types for accounts in Host                           |
| <b>Detailed Description</b> | This algorithm applies hardship relief types for the accounts in the host. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |

|                     |   |
|---------------------|---|
| <b>Program Type</b> | Java  |
| <b>Program Name</b> | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ApplyHardshipReliefTypes  |
| <b>Parameters</b>   | <p><b>Name:</b> Xpath To Completion Flag<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Xpath To Completion Flag</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Exception Transition Condition</p> |

## 5.101 Update Party Warning Indicator C1-UPD-PRTWI

Table 5–198 Update Party Warning Indicator. C1-UPD-PRTWI

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Party Warning Indicator  |
| <b>Detailed Description</b> | <p>This is a generic algorithm that will make a service call to host to update Party level warning indicators for Main Customer.</p> <p>If a Rule Type Code is populated, it will first invoke the rule to determine if the Warning Indicator should be updated.</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.financialHardship.UpdatePartyWarningIndicator   |
| <b>Parameters</b>           | <p><b>Name:</b> Warning Indicator Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Warning Indicator Type</p> <p><b>Name:</b> Warning Indicator Value<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Warning Indicator Value</p> <p><b>Name:</b> Rule Type Code<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Rule Type Code</p> <p><b>Name:</b> Exception Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Exception Transition Condition</p> |



## 5.102 Transition to Next status when all reliefs are app C1-RAPP

**Table 5–199 Transition to Next status when all reliefs are app. C1-RAPP**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to Next status when all reliefs are app   |
| <b>Detailed Description</b> | This is algorithm that will transition the case to the next status when all reliefs have been applied.   |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.TransitionToNextStatusWhenAllReliefsApplied  |
| <b>Parameters</b>           | <p><b>Name:</b> Next Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Status Code</p> <p><b>Name:</b> Next Transition Condition<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Next Transition Condition</p> |

## 5.103 Collection - Entity Activity Population. C1-ENTACTPOP

**Table 5–200 Collection - Entity Activity Population. C1-ENTACTPOP**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collection - Entity Activity Population  |
| <b>Detailed Description</b> | <p>This sample algorithm is called from various entities classes for population of Account Activity.</p> <p>The algorithm takes following input parameters:</p> <ol style="list-style-type: none"> <li>1. Entity Type : Person/Account for which activity is getting created (e.g. Case can be created on Person as well as Account)</li> <li>2. Entity Id : Person/Account Id</li> <li>3. ModeOfOperation: Add/Update/Delete/Cancel</li> <li>4. HostEntityId: Activity Entity Id (e.g PTP/CC/Follow-up/Case Id)</li> <li>5. HostEntityName: PTP/CC/FOLLOWUP/CASE</li> </ol> |
| <b>Algorithm Entity</b>     | Installation – Entity Activity Populate  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseCreation.PopulateAccountActivityAlgo   |
| <b>Parameters</b>           | NA   |

## 5.104 Cancel Hardship Application C1-CXLFH

Table 5–201 Cancel Hardship Application. C1-CXLFH

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Cancel Hardship Application   |
| <b>Detailed Description</b> | This algorithm will make a service call to host to cancel an active Hardship Application.                   |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelHardshipApplication |
| <b>Parameters</b>           | NA  |

## 5.105 Perform Payment Transfer for ROSO C1-ROSOPMTXR

Table 5–202 Perform Payment Transfer for ROSO. C1-ROSOPMTXR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Perform Payment Transfer for ROSO   |
| <b>Detailed Description</b> | This Algorithm Type will call a web service which calls Oracle NGP Core Banking to perform a payment transfer between an eligible delinquent Account and eligible Target Account(s).  |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.rightofSetOff.PerformPaymentXferForROSO  |
| <b>Parameters</b>           | <p><b>Name</b> : Exception Transition Condition<br/> <b>Required (Yes/No)</b>: Yes<br/> <b>Description</b>: Exception Transition Condition</p> <p><b>Name</b>: Host Code<br/> <b>Required (Yes/No)</b>: Yes<br/> <b>Description</b>: Host Code</p> <p><b>Name</b> : Failed Leg Notification To Do Type<br/> <b>Required (Yes/No)</b>: Yes<br/> <b>Description</b>: Failed Leg Notification To Do Type</p> |

## 5.106 Validate ROSO Target Account inputs C1-RS-VALIN

Table 5–203 Validate ROSO Target Account inputs C1-RS-VALIN

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Validate ROSO Target Account inputs   |
| <b>Detailed Description</b> | This Algorithm Type will validate the user inputs entered into the Target Account dynamic panel to ensure they comply with the business rules.<br>If the inputs are not valid, the Case will transition back to the previous status and prompt the user to re-enter the inputs. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.rightofSetOff. ValidateRosolnputs  |
| <b>Parameters</b>           | <p><b>Name</b> : Exception Transition Condition<br/> <b>Required (Yes/No)</b>: No<br/> <b>Description</b>: Exception Transition Condition</p> <p><b>Name</b>: Exception Status<br/> <b>Required (Yes/No)</b>: No<br/> <b>Description</b>: Exception Status</p>                  |

## 5.107 Create RMB Entities from Host Data C1-VCREATE

Table 5–204 Create RMB Entities from Host Data. C1-VCREATE

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Create RMB Entities from Host Data   |
| <b>Detailed Description</b> | <p>Create RMB Entities such as Person, Account ,Account Person, PartyCollect etc from Host Data.<br/> Input parameters:</p> <ol style="list-style-type: none"> <li>1. Source Host Id : Host Identifier Value e.g. NGP -- Removed in R2.2- Host Id will come from UI</li> <li>2. Inapplicable Statuses : Comma separated Host System Statuses for Account (host_sys_acct_stat_flg)</li> <li>3. Exclude Accrual Status Flag: Comma separated Accrual Statuses for Account (accl_stat_flg )</li> <li>4. Exclude Asset Class Code: Comma separated Asset Class Codes for Account (asst_class_cd )</li> <li>5. Exclude User Defined Acct Status: Comma separated User Defined Account Status (usr_def_acct_stat_flg)</li> <li>6. Exclude Offer Id: Comma separated Offer Id (offer_id)</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.AdhocCollectionEntityCreation   |

|                   |   |
|-------------------|---|
| <b>Parameters</b> | <b>Name:</b> Inapplicable Account Statuses For Hardship<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Inapplicable Account Statuses For Hardship |
|                   | <b>Name:</b> Exclude Accrual Status Flag<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Exclude Accrual Status Flag                               |
|                   | <b>Name:</b> Exclude Asset Class Code<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Exclude Asset Class Code                                     |
|                   | <b>Name:</b> Exclude User Defined Acct Status Flg<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Exclude User Defined Acct Status Flg             |
|                   | <b>Name:</b> Offer Id<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Offer Id   |

## 5.108 Populate Activity Table For Notes Creation C1-NTACTIVITY

*Table 5–205 Populate Activity Table For Notes Creation. C1-NTACTIVITY*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Populate Activity Table For Notes Creation                     |
| <b>Detailed Description</b> | Populate Activity Table For Notes Creation                     |
| <b>Algorithm Entity</b>     | Business Object – Post Processing                              |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.PopulateAccountactivityForNote |
| <b>Parameters</b>           | NA   |

## 5.109 Suspend Activity for Account Pre Processing C1-SPATACPRE

*Table 5–206 Suspend Activity for Account Pre Processing C1-USRALCRR*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Suspend Activity for Account Pre Processing |
| <b>Detailed Description</b> | Suspend Activity for Account Pre Processing |
| <b>Algorithm Entity</b>     | Business Object – Pre-Processing            |

|                     |   |
|---------------------|---|
| <b>Program Type</b> | Java  |
| <b>Program Name</b> | com.splwg.ccb.domain.collection.suspendActivity.SuspendActivityPreProcessing_Impl |
| <b>Parameters</b>   | NA  |

## 5.110 Sample TAM Algorithm Type C1-TAMALG

Table 5–207 Sample TAM Algorithm Type. C1-TAMALG

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Sample TAM Algorithm Type  |
| <b>Detailed Description</b> | This algorithm will update account and TAM review date for case.   |
| <b>Algorithm Entity</b>     | Case Type – Treatment Activity Monitor   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.TreatmentActivityMonitorAlgoComp   |
| <b>Parameters</b>           | <p><b>Name:</b> acctReviewDays<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Account Review Days</p> <p><b>Name:</b> tamReviewDays<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> TAM Review Days</p> |

## 5.111 Cancel Approval Request C1-CANAPPR

Table 5–208 Cancel Approval Request. C1-CANAPPR

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Cancel Approval Request   |
| <b>Detailed Description</b> | <p>This algorithm will cancel all pending approval requests for the case.<br/> Example for parameter values for legal Process:<br/> Composite Name:- com.ofss.fc.workflow.process.LegalProcessForApproval<br/> Instance Title:- LEGAL_CASE_<br/> Value of the above parameters are depends upon the SOA approval work flow.</p> |
| <b>Algorithm Entity</b>     | Case Type – Exit Status   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.CancelApprovalReqAlgo   |
| <b>Parameters</b>           | <p><b>Name:</b> Composite Name<br/> <b>Required (Yes/No):</b> Yes</p>   |

|  |   |
|--|---|
|  | <p><b>Description:</b> Composite Name</p> <p><b>Name:</b> Instance Title</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Instance Title</p> |
|--|---|

## 5.112 Update Customer in collections flag C1-CUSINCOL

Table 5–209 Update Customer in collections flag. C1-CUSINCOL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Customer in collections flag  |
| <b>Detailed Description</b> | <p>This algorithm will set or reset in-collections flag for the customer in core banking<br/>While setting the Flag</p> <ul style="list-style-type: none"> <li>■ Skip the process if flag is already set</li> <li>■ If not, set the in-collection flag for the customer to "Y" in the host.</li> </ul> <p>While resetting the Flag,</p> <ul style="list-style-type: none"> <li>■ Skip the process if flag is already reset</li> <li>■ System should check for all cases to which the customer is associated (as primary or secondary entity).</li> <li>■ If any of the cases for the customer belong to a specific category, then system should not reset the flag. List of categories to be checked will be set as parameters to this process.</li> <li>■ If none of the cases for the customer are from those categories, set the in-collection</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.UpdateCustomerInColl Flg   |
| <b>Parameters</b>           | <p><b>Name:</b> Update Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Update Type</p> <p><b>Name:</b> Case Category1<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Case Category1 Code</p> <p><b>Name:</b> Case Category2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Case Category2 Code</p>  |

|  |   |
|--|---|
|  | <p><b>Name:</b> Case Category3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Case Category3 Code</p> <p><b>Name:</b> Case Category4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Case Category4 Code</p> <p><b>Name:</b> Case Category5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Case Category5 Code</p> |
|--|---|

## 5.113 Set Display Date C1-SETDSPDT

*Table 5–210 Set Display Date. C1-SETDSPDT*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Set Display Date  |
| <b>Detailed Description</b> | This process will update the display date for the account.<br>New display date will be computed as = Current display date + offset days<br>If a display date is already present on the account, it should be updated only if new display date is < existing display date. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.SetDisplayDate  |
| <b>Parameters</b>           | <b>Name:</b> Offset Dayes<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Offset Dayes  |

## 5.114 Transition to Default next status after N Days C1-TRNDFLT

*Table 5–211 Transition to Default next status after N Days. C1-TRNDFLT*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Transition to Default next status after N Days  |
| <b>Detailed Description</b> | Transition the case to default next status after specific days. Days will be set as parameter for the process.<br>Case should transition to Default next status if, difference in current date and date of entry into current status is >= specified number of days |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program</b>              | Java  |

|                     |   |
|---------------------|---|
| <b>Type</b>         |   |
| <b>Program Name</b> | com.splwg.ccb.domain.collection.caseType.specialisedCollections.DefaultNextStatusAutoTransition |
| <b>Parameters</b>   | <b>Name:</b> Wait Days<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Wait Days        |

## 5.115 Check current cases on account for exclusion C1-EXCLCASE

*Table 5–212 Check current cases on account for exclusion. C1-EXCLCASE*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Check current cases on account for exclusion  |
| <b>Detailed Description</b> | <p>System should maintain a lookup with list of case categories for set-off exclusion. Processing logic should be as below-</p> <ul style="list-style-type: none"> <li>■ Get all active cases for the account. Account can be primary or secondary entity for that case.</li> <li>■ Get case categories for all these cases</li> <li>■ If the case category for any of the cases is from the exclusion list, validation should fail.</li> <li>■ Check Validation failure option <ul style="list-style-type: none"> <li>• Validation failure option = Fail case creation/transition. Case should not get created or should not transition status</li> <li>• Validation failure option = Transition status. Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters)</li> </ul> </li> <li>■ If the case category for any of the cases is not from the exclusion list, validation is successful and process should move to next step.</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>■ Cancel Reason Char Type: Characteristic type to set as case characteristic if validation failure option is transition status.</li> <li>■ Cancel Reason Char Value: Characteristic value for the defined characteristic type.</li> <li>■ Validation Failure Transition Status: Case transition status in case of validation failure.</li> <li>■ Validation Failure Option: This option is use to determine action to be taken in case of validation failure. Permissible values are FAIL_CASE_CREATION(fail case creation) and TRANSITION_STATUS(transition status).</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program</b>              | Java  |



|                     |  |
|---------------------|--|
| <b>Type</b>         |  |
| <b>Program Name</b> | com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.AccountExclusion  |
| <b>Parameters</b>   | <p><b>Name:</b> Cancel Reason Char Value<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Value</p> <p><b>Name:</b> Cancel Reason Char Type<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Cancel Reason Char Type</p> <p><b>Name:</b> Validation Failure Transition Status<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Validation Failure Transition Status</p> <p><b>Name:</b> Validation Failure Option<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Validation Failure Option</p> |

## 5.116 Update Collateral Status in the host C1-UPCOLLST

Table 5–213 Update Collateral Status in the host. C1-UPCOLLST

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Collateral Status in the host  |
| <b>Detailed Description</b> | Update Collateral Status in the hostdate of entry into current status is >= specified number of days.   |
| <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 Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> To Do Type</p> <p><b>Name:</b> Collateral Status<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Collateral Status</p> |

## 5.117 Initiate collateral valuation C1-COLLVAL

Table 5–214 Initiate collateral valuation. C1-COLLVAL

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Initiate collateral valuation   |
| <b>Detailed Description</b> | Initiate collateral valuation   |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation   |
| <b>Parameters</b>           | <p><b>Name:</b> To Do Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> To Do Type</p> <p><b>Name:</b> Days Since Closure Of Last To Do<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Days Since Closure Of Last To Do</p> <p><b>Name:</b> Assessment Expiry Days<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Assessment Expiry Days</p> |

## 5.118 Mandatory characteristics check for Asset Repo C1-CHARVAL

Table 5–215 Mandatory characteristics check for Asset Repo. C1-CHARVAL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Mandatory characteristics check for Asset Repo   |
| <b>Detailed Description</b> | Mandatory characteristics check for Asset Repo   |
| <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> Reference Characteristic Value<br/> <b>Required (Yes/No):</b> Yes</p>              |

|  |   |
|--|---|
|  | <p><b>Description:</b> Reference Characteristic Value</p> <p><b>Name:</b> Reference Characteristic<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Reference Characteristic</p> <p><b>Name:</b> Case Characteristic5<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Case Characteristic5</p> <p><b>Name:</b> Case Characteristic4<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Case Characteristic4</p> <p><b>Name:</b> Case Characteristic3<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Case Characteristic3</p> <p><b>Name:</b> Case Characteristic2<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Case Characteristic2</p> <p><b>Name:</b> Case Characteristic1<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Case Characteristic1</p> |
|--|---|

## 5.119 Update Collateral Status in the Host C1-UPCOLLSTS

Table 5–216 Update Collateral Status in the Host. C1-UPCOLLSTS

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Update Collateral Status in the Host  |
| <b>Detailed Description</b> | Update Collateral Status in the Host  |
| <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 Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> To Do Type</p>       |

|  |  |
|--|--|
|  | <b>Name:</b> Collateral Status<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Collateral Status |
|--|--|

## 5.120 Set exclusion date for delinquent account. C1-EXCLROSO

Table 5–217 Set exclusion date for delinquent account. C1-EXCLROSO

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Set exclusion date for delinquent account   |
| <b>Detailed Description</b> | <p>This process will set set-off exclusion date for the delinquent account. Processing will be driven by parameters set for the process.</p> <p>Set-off Exclusion date should be updated only if current exclusion date is &lt;= business date. Else, skip all below processing</p> <p>If Cancel Reason char type parameters is not bank</p> <ul style="list-style-type: none"> <li>■ Get the value for the specified char type</li> <li>■ This char type should be used to get the offset days from the Lookup for set-off exclusion days</li> <li>■ Set-off exclusion date should be set as current business days + offset days.</li> <li>■ If mapping for the reason is not found, default value for offset days should be used.</li> </ul> <p>If Cancel Reason char type parameters is blank but Reason code is provided</p> <ul style="list-style-type: none"> <li>■ Get the corresponding offset days from the lookup for the Reason code</li> <li>■ Set-off exclusion date should be set as current business days + offset days</li> <li>■ If mapping for the reason is not found, default value for offset days should be used.</li> </ul> <p>Parameters:</p> <ul style="list-style-type: none"> <li>■ Default Offset - Number of days to add to the set-off exclusion date.</li> <li>■ Reason Code - Code to fetch offset days from lookup.</li> <li>■ Cancel Reason - Characteristic type code to fetch offset days from lookup.</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.roso.UpdateSetoffExclusionDate  |
| <b>Parameters</b>           | <b>Name:</b> Default Offset<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Default Offset  |

|  |   |
|--|---|
|  | <p><b>Name:</b> Reason Code<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Reason Code</p> <p><b>Name:</b> Cancel Reason<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Cancel Reason</p> |
|--|---|

## 5.121 Cancel Set-off C1-CANROSO

*Table 5–218 Cancel Set-off. C1-CANROSO*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Cancel Set-off  |
| <b>Detailed Description</b> | This algorithm will update the set-off status as "Cancelled" for target accounts associated to the case and having set-off status as "Pending". |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CancelSetoff   |
| <b>Parameters</b>           | NA  |

## 5.122 Complete Set-off C1-COMPROSO

*Table 5–219 Complete Set-off. C1-COMPROSO*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Complete Set-off   |
| <b>Detailed Description</b> | <p>This algorithm transitions the case to complete.<br/> Processing Logic will be as below</p> <ul style="list-style-type: none"> <li>■ Validate that at least one of the target account has set-off status = "Success" and Reversed Flag = "N".</li> <li>■ If above validation fails transition to complete should not be allowed and To-do of given To-do Type should be created.</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.roso.CompleteSetoff  |
| <b>Parameters</b>           | NA   |

## 5.123 Reverse Set-off C1-REVROSO

Table 5–220 Reverse Set-off. C1-REVROSO

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Reverse Set-off  |
| <b>Detailed Description</b> | <p>This algorithm transitions the case to Reversed status.<br/>Processing Logic will be as below</p> <ul style="list-style-type: none"> <li>■ Validate below for each target account <ul style="list-style-type: none"> <li>• Set-off status is not "Success" or If set-off status is "Success" then Reversed Flag should be "Y".</li> <li>• There should be at least one account with Reversed Flag as "Y".</li> </ul> </li> <li>■ If above validation fails, transition to this status should not be allowed.</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.roso.ReverseSetoff   |
| <b>Parameters</b>           | NA   |

## 5.124 Algorithm type for update case id for Insurance C1-UPCASFINS

Table 5–221 Algorithm type for update case id for Insurance. C1-UPCASFINS

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm type for update case id for Insurance  |
| <b>Detailed Description</b> | Algorithm type for update case id for Insurance  |
| <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.UpdateInsuranceCaseDetails |
| <b>Parameters</b>           | NA   |

## 5.125 Case Creation on enter processing C1-CCOENTER

Table 5–222 Case Creation on enter processing. C1-CCOENTER

|                    |  |
|--------------------|--|
| <b>Description</b> | Case Creation on enter processing  |
| <b>Detailed</b>    | This Algorithm will create a new case for the given Case Type on enter processing. |

|                         |  |
|-------------------------|--|
| <b>Description</b>      |  |
| <b>Algorithm Entity</b> | Case Type – Enter Status   |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnEnterAlgo       |
| <b>Parameters</b>       | <b>Name:</b> Case Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Type |

## 5.126 Collection - Case Creation On Exit of Status C1-CCOE

*Table 5–223 Collection - Case Creation On Exit of Status. C1-CCOE*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collection - Case Creation On Exit of Status   |
| <b>Detailed Description</b> | This algorithm will create a case on the exit processing of the status.<br>This algorithm will create a case for the account in context and the provided Case type soft parameter. |
| <b>Algorithm Entity</b>     | Case Type – Exit Status  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnExitAlgo  |
| <b>Parameters</b>           | <b>Name:</b> Case Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Case Type   |

## 5.127 Action category Validation algorithm C1-ACTCAT

*Table 5–224 Action category Validation algorithm. C1-ACTCAT*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Action category Validation algorithm.   |
| <b>Detailed Description</b> | Action category Validation algorithm. This algorithm checks that there should be atleast on action category entity on it. |
| <b>Algorithm Entity</b>     | Business Object – Validation  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionCategory.ActionCategoryValidation                                      |
| <b>Parameters</b>           | NA  |

## 5.128 Action Type Algorithm Type C1-ACTTYP

Table 5–225 Action Type Algorithm Type. C1-ACTTYP

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Action Type Algorithm Type   |
| <b>Detailed Description</b> | Action Type Algorithm Type   |
| <b>Algorithm Entity</b>     | Business Object – Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionType.ActionTypeResultTypeValidation |
| <b>Parameters</b>           | NA   |

## 5.129 Case Type Status Mapping Algorithm Type C1-CASETYMP

Table 5–226 Case Type Status Mapping Algorithm Type C1-CASETYMP

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Case Type Status Mapping Algorithm Type  |
| <b>Detailed Description</b> | Case Type Status Mapping Algorithm Type  |
| <b>Algorithm Entity</b>     | Business Object – Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.caseTypeMapping.CaseTypeMappingValidation |
| <b>Parameters</b>           | NA   |

## 5.130 Collection - Close Processing Algorithm C1-CCALG

Table 5–227 Collection - Close Processing Algorithm. C1-CCALG

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collection - Close Processing Algorithm  |
| <b>Detailed Description</b> | <p>This algorithm will perform processing done when a Pending Stop Contract is picked up by the Overdue Monitor (collection is to be closed for an account).</p> <ul style="list-style-type: none"> <li>■ It will update the financial balance of the Contract to zero through an adjustment.</li> </ul> |



|                         |  |
|-------------------------|--|
|                         | <ul style="list-style-type: none"> <li>■ Check if there is one or more active promise to pay for the account, if it does it will update the promise to pay status to cancelled and provides the cancel reason</li> <li>■ If it is required to close any cases, then it will check if the case has a next status in a final status and if it does will transition to that state. If the case has multiple next statuses which are final statuses, then it will use the default final status defined in the algorithm</li> </ul> <p>The following parameters are available and are required:</p> <ul style="list-style-type: none"> <li>■ Adjustment Type used for the adjustment created by this algorithm.</li> <li>■ Cancellation Reason Code used while canceling Active PTPs</li> <li>■ Is Closing Required Flag to specify if the cases associated have to be closed. If this flag is Y but one or more cases cannot be closed the algorithm will generate an error.</li> <li>■ Final Default Case Status - If the case to be closed has multiple next statuses that are final and the status specified in this parameter is one of those final statuses, the case will be moved to the status specified in this parameter.</li> </ul> |
| <b>Algorithm Entity</b> | Collection – Collection Case Processing  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseCreation.CollectionClosingAlgo   |
| <b>Parameters</b>       | <p><b>Name:</b> Adjustment Type<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Adjustment Type</p> <p><b>Name:</b> Cancellation Reason Code for PTP<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Cancellation Reason Code for PTP</p> <p><b>Name:</b> Is Closing Required Flag (Y/N)<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Is Closing Required Flag</p> <p><b>Name:</b> Final Default Case Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Final Default Case Status</p>  |

## 5.131 Algorithm type for case list update C1-CASELIST

Table 5–228 Algorithm type for case list update. C1-CASELIST

|                    |                                     |
|--------------------|-------------------------------------|
| <b>Description</b> | Algorithm type for case list update |
|--------------------|-------------------------------------|

|                             |  |
|-----------------------------|--|
| <b>Detailed Description</b> | Algorithm type for case list update or insert in CI_LIST_MODE_UPDATE table.                              |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardShipCaseListUpdate |
| <b>Parameters</b>           | NA   |

## 5.132 Copy Characteristics Algorithm C1-COPYCHAR

Table 5–229 Copy Characteristics Algorithm. C1-COPYCHAR

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Copy Characteristics Algorithm   |
| <b>Detailed Description</b> | Copy Characteristics Algorithm to copy the Characteristics of recently closed case of a particular Case Category to newly created Case of the same Case Category, when "CONTACT_ALT_SW" in CI_ACCT_EXTN table is set to "Y".         |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.CopyCharacteristicsOnCaseCreate  |
| <b>Parameters</b>           | <p><b>Name:</b> Case Category<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Case Category</p> <p><b>Name:</b> Characteristics List<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Characteristics List</p> |

## 5.133 Call Advice - Red/Green logic calculation C1-CALADVICE

Table 5–230 Call Advice - Red/Green logic calculation. C1-CALADVICE

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Call Advice - Red/Green logic calculation   |
| <b>Detailed Description</b> | <p>Call Advice - Red/Green logic calculation</p> <ul style="list-style-type: none"> <li>■ Call Advice will be 'Green' if</li> </ul> |

|                         |  |
|-------------------------|--|
|                         | <ul style="list-style-type: none"> <li>• 'Permission to Call' is Yes And</li> <li>• Current Time is within the State level Acceptable Time Limits And</li> <li>• Current Time is within the preferred times of the Customer And</li> <li>• Current Date is not within the Do Not Disturb Dates</li> </ul> <p>Else it is 'Red'.</p> |
| <b>Algorithm Entity</b> | Installation – Contact Information   |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.collectionLandingPage.ContactInformationCallAdviceAlgo   |
| <b>Parameters</b>       | NA   |

## 5.134 Task Case Mapping Validation Algorithm C1-TCVAL

Table 5–231 Task Case Mapping Validation Algorithm. C1-TCVAL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Task Case Mapping Validation Algorithm   |
| <b>Detailed Description</b> | <p>Task Case Mapping Validation Algorithm</p> <p>Algorithm will validate Repossession Date cannot be greater than future date for the process field mapped to Task Type Code and Case Type Code mentioned in soft parameters.</p> <p>This algorithm will validate the Repossession Date field only if value is already present.</p> <p>Validation Date can be SYSTEM DATE or POSTING DATE.</p> |
| <b>Algorithm Entity</b>     | Task Type Case Type Validation Algorithm Spot  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepoDateValidation   |
| <b>Parameters</b>           | <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p> <p><b>Name:</b> Case Type Code<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Case Type Code</p> <p><b>Name:</b> Task Type Code<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Task Type Code</p>  |

## 5.135 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status C1-DISMON

**Table 5–232 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status. C1-DISMON**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Monitoring Algorithm For Dispute Resolved to exit from Dispute Status.   |
| <b>Detailed Description</b> | <p>This Monitoring Algorithm exit the Dispute status and move into Contact Status 'x' days after the resolution date.<br/>Also it set Dispute Flag at account level to N when case exits this status.<br/>Below are the soft parameter example:</p> <ul style="list-style-type: none"> <li>■ Validation Date: This Validation Date will validate and compare the date with New Dispute Resolved Date (Date obtained after adding no. of grace days).<br/>It's value can be SYSTEM DATE or POSTING DATE.<br/>This is mandatory parameter.</li> <li>■ No Of Grace Days : This numeric parameter will add those number of days to Dispute Resolved Date. Here Dispute Resolved date is one which is captured during Dispute Resolved Follow Up.<br/>This is mandatory parameter</li> <li>■ Contact Status : Case will move to Contact status if Contact RM and Contact Alternate Flag is not present on account OR if case is not able to transition to contact rm and contact alternate status.</li> <li>■ Contact RM Status: Case will move to Contact Relation Manager status if relation manager exist for that account and Contact Rm status exist</li> <li>■ Contact Alternate Status : Case transition to contact alternate status if (RM does not exist on an account or Rm exist but Contact Rm status is not specified) and contact alternate switch on an account = Y</li> <li>■ Characteristic Code : Characteristic Type Code to be referred while registering dispute resolution follow up to set new Risk Indicator on Recovery Account.</li> <li>■ Characteristic Value : Characteristic Value Code to be referred while registering dispute resolution follow up to set new Risk Indicator on Recovery Account.</li> <li>■ Risk Indicator Code : Sets the Risk Indicator Code on Recovery Account based on Characteristic Code and Characteristic Value mentioned while Dispute Resolution.</li> </ul> |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateDisputeMonitor   |
| <b>Parameters</b>           | <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Validation Date</p>  |

|  |   |
|--|---|
|  | <p><b>Name:</b> No Of Grace Days<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> No Of Grace Days</p> <p><b>Name:</b> Contact Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Status</p> <p><b>Name:</b> Contact RM Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact RM Status</p> <p><b>Name:</b> Contact Alternate Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Alternate Status</p> <p><b>Name:</b> Characteristic Code<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Characteristic Code</p> <p><b>Name:</b> Characteristic Value<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Characteristic Value</p> <p><b>Name:</b> Risk Indicator Code<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Risk Indicator Code</p> |
|--|---|

## 5.136 SLA Parameters validation algorithm C1-SLAPARAM

*Table 5–233 SLA Parameters validation algorithm. C1-SLAPARAM*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | SLA Parameters validation algorithm  |
| <b>Detailed Description</b> | SLA Parameters validation algorithm created for Recovery 2.6.2 release. This algorithm to be called along with C1-SLABO. |
| <b>Algorithm Entity</b>     | Business Object – Post Processing  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.vendor.SLAParametersPostProcessAlgo  |
| <b>Parameters</b>           | NA   |

## 5.137 Case Group add validation algorithm C1-CGVAL

Table 5–234 Case Group add validation algorithm. C1-CGVAL

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Case Group add validation algorithm                                    |
| <b>Detailed Description</b> | Case Group add validation algorithm                                    |
| <b>Algorithm Entity</b>     | Business Object – Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseGroup.CaseGroupValidationAlgorithm |
| <b>Parameters</b>           | NA   |

## 5.138 Collection - Get Strategy Algorithm C1-COLGS

Table 5–235 Collection - Get Strategy Algorithm. C1-COLGS

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collection - Get Strategy Algorithm  |
| <b>Detailed Description</b> | <p>This algorithm calls the Rules Engine to determine a collection strategy. It is invoked by the Collection Class Overdue Rule - Overdue Monitor Rule</p> <p>The following parameters are passed to the Rules Engine :</p> <ul style="list-style-type: none"> <li>■ Rule Type (defined in the input parameter)</li> <li>■ Case Type (if any)</li> <li>■ Days Past Due</li> <li>■ Overdue Amount</li> <li>■ Collection Type</li> </ul> |
| <b>Algorithm Entity</b>     | Collection – Get Strategy Algorithm  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseCreation.GetStrategyAlgo   |
| <b>Parameters</b>           | <p><b>Name:</b> Rule Type</p> <p><b>Required (Yes/No):</b> Yes</p> <p><b>Description:</b> Rule Type</p>  |

## 5.139 Create/Move Collection Strategy Cases for Account C1-COLOMR

Table 5–236 Create/Move Collection Strategy Cases for Account. C1-COLOMR

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Create/Move Collection Strategy Cases for Account  |
| <b>Detailed Description</b> | <p>This overdue monitor rule algorithm is used to determine the appropriate case type to be used to create a case for an account in collections. It is also responsible for creating the case or for case movement.</p> <p>It will first check for the Collection events (contracts) that are under the account.</p> |

|                         |   |
|-------------------------|---|
|                         | <p>For Active Contracts it will call the Collection - Get Strategy Algorithm, to determine which Case Type should be used before creating a case. If one or more cases already exist for the Contract they may get closed and new cases created (case movement) if Collection - Get Strategy Algorithm indicates that the strategy need to be changed and the current cases can be closed. This algorithm also consider the feature configuration 'C1-NMCSTY' to determine the cases that should not be moved.</p> <p>For Pending Stop Contracts it will call the Collection - Close Processing Algorithm to move the Contract into a closed status. May also close the Cases attached to the contract and reduce the overdue amount on the contract to zero. All other SA statuses are ignored by this algorithm.</p> <p>Notes on the algorithm parameters</p> <ul style="list-style-type: none"> <li>■ Final Default Case Status - If the case to be closed have multiple next statuses that are final and the status specified in this parameter is one of those final statuses, the case will be moved to the status specified in this parameter.</li> <li>■ Is Closing required - Flag indicate whether case closing is required or not (Y/N)</li> <li>■ Collection Closing Algorithm - This is the algorithm code for Collection - Close Processing Algorithm</li> <li>■ Get Strategy Algorithm - This is the algorithm code for Collection - Get Strategy Algorithm</li> </ul> |
| <b>Algorithm Entity</b> | Collection Class Overdue Rules– Overdue Monitor Rule  |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseCreation.CaseOverdueMonitorRuleAlgo   |
| <b>Parameters</b>       | <p><b>Name:</b> Final Default Case Status<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Final Default Case Status</p> <p><b>Name:</b> Is Closing Required Flag (Y/N)<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Is Closing Required Flag (Y/N)</p> <p><b>Name:</b> Collection Closing Algorithm Code<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Collection Closing Algorithm Code</p> <p><b>Name:</b> Get Strategy Algorithm Code<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Get Strategy Algorithm Code</p>   |

## 5.140 Collection - Case SA Update for Manual Creation C1-CSAUPD

*Table 5–237 Collection - Case SA Update for Manual Creation. C1-CSAUPD*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Collection - Case SA Update for Manual Creation                      |
| <b>Detailed Description</b> | This Algorithm will update Case SA table for Manual Case Creation    |
| <b>Algorithm Entity</b>     | Business Object – Audit  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseCreation.CaseSaUpdateBoAuditAlgo |
| <b>Parameters</b>           | NA   |

## 5.141 Promise to Pay - Additional Grace Days Sample Algo C1-PPADDLGRD

*Table 5–238 Promise to Pay - Additional Grace Days Sample Algo. C1-PPADDLGRD*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Promise to Pay - Additional Grace Days Sample Algo   |
| <b>Detailed Description</b> | <p>This sample algorithm is called by the Promise to Pay Monitor; it takes the output, which represents additional grace days that should be added to a promise to pay's scheduled payment date.</p> <p>The algorithm takes the input parameter value and passes it back to the Promise to Pay Monitor as additional grace days.</p> |
| <b>Algorithm Entity</b>     | Installation – Additional Grace Days   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.payPlan.AdditionalGraceDaysCalculationAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> Additional Grace Days</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Additional Grace Days</p>   |

## 5.142 Promise to Pay Threshold Percentage C1-PPTHRESH

*Table 5–239 Promise to Pay Threshold Percentage. C1-PPTHRESH*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Promise to Pay Threshold Percentage  |
| <b>Detailed Description</b> | <p>This algorithm is called by the Pay Plan Monitor when an expected scheduled payment is not fully met. At this point the promise to pay has been marked to be broken.</p> <p>It receives the following inputs from the pay plan monitor</p> <ul style="list-style-type: none"> <li>■ Promise to Pay ID</li> <li>■ Total Amount Paid towards the promise to pay</li> <li>■ Date (Business Date - Grace Days)</li> </ul> |



|                         |   |
|-------------------------|---|
|                         | <ul style="list-style-type: none"> <li>Array of Promise to Pay Scheduled Payments balance</li> </ul> <p>The algorithm will check if the Total Amount Paid is within the threshold percentage (input parameter) of the Total Scheduled Payments expected.</p> <p>If the payments are within the threshold, then the algorithms returns a value of "Y" indicating the promise to pay that was set to be broken should be overridden and remain active/kept</p> <p>Else if the total payments are not within the threshold, then the algorithm returns a value of "N" indicating the promise to pay should be set to broken.</p> |
| <b>Algorithm Entity</b> | Installation – Payment Threshold Percentage   |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.payPlan.PaymentThresholdPercentageCalculationAlgorithm  |
| <b>Parameters</b>       | <p><b>Name:</b> Threshold Percentage</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Threshold Percentage</p>  |

## 5.143 Result type Post Processing Case Transition Algo C1-RTPCC

Table 5–240 Result type Post Processing Case Transition Algo. C1-RTPCC

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Result type Post Processing Case Transition Algo.  |
| <b>Detailed Description</b> | <p>If specified on the Result Type, this algorithm will be invoked when the corresponding result is recorded for a Case (Action/Result UI).</p> <p>This can be used to transition the case from the current status to the next possible status as follow,</p> <ul style="list-style-type: none"> <li>This algorithm has a parameter Output Status i.e. next possible status, so for case transition it will be checked whether Output Status is one of the next possible status, if YES, it will transiiton the case to that status</li> <li>This algorithm has a parameter Input Status, which will be checked against the current status of the Case. This is an optional parameter. If specified the Case transition will happen only when the current status of the case matches with this parameter.</li> </ul> |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcCaseTransAlgo  |
| <b>Parameters</b>           | <p><b>Name:</b> Output Status</p> <p><b>Required (Yes/No):</b> Yes</p>   |

|  |   |
|--|---|
|  | <p><b>Description:</b> Output Status</p> <p><b>Name:</b> Input Status</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Input Status</p> |
|--|---|

## 5.144 Inbound Customer algorithm C1-IN-CUST

Table 5–241 Inbound Customer algorithm. C1-IN-CUST

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Inbound Customer algorithm   |
| <b>Detailed Description</b> | <p>This algorithm will create the Person, Account, SA, SAcollection object and Adjustment from FACT clob. This is a Business Object Status Enter algorithm. The algorithm perform the following actions</p> <ul style="list-style-type: none"> <li>■ Retrieve the XML message containing the customer information, which stored on the FACT MO.</li> <li>■ Read the XML and determine if the action is to add a new customer or update an existing customer.</li> <li>■ It may create a combination of Person, Account, Contract, Contract, or Adjustment, depending on what was contained in the XML.</li> <li>■ If all objects are created successfully it will transition the lifecycle to the 'Created' status</li> <li>■ Else if any of the objects experienced an error while processing it will transition the lifecycle to the "Rejected" status.</li> </ul> <p>It has two parameters , both optional.</p> <ul style="list-style-type: none"> <li>■ Account Id Type identifies the Account Identifier Type used to locate the account in ORMB.</li> <li>■ Person Id Type identifies the Person Identifier Type used to locate the person in ORMB.</li> </ul> |
| <b>Algorithm Entity</b>     | Business Object Status - Enter   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.inboundCustomer.CreateEntityAlgo   |
| <b>Parameters</b>           | <p><b>Name:</b> Account Id Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Account Id Type</p> <p><b>Name:</b> Person Id Type<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Person Id Type</p>  |

## 5.145 Result Type Pre-processing Algorithm Type C1-RSTPRE

Table 5–242 Result Type Pre-processing Algorithm Type. C1-RSTPRE

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Result Type Pre-processing Algorithm Type                                     |
| <b>Detailed Description</b> | Result Type Pre-processing Algorithm Type                                     |
| <b>Algorithm Entity</b>     | Result Type – Pre Processing Algorithm  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePreProcAlgo |
| <b>Parameters</b>           | NA  |

## 5.146 Result Type Post-processing Algorithm Type C1-RSTPOST

Table 5–243 Result Type Post-processing Algorithm Type. C1-RSTPOST

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Result Type Post-processing Algorithm Type   |
| <b>Detailed Description</b> | Result Type Post-processing Algorithm Type   |
| <b>Algorithm Entity</b>     | Result Type – Pre Processing Algorithm   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcAlgo   |
| <b>Parameters</b>           | <p><b>Name:</b> contactType<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Contact Type</p> <p><b>Name:</b> contactClass<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Contact Class</p> <p><b>Name:</b> contactMethod<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Contact Method</p> |

## 5.147 Characteristic Type :Validate Date Field (Custom) C1-ADHDATE

Table 5–244 Characteristic Type :Validate Date Field (Custom). C1-ADHDATE

|                    |   |
|--------------------|---|
| <b>Description</b> | Characteristic Type :Validate Date Field (Custom) |
|--------------------|---|

|                             |   |
|-----------------------------|---|
| <b>Detailed Description</b> | <p>Custom Date validation</p> <p>This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time. The Parameters From Date and To Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is specified, they must be in the format YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field.</p> <p>The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS.</p> <p>Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system.</p> |
| <b>Algorithm Entity</b>     | Characteristic Type – Adhoc Validation  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.CustomAdhocDateValidationAlgComp  |
| <b>Parameters</b>           | NA  |

## 5.148 Algorithm Type for Dialer Results Upload C1-DLRRSUPLD

Table 5–245 Algorithm Type for Dialer Results Upload. C1-DLRRSUPLD

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Algorithm Type for Dialer Results Upload                                  |
| <b>Detailed Description</b> | Algorithm Type for Dialer Results Upload                                  |
| <b>Algorithm Entity</b>     | Dialer Result Upload – Account, Customer, CaseType                        |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.dialerResultUpload.DialerResultUploadAlgo |
| <b>Parameters</b>           | NA  |

## 5.149 Algorithm for Hardship case creation activity C1-CRTHDSP

Table 5–246 Algorithm for Hardship case creation activity. C1-CRTHDSP

|                    |   |
|--------------------|---|
| <b>Description</b> | Algorithm for Hardship case creation activity   |
| <b>Detailed</b>    | This Algorithm is responsible for making a Hardship Case entry on the Party, when the |

|                         |  |
|-------------------------|--|
| <b>Description</b>      | Hardship case is created.  |
| <b>Algorithm Entity</b> | Case Type – Enter Status   |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCaseCreationActivity |
| <b>Parameters</b>       | NA   |

## 5.150 This Algorithm is used to abort Approval work item C1-ABORTAPP

*Table 5–247 This Algorithm is used to abort Approval work item. C1-ABORTAPP*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | This Algorithm is used to abort Approval work item   |
| <b>Detailed Description</b> | <p>This algorithm is used to abort approval work item. Input to the algorithm is composite name, instance title and case status exclusion list. If next case status is present in case status exclusion list then work item instance is not aborted.</p> <ul style="list-style-type: none"> <li>■ caseStatusExclusionList: Comma separated list of case status for which approval work item shouldn't be aborted.</li> <li>■ Composite Name: Fully qualified approval class name.</li> <li>■ Instance Title: Approval instance work item title prefix. <ul style="list-style-type: none"> <li>• Example: Input parameters and it's applicable value for ROSO Process,</li> </ul> </li> <li>■ Composite Name: com.ofss.fc.workflow.process.ROSOProcessForApproval</li> <li>■ Instance Title: ROSO_CASE_Value of the above parameters is dependent upon the SOA approval work flow.</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.AbortApprovalWorkItemsAlgo   |
| <b>Parameters</b>           | <p><b>Name:</b> Composite Name<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Composite Name</p> <p><b>Name:</b> Instance Title<br/> <b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Instance Title</p>  |

|  |   |
|--|---|
|  | <b>Name:</b> Case Status Exclusion List<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Case Status Exclusion List |
|--|---|

## 5.151 Cancel Process Approval Request:Financial Hardship C1-CANFHAPPR

Table 5–248 Cancel Process Approval Request:Financial Hardship. C1-CANFHAPPR

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Cancel Process Approval Request:Financial Hardship   |
| <b>Detailed Description</b> | <p>This algorithm will cancel all pending approval requests for the case.<br/> Example for parameter values for hardship Process:</p> <ul style="list-style-type: none"> <li>■ Composite Name:<br/>com.ofss.fc.workflow.process.FinancialHardshipProcessForApproval</li> <li>■ Instance Title: FINANCIAL_HARDSHIP_CASE_</li> </ul> <p>Value of the above parameters depends upon the SOA approval work flow.</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.financialHardship.CancelFinancialHardshipApprovalReqAlgo   |
| <b>Parameters</b>           | <b>Name:</b> Composite Name<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Composite Name<br><br><b>Name:</b> Instance Title<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Instance Title   |

## 5.152 Hardship Characteristic Association C1-FHCHARASC

Table 5–249 Hardship Characteristic Association. C1-FHCHARASC

|                             |                                     |
|-----------------------------|-------------------------------------|
| <b>Description</b>          | Hardship Characteristic Association |
| <b>Detailed Description</b> | Hardship Characteristic Association |

|                         |   |
|-------------------------|---|
| <b>Algorithm Entity</b> | Case Type – Enter Status  |
| <b>Program Type</b>     | Java  |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCharAssociation |
| <b>Parameters</b>       | NA  |

## 5.153 Pre-Populated system facts to be used for Rule C1-PPSF

*Table 5–250 Pre-Populated system facts to be used for Rule. C1-PPSF*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Pre-Populated system facts to be used for Rule   |
| <b>Detailed Description</b> | This algorithm is used to populate input system fact for Rule. It used as an input to RuleFactPopulation algorithm. System Facts populated through this algorithm are SystemDate and PostingDate.<br>This is sample implementation to provide populated facts to RuleFactPopulation algorithm. User can create his own algorithm type based on his requirement (Algorithm Entity must be Rule Execution - Pre Populated Rule Facts). |
| <b>Algorithm Entity</b>     | Rule Execution – Pre Populated Rule Facts  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.PrePopulatedSystemFacts  |
| <b>Parameters</b>           | NA   |

## 5.154 Assign Batch level TODOs(task) to a queue C1-ASGNTASK

*Table 5–251 Assign Batch level TODOs(task) to a queue. C1-ASGNTASK*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Assign Batch level TODOs(task) to a queue.                                 |
| <b>Detailed Description</b> | Assign Batch level TODOs(task) to a queue.                                 |
| <b>Algorithm Entity</b>     | To Do Type – To Do Post Processing   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.batch.algorithm.AssignTaskToQueueAlgorithm |
| <b>Parameters</b>           | NA   |

## 5.155 Validate Date Field :Custom C1-ADHV-DTD

*Table 5–252 Validate Date Field :Custom. C1-ADHV-DTD*

|                  |                             |
|------------------|-----------------------------|
| <b>Descripti</b> | Validate Date Field :Custom |
|------------------|-----------------------------|

|                             |   |
|-----------------------------|---|
| on                          |   |
| <b>Detailed Description</b> | <p>This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time. The Parameters From Date and To Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is specified, they must be in the format YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field.</p> <p>The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS.</p> <p>Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system.</p> |
| <b>Algorithm Entity</b>     | Characteristic Type – Adhoc Validation  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.ProductAdhocDateValidationAlgComp   |
| <b>Parameters</b>           | <p><b>Name:</b> From Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> From Date</p> <p><b>Name:</b> To Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> To Date</p> <p><b>Name:</b> Date Format1 (Stored Format)<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Date Format1 (Stored Format)</p> <p><b>Name:</b> Date Format2<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Date Format2</p> <p><b>Name:</b> Date Format3<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Date Format3</p> <p><b>Name:</b> Date Format4<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Date Format4</p>  |



|  |   |
|--|---|
|  | <p><b>Name:</b> Date Format5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format5</p> <p><b>Name:</b> Date Format6<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format6</p> |
|--|---|

## 5.156 Characteristic Date field Validation C1-CHARDTVAl

Table 5–253 Characteristic Date field Validation. C1-CHARDTVAl

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Characteristic Date field Validation  |
| <b>Detailed Description</b> | <p>This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time.</p> <p>The Parameters From Date and To Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is specified, they must be in the format YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field.</p> <p>The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS.</p> <p>Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system.</p> <p>Parameter 9: valid values are true/false. When Business date validation required is true, algorithm will validate the given date to check if its a valid business date.</p> |
| <b>Algorithm Entity</b>     | Characteristic Type – Adhoc Validation  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.CharAdhocDateValidation  |
| <b>Parameters</b>           | <p><b>Name:</b> From Date<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> From Date</p> <p><b>Name:</b> To Date<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> To Date</p> <p><b>Name:</b> Date Format1 (Stored Format)</p>  |

|  |  |
|--|--|
|  | <p><b>Required (Yes/No):</b> Yes<br/> <b>Description:</b> Date Format1 (Stored Format)</p> <p><b>Name:</b> Date Format2<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format2</p> <p><b>Name:</b> Date Format3<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format3</p> <p><b>Name:</b> Date Format4<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format4</p> <p><b>Name:</b> Date Format5<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format5</p> <p><b>Name:</b> Date Format6<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Date Format6</p> <p><b>Name:</b> Business Date Validation Required<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Business Date Validation Required</p> |
|--|--|

## 5.157 Retry Case in Error C1-RCASEE

Table 5–254 Retry Case in Error C1-RCASEE

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Retry Case in Error   |
| <b>Detailed Description</b> | This algorithm is plugged-in on auto-transition of error states and attempts to retry validation, completion or wait if the To Do Entry associated is not being worked on. The retry will be performed only until the input Maximum Number of Retries is reached. |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.RetryCaseInErrorForHardshipApp  |
| <b>Parameters</b>           | <b>Name:</b> Retry Case Status Code<br><b>Required (Yes/No):</b> No   |

|  |   |
|--|---|
|  | <b>Description:</b> Retry Case Status Code<br><br><b>Name:</b> Max Retries<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Max Retries |
|--|---|

## 5.158 Allocate Queue for Customer Level Case C1-ALLOCQUE

*Table 5–255 Allocate Queue for Customer Level Case. C1-ALLOCQUE*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Allocate Queue for Customer Level Case  |
| <b>Detailed Description</b> | Allocate Queue for Customer Level Case. Only Queue Allocation would be done. User Allocation would be skipped for customer level cases. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.PerformQueueAllocation                                |
| <b>Parameters</b>           | <b>Name:</b> Queue Code<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Queue Code  |

## 5.159 Person Address – Collection C1-PERADDRRC

*Table 5–256 Person Address – Collection. C1-PERADDRRC*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Person Address – Collection  |
| <b>Detailed Description</b> | This Algorithm is a reference implementation for consulting. This algorithm will be used for validating Person address as per requirement. |
| <b>Algorithm Entity</b>     | Business Object – Validation   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.address.PersonCollectionAddressValidation  |
| <b>Parameters</b>           | NA   |

## 5.160 Person Contact Point Update - Post Processing C1-PERCONTPP

*Table 5–257 Person Contact Point Update - Post Processing. C1-PERCONTPP*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Person Contact Point Update - Post Processing  |
| <b>Detailed Description</b> | This is a reference implementation of Post processing Algo. Customization team can utilize this hook |
| <b>Algorithm Entity</b>     | Collection Contact Preference – Post Process   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot                     |
| <b>Parameters</b>           | NA   |

## 5.161 Update Self Serve Flag Algorithm C1-SELFERVE

*Table 5–258 Update Self Serve Flag Algorithm. C1-SELFERVE*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Update Self Serve Flag Algorithm   |
| <b>Detailed Description</b> | Action -soft parameter mentioned in algorithm type which will update the self_ serve flag to Y or N.<br>If Action = Set make Self Serve Flag = Y<br>If Action = Reset make Self Serve Flag = N |
| <b>Algorithm Entity</b>     | Case Type – Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.UpdateSelfServeFlag   |
| <b>Parameters</b>           | <b>Name:</b> Action<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Action   |

## 5.162 Create Task for Self Serve Request for Assistance transaction C1-FLWRTSK

*Table 5–259 Create Task for Self Serve Request for Assistance transaction. C1-FLWRTSK*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Create Task for Self Serve Request for Assistance transaction                     |
| <b>Detailed Description</b> | This algorithm will be used to Create Task post Follow Up.                        |
| <b>Algorithm Entity</b>     | Result Type – Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.actionObject.actionHistory.FollowUpResultTaskAlgo |

|                   |  |
|-------------------|--|
| <b>Parameters</b> | <b>Name:</b> Task For<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Task For     |
|                   | <b>Name:</b> Task Type<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Task Type   |
|                   | <b>Name:</b> Task Queue<br><b>Required (Yes/No):</b> Yes<br><b>Description:</b> Task Queue |

## 5.163 Transition to Under Settlement Offer Status

Table 5–260 Transition to Under Settlement Offer Status. C1-TSOF

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Transition to Under Settlement Offer Status if Active Settlement Offer Flag is Yes   |
| <b>Detailed Description</b> | Transition the case to Under Settlement Offer Status if Active Settlement Offer Flag is Yes.<br>Parameters:<br><ol style="list-style-type: none"> <li>Under Settlement Offer Status: Value should be "Under Settlement Offer Status Code". Case will be transitioned to this Status.</li> <li>Reallocate Switch : Value could be "Y" on "N"</li> </ol> |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.TransitionToUnderSettlement Offer   |
| <b>Parameters</b>           | <b>Name:</b> Under Settlement Offer Status<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Under Settlement Offer Status<br><br><b>Name:</b> Reallocate Switch<br><b>Required (Yes/No):</b> No<br><b>Description:</b> Reallocate Switch   |

## 5.164 Create PTP Validation Algorithm

Table 5–261 Create PTP Validation Algorithm. C1-VPTP

|                    |                                 |
|--------------------|---------------------------------|
| <b>Description</b> | Create PTP Validation Algorithm |
|--------------------|---------------------------------|

|                             |  |
|-----------------------------|--|
| <b>Detailed Description</b> | This algorithm performs below validations before creating PTP:<br>If Active Settlement Offer Flag is Yes and last PTP instalment date <= Acceptance Date + Settlement term and sum of PTP installment amount = Settlement amount, it will continue process to create PTP, otherwise Validation error will be thrown. |
| <b>Algorithm Entity</b>     | Case Type – Enter Status Validation  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.ValidatePTPForSettlementOffer   |
| <b>Parameters</b>           | No Parameters  |

## 5.165 Under Settlement Offer Status Monitor Algorithm for Settlement Expiry/Offer Acceptance/Rejection

*Table 5–262 Under Settlement Offer Status Monitor Algorithm for Settlement Expiry/Offer Acceptance/Rejection. C1-MONUS*

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Under Settlement Offer Status Monitor Algorithm for Settlement Expiry/Offer Acceptance/Rejection  |
| <b>Detailed Description</b> | <p>If Active Settlement Flag is 'Y' And if Settlement offer status is 'Accepted By Customer', then</p> <p>If Task Type provided in the Parameter task does not exist on Account, then it creates Account Level task and assign to queue.</p> <p>Else</p> <p>If settlement offer status is 'Valid' and Validation Date &gt; Offer Expiry Date, then this Algorithm will mark Settlement offer status as 'Expired' and set Active Settlement Offer Flag as 'N' and move case status to 'Contact/Contact Alternate' based on contact alternate flag and Set Re-Allocation Switch = Y for the case.</p> <p>If settlement offer status is 'Rejected By Customer', then this Algorithm will move case status to 'Contact/Contact Alternate' based on contact alternate flag and Set Re-Allocation Switch as 'Y' for the case in the context.</p> <p>Possible valid values:</p> <p>Reallocate Switch - Y or N</p> <p>Validation Date - POSTING_DATE or SYSTEM_DATE</p> <p>All other parameter values will be their respective codes.</p> |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.MonitorSettlementOffer   |
| <b>Parameters</b>           | <p><b>Name:</b> Task Type</p> <p><b>Required (Yes/No):</b> No</p> <p><b>Description:</b> Task Type</p><br><p><b>Name:</b> Task Queue</p>  |

|  |   |
|--|---|
|  | <p><b>Required (Yes/No):</b> No<br/> <b>Description:</b> Task Queue</p> <p><b>Name:</b> Contact Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Status</p> <p><b>Name:</b> Contact Alt Status<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Contact Alt Status</p> <p><b>Name:</b> Reallocate Switch<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Reallocate Switch</p> <p><b>Name:</b> Validation Date<br/> <b>Required (Yes/No):</b> No<br/> <b>Description:</b> Validation Date</p> |
|--|---|

## 5.166 Algorithm to identify Partial Charge off Accounts

Table 5–263 Algorithm to identify Partial Charge off Accounts BATCHJOBPARM

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Algorithm to identify Partial Charge off Accounts  |
| <b>Detailed Description</b> | <p>Algorithm to identify eligible Partial Charge off Accounts to be processed by batch.</p> <p>Soft Parameters:</p> <ol style="list-style-type: none"> <li>1. Host - Host Code</li> <li>2. Collateral Type - Collateral Type Codes to be Considered for Assessment</li> <li>3. Product Group Code - Account Product Group Codes to be Considered for Assessment</li> <li>4. Bankruptcy Risk Indicator Code - Bankruptcy Risk Indicator Code to be Considered for Party</li> <li>5. Deceased Risk Indicator Code - Deceased Risk Indicator Code to be Considered for Party</li> </ol> |
| <b>Algorithm Entity</b>     | Generic Batch Job Parameters Algorithm Spot  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.partialChargeOffBatchJobParamsNGP  |
| <b>Parameters</b>           | <b>Name:</b> Host  |

|  |   |
|--|---|
|  | <p><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Host</p> <p><b>Name:</b> Collateral Type<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Collateral Type</p> <p><b>Name:</b> Product Group Code<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Product Group Code</p> <p><b>Name:</b> Bankruptcy Risk Indicator Code<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Bankruptcy Risk Indicator Code</p> <p><b>Name:</b> Deceased Risk Indicator Code<br/><b>Required (Yes/No):</b> Yes<br/><b>Description:</b> Deceased Risk Indicator Code</p> |
|--|---|

## 5.167 Monitor Settlement Offer Payments

Table 5–264 Monitor Settlement Offer Payments C1-MSOP

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Monitor Settlement Offer Payments   |
| <b>Detailed Description</b> | <p>If Active Settlement Offer Flag = Y AND Settlement Offer Payment Total &lt; Settlement Amount, then check:</p> <p>If System date &gt; Acceptance Date + Settlement Term, then change Offer Status = Settlement Broken</p> <p>Possible valid values of Validation Date are POSTING DATE or SYSTEM DATE.</p> |
| <b>Algorithm Entity</b>     | Case Type – Auto Transition   |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.recovery.externalLifecycle.algorithms.MonitorSettlementOfferPaymentsAlgorithm  |
| <b>Parameters</b>           | <p><b>Name:</b> Validation Date<br/><b>Required (Yes/No):</b> No<br/><b>Description:</b> Validation Date</p>  |





# 6 Localized Algorithms

## 6.1 Localized Algorithms

*Table 6–1 Case Transition for Active Service Member C1-ACTMEMCHK*

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Case Transition for Active Service Member  |
| <b>Detailed Description</b> | <p>This algorithm will transit the case to Suspend Status if the customer is in Active Service or dependent of a person in Active Service.</p> <p>Validate against all Financial Owners parameter will decide if check has to be done for main customer or all financial owners. If Validate against all Financial Owners parameter value is Y, algorithm will check active service member against all financial owners.</p>   |
| <b>Algorithm Entity</b>     | Case Type - Auto Transition  |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.scra.algorithm.ActiveServiceAlgorithm  |
| <b>Parameters</b>           | <p>Name: Suspend Status<br/>Required (Yes/No): No<br/>Description: Suspend Status</p> <p>Name: All Financial Owner Validation<br/>Required (Yes/No): Yes<br/>Description: All Financial Owner Validation</p> <p>Name: Validation Date<br/>Required (Yes/No): Yes<br/>Description: Validation Date</p> <p>Name: Dependent Validation<br/>Required (Yes/No): Yes<br/>Description: Dependent Validation</p> <p>Name: Suspend Reason Characteristics<br/>Required (Yes/No): No<br/>Description: Suspend Reason Characteristics</p> |
| <b>Detailed Design</b>      | This algorithm will transit the case to Suspend Status if the customer is in Active Service or dependent of a person in Active Service.  |

**Table 6–2 Active Military Check on Associated Customers - Enter Validation C1-BLOCKREPO**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Block Repossession - Enter Status   |
| <b>Detailed Description</b> | Verify if repossession needs to be blocked as per SCRA regulations  |
| <b>Algorithm Entity</b>     | Case Status - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ActiveMilitaryServiceCheckonAssociatedCustomers  |
| <b>Parameters</b>           | Name: Validation Date<br>Required (Yes/No): Yes<br>Description: Validation Date<br><br>Name: Repossession Block Period<br>Required (Yes/No): No<br>Description: Repossession Block Period |
| <b>Detailed Design</b>      | Verify if repossession needs to be blocked as per SCRA regulations  |

**Table 6–3 Metro 2 Reporting - Account Status Code post Liquidation C1- ASCLIQU**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Metro 2 Reporting - Account Status Code post Liquidation   |
| <b>Detailed Description</b> | <p>If Repossession Reason = Voluntary Surrender<br/>If Account Status Condition = Consumer not responsible for Remaining Balance/ No Deficiency Balance<br/>Set Account Status Code = 95;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance'<br/>Set Account Status Code = 95;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full<br/>Set Account Status Code = 61;<br/>Else</p> <p>If Account Status Condition = Consumer not responsible for Remaining Balance/ No Deficiency Balance</p> |

|                         |  |
|-------------------------|--|
|                         | <p>Set Account Status Code = 96;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance'<br/>Set Account Status Code = 96;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full<br/>Set Account Status Code = 63;</p> <p>Additionally record the Repossession Date and the Last Payment Date in each of the scenarios.<br/>Data to be logged:<br/>(Current Date, Account Number, Account Status Code, Repossession Date, Last Payment Date)</p> <p>Account status Code Char value should be C1-ASCOD. It should be product shipped.<br/>Char Values are: CNRBND,CRBAP,CRRB</p> |
| <b>Algorithm Entity</b> | Result Type – Post Processing Algorithm  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithm s. Metro2AcctStatusCodePostLiquidationPostProcessing   |
| <b>Parameters</b>       | <p>Name: Account status Code Char<br/>Required (Yes/No): Yes<br/>Description: Account status Code Char</p> <p>Name: Voluntary Surrender Code<br/>Required (Yes/No): Yes<br/>Description: Voluntary Surrender Code</p>  |
| <b>Detailed Design</b>  | Metro 2 Reporting - Account Status Code post Liquidation   |

**Table 6–4 Metro 2 Reporting - Account Status Code C1- ASCREPO**

|                                       |  |
|---------------------------------------|--|
| <b>Descri<br/>ption</b>               | Metro 2 Reporting - Account Status Code  |
| <b>Detail<br/>ed Descri<br/>ption</b> | <p>If Repossession Reason = Voluntary Surrender<br/>Set Account Status Code = Account Status Code for Voluntary Surrender<br/>Special Comment Code = Special Comment Code for Voluntary Surrender<br/>Else<br/>Set Account Status Code = Account Status Code for Normal Repossession<br/>Special Comment Code = Special Comment Code for Normal Repossession<br/>(If multiple accounts associated with the case, the Account Status Code should be set for all</p> |

|                         |  |
|-------------------------|--|
|                         | associated accounts)   |
| <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.algorithms.Metro2AcctStatusCodeEnterProcessingAlgo   |
| <b>Parameters</b>       | <p>Name: Special Comment Code for Normal Repossession<br/>Required (Yes/No): No<br/>Description: Special Comment Code for Normal Repossession</p> <p>Name: Special Comment Code for Voluntary Surrender<br/>Required (Yes/No): No<br/>Description: Special Comment Code for Voluntary Surrender</p> <p>Name: Voluntary Surrender Code<br/>Required (Yes/No): Yes<br/>Description: Voluntary Surrender Code</p> <p>Name: Account Status Code for Normal Repossession<br/>Required (Yes/No): Yes<br/>Description: Account Status Code for Normal Repossession</p> <p>Name: Account Status Code for Voluntary Surrender<br/>Required (Yes/No): Yes<br/>Description: Account Status Code for Voluntary Surrender</p> |
| <b>Detailed Design</b>  | Metro 2 Reporting - Account Status Code  |

**Table 6–5 Metro 2 Reporting - Compliance condition code C1- COMCODE**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Metro 2 Reporting - Compliance condition code   |
| <b>Detailed Description</b> | Set the Compliance Condition Code sent to Credit Bureau with the value selected in the characteristic given in the parameter. |
| <b>Algorithm Entity</b>     | Result Type - Post Processing Algorithm   |
| <b>Program Type</b>         | Java  |

|                        |   |
|------------------------|---|
| <b>Program Name</b>    | com.splwg.ccb.domain.collection.caseType.earlyCollections.Metro2ComplianceCodePostProcessingAlgo                              |
| <b>Parameters</b>      | Name: Compliance Condition Code Char<br>Required (Yes/No): Yes<br>Description: Compliance Condition Code Char                 |
| <b>Detailed Design</b> | Set the Compliance Condition Code sent to Credit Bureau with the value selected in the characteristic given in the parameter. |

**Table 6–6 Metro 2 Reporting - Marking Account as Close C1- CFOSEP**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Metro 2 Reporting - Marking Account as Close   |
| <b>Detailed Description</b> | The logic is incorporated for Metro Algorithm only if an Account is close then it should be marked as Close        |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2CheckForOpenStatusEnterProcessing |
| <b>Parameters</b>           | NA   |
| <b>Detailed Design</b>      | The logic is incorporated for Metro Algorithm only if an Account is close than it should be marked as Close        |

**Table 6–7 Metro 2 Reporting - Consumer Information Indicator C1- CONINFOIN**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Metro 2 Reporting - Consumer Information Indicator  |
| <b>Detailed Description</b> | Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.          |
| <b>Algorithm Entity</b>     | Case Type - Enter Status  |
| <b>Program Type</b>         | Java  |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInformationIndicator |
| <b>Parameters</b>           | Name: Chapter7 CII Code<br>Required (Yes/No): Yes   |

|                        |   |
|------------------------|---|
|                        | <p>Description: Chapter7 CII Code</p> <p>Name: Chapter11 CII Code<br/>Required (Yes/No): Yes<br/>Description: Chapter11 CII Code</p> <p>Name: Chapter12 CII Code<br/>Required (Yes/No): Yes<br/>Description: Chapter12 CII Code</p> <p>Name: Chapter13 CII Code<br/>Required (Yes/No): Yes<br/>Description: Chapter13 CII Code</p> <p>Name: Other CII Code<br/>Required (Yes/No): Yes<br/>Description: Other CII Code</p> |
| <b>Detailed Design</b> | Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.  |

**Table 6–8 Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post Discharge C1- CIIPSTDIS**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post Discharge   |
| <b>Detailed Description</b> | <p>If any associated secured account without confirmed plan on it report CII as per No Confirmed Plan CII Code parameter.</p> <p>Else</p> <p>Report CII = &lt;Chapter12 CII Code&gt; for Chapter 12</p> <p>Report CII = &lt;Chapter13 CII Code&gt;for Chapter 13</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.bankruptcy.Metro2ConsumerInfoIndiChap13PostDis   |
| <b>Parameters</b>           | <p>Name:No Confirmed Plan CII Code<br/>Required (Yes/No): Yes<br/>Description: No Confirmed Plan CII Code</p> <p>Name: Chapter12 CII Code<br/>Required (Yes/No): Yes<br/>Description: Chapter12 CII Code</p>   |

|                        |   |
|------------------------|---|
|                        | Name: Chapter13 CII Code<br>Required (Yes/No): Yes<br>Description: Chapter13 CII Code   |
| <b>Detailed Design</b> | If any associated secured account without confirmed plan on it report CII as per No Confirmed Plan CII Code parameter.<br>Else<br>Report CII = <Chapter12 CII Code> for Chapter 12<br>Report CII = <Chapter13 CII Code>for Chapter 13 |

**Table 6–9 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1-CGCLC**

|                             |  |
|-----------------------------|--|
| <b>Description</b>          | Credit Grantor Cannot Locate Consumer  |
| <b>Detailed Description</b> | Automatically Set in Skip Tracing Status - Enter Processing.<br>Set for all borrowers on the account.<br>Parameter<br>CII CODE - Mandatory |
| <b>Algorithm Entity</b>     | Case Type - Enter Status   |
| <b>Program Type</b>         | Java   |
| <b>Program Name</b>         | com.splwg.ccb.domain.collection.caseType.earlyCollections.CreditGrantorCannotLocateConsumer  |
| <b>Parameters</b>           | Name: Cii Code<br>Required (Yes/No): Yes<br>Description: Cii Code  |
| <b>Detailed Design</b>      | Credit Grantor Cannot Locate Consumer  |

**Table 6–10 Metro 2 Reporting - Consumer Now Located (Removes previously reported T Indicator) C1-CNLREM**

|                             |   |
|-----------------------------|---|
| <b>Description</b>          | Consumer Now Located (Removes previously reported T Indicator)  |
| <b>Detailed Description</b> | Skip Tracing Actions<br>New Result: Metro 2: Consumer Information Indicator<br>Chars:<br>Party Id (Adhoc)<br>CII (Predefined Values: (T,U)<br>Post Processing:<br>Set the given CII Code for the party id provided. |



|                         |  |
|-------------------------|--|
| <b>Algorithm Entity</b> | Result Type - Post Processing Algorithm  |
| <b>Program Type</b>     | Java   |
| <b>Program Name</b>     | com.splwg.ccb.domain.collection.caseType.earlyCollections.ConsumerNowLocated   |
| <b>Parameters</b>       | Name: Party Id Char<br>Required (Yes/No): Yes<br>Description: Party Id Char<br><br>Name: Cii Char<br>Required (Yes/No): Yes<br>Description: Cii Char |
| <b>Detailed Design</b>  | Consumer Now Located (Removes previously reported T Indicator)   |

**Table 6–11 Metro 2 Reporting - Set DPD and Outstanding amount to all associated accounts C1-SETDPD**

|                                  |   |
|----------------------------------|---|
| <b>Descripti<br/>on</b>          | Set DPD and Outstanding amount to all associated accounts   |
| <b>Detailed Descripti<br/>on</b> | Record the DPD and the Outstanding Balance at account level if number of cases associated with the account of given case type < 2 |
| <b>Algorith<br/>m Entity</b>     | Case Type - Enter Status  |
| <b>Program<br/>Type</b>          | Java  |
| <b>Program<br/>Name</b>          | com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.SetDPDOut<br>standingAmount                            |
| <b>Paramete<br/>rs</b>           | Name: Bankruptcy Case Type<br>Required (Yes/No): No<br>Description: Bankruptcy Case Type  |
| <b>Detailed<br/>Design</b>       | Set DPD and Outstanding amount to all associated accounts   |

# 7 Feeder Services

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

**Table 7–1 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, hrshipAppld |
| AccountArrearFeederApplicationService       | AccountArrearFeederResponse update(SessionContext sessionContext, AccountArrearFeeder   | This   | hostAcctNumber, srcHostId, referenceVal                                      |

| Service Name                                     | Method Name   | Description  | Mandatory Fields                     |
|--|---|--|--------------------------------------|
|  | <p>WrapperDTO<br/>accountArrearFeederWrapperDTO)<br/>throws FatalException;</p>   | <p>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.</p> |                                      |
| <p>AccountWarningIndFeederApplicationService</p> | <p>AccountWarningIndFeederResponse<br/>update(SessionContext<br/>sessionContext,AccountWarningIndFeederWrapperDTO<br/>accountWarningIndFeederWrapperDTO)<br/>throws FatalException;</p> | <p>This service adds or updates account warning indicator or related fields in the feeder table. It handles add, update and delete operations.</p>   | <p>hostAcctNumber,<br/>srcHostId</p> |

| Service Name                    | Method Name  | Description  | Mandatory Fields                                     |
|---------------------------------|--|--|--|
| 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           |
| FeederPersonApplicationService  | FeederPersonResponse update (SessionContext sessionContext, AccountFeederWrapperDTO 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, FeederPerAddrWrapperDTO) throws FatalException                          | This service adds or updates party addresses related fields in the feeder table. It handles add, update and                        | srcHostId, hostCustomerNbr, fdrAddrSeqId, addrTypeCd |

| Service Name                           | Method Name  | Description  | Mandatory Fields   |
|--|--|--|--|
|  |  | delete operations.   |  |
| FeederPerEmpProfileApplication Service | 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, hostCustomerNumber, determinantValue, fdrEmpSeqId     |
| FeederContactPrefApplication Service   | 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, hostCustomerNumber, 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   | srcHostId, hostCustomerNumber, idType                            |

| Service Name                                  | Method Name  | Description  | Mandatory Fields   |
|---|--|--|--|
|   |  | license and so on in the feeder table. It handles add, update and delete operations.   |  |
| AccountFeederUpdateForBatchApplicationService | AccountFeederResponse update (SessionContext sessionContext, AccountFeederWrapperDTO accountFeederWrapperDTO) throws FatalException        | This service is used for OBP EOD/BOD batch shells. This service adds or updates account related fields in the feeder table. It handles add, update and delete operations | hostAcctNumber, srcHostId                                  |
| ScraHistFeederApplicationService              | ScraHistFeederResponse update (SessionContext p_SessionContext, ScraHistFeederWrapperDTO p_ScraHistFeederWrapperDTO) throws FatalException | This service is used for OBP EOD/BOD batch shells. This service adds or  | hostCustomerNumber, determinantValue, svcOrdNum, srcHostId |

| Service Name                                 | Method Name   | Description   | Mandatory Fields                   |
|--|---|---|------------------------------------|
|  |   | updates customer related fields in the feeder table. It handles add, update and delete operations.  |                                    |
| MinimumAmountDueFeederApplicationService     | MinimumAmountDueFeederResponse update(SessionContext p_SessionContext, MinimumAmountDueFeederWrapperDTO p_MinimumAmountDueFeederWrapperDTO) throws FatalException             | This service is used for OBP EOD/BOD batch shells. This service adds or updates account related fields in the feeder table. It handles add, update and delete operations. | hostAcctNumber, srcHostId, dueDate |
| CollateralAutomobileFeederApplicationService | CollateralAutomobileFeederResponse update(SessionContext p_SessionContext, CollateralAutomobileFeederWrapperDTO p_CollateralAutomobileFeederWrapperDTO) throws FatalException |   | srcHostId, collateralCd            |
| PaymentTrackerDetailsApplicationService      | PaymentTrackerDetailsResponse update(SessionContext p_SessionContext,   | This  | hostAcctNumber, srcHostId, dueDate |

| Service Name | Method Name   | Description   | Mandatory Fields |
|--------------|---|---|------------------|
|              | PaymentTrackerDetailsWrapperDTO<br>p_ PaymentTrackerDetailsWrapperDTO)<br>throws FatalException | service is used for OBP EOD/BOD batch shells. This service adds or updates payment related fields in the feeder table. It handles add, update and delete operations |                  |





# 8 Dialer Webservice Integration

Dialer web service can be consumed by consultants to notify collector about the outbound call to customer by vendor.

## 8.1 Generic Data Type

This section provides details of the generic data type.

**Table 8–1 Generic Data Type**

| Data Type                 | Format   |
|---------------------------|--|
| A (CHARACTERS A-Z)        | UPPERCASE Characters, LEFT JUSTIFIED, PADDING with BLANK spaces.   |
| AN (ALPHANUMERIC A-Z,0-9) | Alphanumeric fields those are mainly used as RECORD IDENTIFIER must be in UPPERCASE and LEFT JUSTIFIED and PADDING characters are BLANK SPACES.  |
| DT (DATE)                 | <ul style="list-style-type: none"> <li>■ Format: YYYYMMDD.</li> <li>■ For example, 10-DEC-1982 is represented as 10121982</li> <li>■ If a date field is optional and the date is not known or the date field is optional and the date is not applicable, the field should be zero filled.</li> </ul>   |
| N (Numeric 0-9)           | Number field: Numeric values must be right justified and zero filled.  |
| Amount field              | <ul style="list-style-type: none"> <li>■ Values must be right justified and zero filled.</li> <li>■ Must not contain alpha characters, dollar signs, commas, plus and minus signs, decimal point or spaces.</li> </ul>   |
| M - Mandatory             | <p>A valid value must be reported. For a single character mandatory field, blank (space) is not a valid value. A mandatory:</p> <ul style="list-style-type: none"> <li>■ alpha field, must not start with a space or be space filled</li> <li>■ alphanumeric field, must not start with a space or be space filled</li> <li>■ numeric field, must not start with a space or be space filled and must not be zero filled, except where specified</li> <li>■ date field, must not be zero filled.</li> </ul> |
| O - Optional              | <p>A valid value must be reported where specified conditions are met. In this case the field becomes mandatory. See rules above.</p> <p>If the data is not available, then:</p> <ul style="list-style-type: none"> <li>■ alpha field, must be space filled</li> <li>■ alphanumeric field, must be space filled</li> </ul>  |

| Data Type | Format  |
|-----------|---|
|           | <ul style="list-style-type: none"> <li>■ numeric, must be zero filled</li> <li>■ date field, must be zero filled</li> </ul> |

## 8.2 Summary

This section provides the detail summary for dialer webservice.

**Table 8–2 Summary**

| Field Name       | Description                                   |
|------------------|---|
| Description      | Dialer Webservice                             |
| Service Status   | New   |
| Business Process | Notify collector on outbound call to customer |
| Owner            | OB Collections                                |
| Source System(s) | OB Collections                                |
| Target System(s) | OB Collections consultant                     |
| Service Layer    | Data Service                                  |
| Service Scope    | OB Collections                                |
| Service Domain   | OB Collections                                |

## 8.3 Interface

This section provides the details on the interface.

**Table 8–3 Interface**

| Field Name               | Description                  |
|--------------------------|------------------------------|
| Direction                | Outbound                     |
| Interaction Pattern      | Online                       |
| Protocol                 | < SOAP/HTTP >                |
| Webservice Name          | CollectionDialerWebService   |
| Method                   | notifyUser                   |
| Message Exchange Pattern | Synchronous Request Response |
| Filename Format          | NA                           |
| Security Pattern         | SAML assertion               |

| Field Name                | Description   |
|---------------------------|---|
| Transaction Pattern       | Atomic  |
| Error Pattern             | Handled in Synchronous response                     |
| Recovery Pattern          | None  |
| Business Data Element (s) | OB Collections Case Data                            |
| Request Data Object(s)    | Customer Information                                |
| Response Data Object (s)  | Successful  |
| Pre-Conditions            | Customer Information exists in OB Collections       |
| Post-Conditions           | Customer information is passed to requesting system |

## 8.4 Service Management

This section provides the details on service management.

**Table 8–4 Service Management**

| Field Name                 | Description                                    |
|----------------------------|--|
| Monitoring                 |  |
| Alerting                   |  |
| Availability               | 24*7   |
| Failover / Failback        |  |
| Backup / Restore           |  |
| Performance (Latency)      | 2 Seconds per request                          |
| Performance (Peak Volumes) |  |
| Performance (Data Volumes) |  |
| Performance (Concurrency)  | Expectation of maximum 20 concurrent requests. |
| Error Logging              | Required                                       |
| Auditing / Logging         | Required                                       |

## 8.5 Request Message Details

As a part of request, consultant will pass user ID of logged in user, account number, case ID, party ID and Transaction Branch, Target Unit, Accessible Target Units, Host String. These fields will also be sent as these

are required by OB Collections to perform Authentication and Authorization checks.

## 8.6 Header Record

Not Applicable

## 8.7 Detail Record

This section provides the information on detail record.

**Table 8–5 Detail Record**

| Sr. No | OBP Field Name  | Data Type | Length | Mandatory / Optional | Description  | DTO Mapping   |
|--------|-----------------|-----------|--------|----------------------|--|---|
| 1      | User ID         | AN        | 255    | Mandatory            | User ID of logged in user  | Username  |
| 2      | Account Number  | N         | 40     | Mandatory            | Unique identifier of account   | SessionContext.transactionBranch                                |
| 3      | Case id         | N         | 10     | Optional             | Unique identifier of case  | SessionContext.targetUnit                                       |
| 4      | Customer Number | N         | 40     | Optional             | Unique identifier of customer  | SessionContext.accessibleTargetUnits                            |
| 5      | Host String     | AN        | 120    | Mandatory            | Source Host String field provides the information about the host where the concerned account is stored | AccountCustomerProfileWrapperDTO.CollectionDTO.sourceHostString |

## 8.8 Translation Rules

Not Applicable

---

## 8.9 Response Message Details

As a response to the request, the proper success or failure success or failure response message will be sent. Service notify collector working on specified account/customer in the input about the outbound call made by vendor.

## 8.10 Customer Information

This section provides the details on customer information.

*Table 8–6 Customer information*

| Sr. No | OBP Field Name | Data Type | Length    | Mandatory / Optional       | Description                |
|--------|----------------|-----------|-----------|----------------------------|----------------------------|
| 1      | Message        | AN        | Mandatory | Success or failure message | Success or failure message |

## 8.11 Constraints

Not Applicable