

Oracle® Banking Platform Collections and Recovery

Interface Specification Guide

Release 2.12.0.0.0

F43567-01

June 2021

Oracle Banking Platform Collections and Recovery Interface Specification Guide, Release 2.12.0.0.0

F43567-01

Copyright © 2011, 2021, 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

Preface	27
Audience	27
Documentation Accessibility	27
Organization of the Guide	27
Related Documents	28
Conventions	28
1 About this Guide	30
2 Introduction	32
3 System Overview	34
4 Staging Area	36
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
5 Algorithms	100
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 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- 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- fer Acceptance/Rejection	270
5.166 Algorithm to identify Partial Charge off Accounts	271
5.167 Monitor Settlement Offer Payments	272
6 Localized Algorithms	274
6.1 Localized Algorithms	274
7 Feeder Services	282

8 Dialer Webservice Integration	290
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- 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- cessing: C1-CKSPLCASE	142
Table 5–77 Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullINDAcc	143
Table 5–78 Associate all accounts to the case where customer is a primary bor- rower- Enter Processing: C1-ASSCTEACC	144
Table 5–79 Exclude all the associated accounts from Dialer- Enter Processing: C1-ExcAccDir	144
Table 5–80 Initiate Collateral Valuation for all collaterals whose last valuation 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 monitor delinquency- Monitoring: C1-MTRCRGDQY	146
Table 5–82 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1- MTR341HRG	147
Table 5–83 Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring: C1-MTRPYMPLN	148
Table 5–84 Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD	148
Table 5–85 Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring: C1-MTRHRNGDT	149
Table 5–86 Determine in which status the case should proceed for Bankruptcy Treatment- Post Processing C1-DTMBKTRTM	150
Table 5–87 Validate if appropriate Case Details have been entered by the user- 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 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-REMCSCH ..	227
Table 5–190 Transition case on a date on a case characteristic. C1-TRANSDT ...	227
Table 5–191 Set Account Nxt Credit Review Date to current date. C1- 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- 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 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- BLOCKREPO	275
Table 6–3 Metro 2 Reporting - Account Status Code post Liquidation C1- 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 Discharge C1- CIIPSTDIS	279
Table 6–9 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1- CGCLC	280
Table 6–10 Metro 2 Reporting - Consumer Now Located (Removes previously 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/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Organization of the Guide

This document contains:

Chapter 1 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 Platform modules and technology stacks such as DMS / Alert /Email systems, see the Oracle Banking Platform Collections and Recovery Adapter Configuration Guide.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	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.

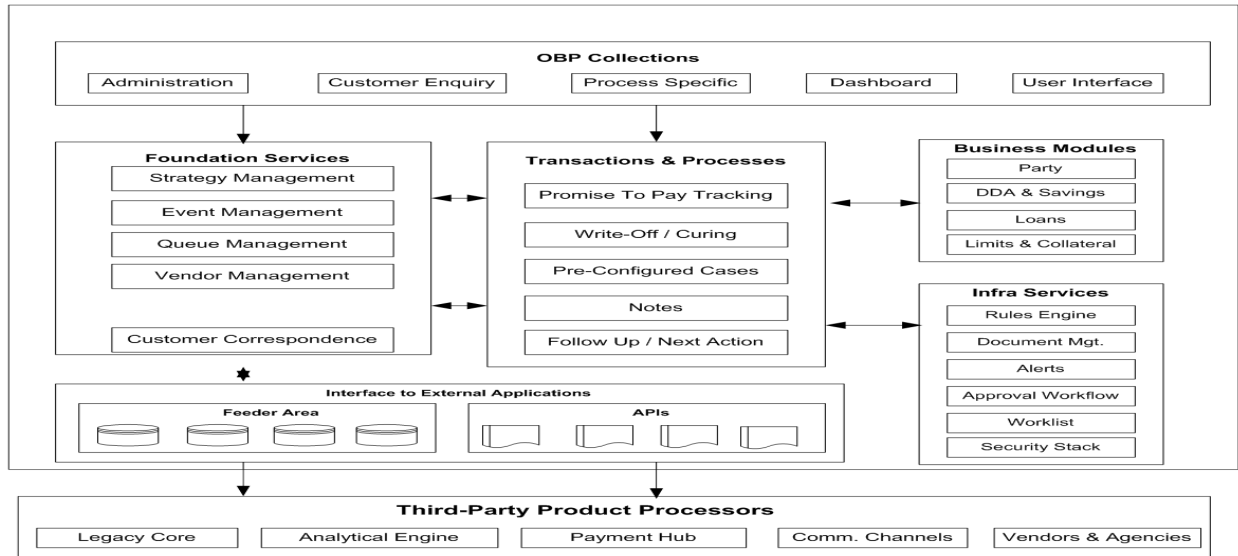
The feeder services exposed by Collections are invoked when changes in data take place in OBP. These services bring modified data into staging tables before batch processes these and update collections tables.

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: First Disbursement Date OD: Date on which OD facility is provided Current Account with TOD facility: TOD utilization Date - Derived		DATE	10	Y	SETUP_DT
Account Currency Code	Currency code of the account		VARCHAR2	3	Y	ACCT_CURR_CD
Outstanding Amount	Outstanding Amount for the account	OD Accounts: OD Limit Utilized + AUF Limit Utilized + Overdue Amount Term Loans : Outstanding Principal - RPA Balance + Overdue Amount	NUMBER	36,18	Y	OUTSTANDING_AMT
Overdue Amount	Overdue amount for the account	OD Accounts: TOD utilized + Overline utilized + Temporary Excess utilized Term Loans : All amounts due and still unpaid	NUMBER	36,18	Y	OVERDUE_AMT
Account Limit	Sanctioned Limit offered to the account	OD Accounts : OD limit + Temporary Excess limit Term	NUMBER	36,18	Y	OVERLIMIT_AMT

Field Name	Description	Values	Data Type	Length	Required	Column Name
		Loans : Sanctioned 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	Exposure at Default : 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	Loss Given Default : 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	Expected Loss : 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	Risk Weighted Asset Calculation : 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 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 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 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 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 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 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 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 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 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 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	Employment Status: 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	Employment Type: 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 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 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 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

4.1 Feeder Tables

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 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 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 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	ASSESSED_VALUE
Assessment Date	Date of assessment		DATE	10	N	ASSESSED_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 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. Default is 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
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 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 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 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 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 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 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

Description	This algorithm type is used to stop the contract.
Detailed Description	Contract Stop Algorithm
Algorithm Entity	Cure Entity
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.CureEntityAlgorithm
Parameters	NA
Detailed Design	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

Description	This algorithm is used to invoke the OBP Services when contract is stopped during the finalize collection process.
Detailed Description	<p>This algorithm performs the following activities:</p> <ul style="list-style-type: none"> - Invoke OBP service to set the incollection flag in host as "N". - Mark incollection flag as "N" in collections. - Set end date in CI_PARTY_COLLECT as posting date. - Update number of times account is self cured (used for statistics). - Remove strategy review date.
Algorithm Entity	Contract Type - Contract Stop
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.FinalizeCollectionContractStopAlgoComp
Parameters	<p>Name: contactMethods</p> <p>Required (Yes/No): Yes</p> <p>Description: Contact Methods soft parameter has a comma-separated value of customer</p>

	contact methods. For example, SMS, EM, and so on. This value is used to calculate the number of self cured statistic.
Detailed Design	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. 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

Algorithm Name	C1-FINCOL
Parameters	Name: contactMethods Value: 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

Description	Allocation Group Queue Allocation.
Detailed Description	This Algorithm type is used to allocate the entities such as cases to queues. For parameter "queueAllocationView" ci_allocation_monitor_vw view is shipped from product to filter cases. For parameter "queueAllocationTable" ci_allocation_monitor table is shipped from product for improvising performance of batch. This is optional parameter.
Algorithm Entity	Allocation Group -Queue Allocation
Program Type	Java
Program Name	Com.splwg.ccb.domain.collection.batch.algorithm.AllocationGroupQueueAlgoComp
Parameters	Name: queueAllocationView (soft parameter) Required (Yes/No): Yes Description: View for allocation Name: queueAllocationTable Required (Yes/No): No Description: Table for allocation
Detailed Design	This algorithm receives input as Allocation Group code from the batch. The view used to filter cases is accepted as an algorithm soft parameter. Product will ship CI_ALLOCATION_MONITOR_VW view. 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

Algorithm Name	C1-ALLOCQUEU
Parameters	Name: queueAllocationView Value: 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

Description	This algorithm is used to update the customer level case switch.
Detailed Description	This algorithm is used to update customer level case status on case enter processing. Customer Level Switch Name: Specify the customer level case status switch that should be updated. For example, BANKRUPT_SW, HARDSHIP_SW, IMPRISONED_SW, DECEASED_SW, ABSCONDING_SW, and so on. Switch Value: Please enter the switch value as Y or N
Algorithm Entity	Case Type - Enter Status
Program Type	java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.CustomerLevelSwitchUpdateAlgorithm
Parameters	Name: Customer Level Switch Name Required (Yes/No): Yes Description: Name of column or switch to be processed Name: Switch Value Required (Yes/No): Yes Description: Y or N
Detailed Design	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. The Customer Level switch name soft parameter accepts the column name that must be updated with switch values as Y or N. 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

Algorithm Name	C1-BRUPTSW
Parameters	Name: Customer Level Switch Name Value: BANKRUPT_SW

	Name: Switch Value Value: 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

Description	This algorithm is used to update Legal and Repo case status on enter processing.
Detailed Description	<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>
Algorithm Entity	Case Type - Enter Status
Program Type	java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.RepoAndLegalCaseUpdateAlgorithm
Parameters	<p>Name: Legal Repo Switch Name Required (Yes/No): Yes Description: Name of column or switch to be processed</p> <p>Name: Switch Value Required (Yes/No): Yes Description: Y or N</p>
Detailed Design	<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"> ■ 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. ■ 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.

Table 5–9 Update Legal/Repo Switch: Sample Algorithm

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

Description	This algorithm is used to allocate cases to users or teams in round-robin method.
Detailed Description	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).
Algorithm Entity	User Allocation
Program Type	java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationRoundRobinAlgorithm
Parameters	NA
Detailed Design	<p>This algorithm receives input as queue code. The computation logic is explained below:</p> <ul style="list-style-type: none"> ■ A1 = Total allocation for the user or team across all queues. ■ B1 = Total capacity of the user or team. This has to be defined in user or collection team configuration. ■ C1 = B1 - A1 = Total available capacity of the user or team. ■ A2 = Existing allocation to the user or team for the current queue. ■ B2 = Capacity of the user or team for the queue. This is defined in queue master. ■ C2 = B2 - A2 = Total available capacity of the user or team for the current queue. ■ Available capacity of the user or team for the queue is lower of C1 and C2. ■ Get all cases which are allocated to the queue and: <ul style="list-style-type: none"> • Have no users or teams attached to it OR • Current allocated user or team does not have active association with the queue ■ Get available capacity for each user or team. ■ 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. ■ A count of freshly allocated cases should be maintained for each user or team. ■ Allocation to a particular user will be skipped if the user is on leave. ■ Allocation to a particular user or team will be skipped if count of newly allocated cases = available capacity. ■ 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.

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

Description	This algorithm is used for allocating cases to users or teams in percentage-based method.																		
Detailed Description	<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>																		
Algorithm Entity	User Allocation																		
Program Type	Java																		
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPerBasedAllocRoundOff																		
Parameters	NA																		
Detailed Design	<ul style="list-style-type: none"> ■ Open unallocated cases will be allocated to valid queue users based on corresponding percentage distribution value mentioned in the Queue Details screen. ■ Algorithm will round off the decimal percent allocated case count to the nearest whole integer value. <p>For example: User allocation percentage of Queue Q1 is as follows:</p> <p>Table 5–12 User allocation percentage of Queue Q1</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 Then, cases will be allocated as per following calculations:</p> <p>Table 5–13 Calculations for allocating cases</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>Table 5–17 Final distribution in cases</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"> ■ If capacities of all queue users are exhausted, unallocated cases will be assigned to the Queue Exception User. 	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

Description	This algorithm is used for allocating cases to vendors in round-robin method.
Detailed Description	This algorithm allocates cases to vendors in round-robin method. This algorithm is invoked from the User Allocation batch (C1-USALC). OverFlow cases will get assigned to Exception User of the queue.
Algorithm Entity	Vendor Allocation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationRoundRobinAlgorithm
Parameters	NA
Detailed Design	<p>This algorithm takes input as Queue code. The computation logic for case capacity is as below:</p> <ul style="list-style-type: none"> ■ A1 = Total existing allocation for the vendor across all queues. ■ B1 = Total capacity of the vendor. This has to be defined in vendor on boarding screen. ■ C1 = B1 - A1 = Total available capacity of the vendor across all service types. ■ A2 = Existing allocation of the vendor for the current queue. ■ B2 = Capacity of the vendor for the queue. This is defined in queue master. ■ C2 = B2 - A2 = Total available capacity of the vendor for the current queue. ■ D1 = Available capacity for number of cases of the vendor for the queue is lower of

	<p>C1 and C2.</p> <ul style="list-style-type: none"> ■ A3 = Existing allocation to the vendor for a service type attached to the vendor. ■ 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. ■ $C3 = B3 - A3$ = Total available capacity for number of cases for a vendor service type. Repeat above steps for each service type attached to the vendor. ■ 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. ■ Get all cases which are allocated to the queue and: <ul style="list-style-type: none"> • Have no vendors attached to it OR • Current allocated vendor does not have active association with the queue. ■ Get "available capacity" of cases of each vendor for each service type attached (A). ■ Get "available capacity" of OS amount of each vendor for each service type attached (B). ■ 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. ■ 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"> • Yes: Allocate if count of newly allocated cases for that service type and OS balance of newly allocated cases for that service type $< A$ and B respectively. If value for B is blank then ignore validating it. • No: Move to next vendor in queue. ■ A count of freshly allocated cases should be maintained for each vendor. ■ 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. ■ 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. ■ 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.
--	--

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

Description	This algorithm is used for allocating cases to vendors in percentage-based method.
Detailed Description	This algorithm allocates cases to vendors in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC). OverFlow cases will get assigned to Exception User of the queue.
Algorithm Entity	Vendor Allocation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationPercentageBaseAlgorithm
Parameters	NA
Detailed Design	<p>This algorithm takes input as Queue code. The computation logic for case capacity is as below:</p> <ul style="list-style-type: none"> ■ A1 = Total existing allocation for the vendor across all queues. ■ B1 = Total capacity of the vendor. This has to be defined in vendor on boarding screen. ■ C1 = B1 - A1 = Total available capacity of the vendor across all service types. ■ D1 = Available capacity for no. of cases of the vendor for the queue is C1. ■ A3 = Existing allocation to the vendor for a service type attached to the vendor. ■ 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. ■ 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. ■ 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. ■ Get all cases which are allocated to the queue and <ul style="list-style-type: none"> • Have no vendors attached to it OR • Current allocated vendor does not have active association with the queue. ■ Calculate % allocation for each vendor in the queue to find maximum cases of new cases that can be allocated to each vendor. ■ Get "available capacity" of cases of each vendor for each service type attached (A). ■ Get "available capacity" of OS amount of each vendor for each service type attached (B). ■ 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

	<p>calculation) and then in decreasing order of capacity.</p> <ul style="list-style-type: none"> ■ 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"> • Yes: Allocate if count of newly allocated cases for that service type and OS balance of newly allocated cases for that service type < A and B respectively. If value for B is blank then ignore validating it • No: Move to next vendor in queue. ■ A count of freshly allocated cases should be maintained for each vendor. ■ 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. ■ 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. ■ 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.
--	--

5.10 Bulk Contact Creation: C1-BLKCNTCRE

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

Table 5–20 Bulk Contact Creation: C1-BLKCNTCRE

Description	Bulk Contact Creation Algorithm
Detailed Description	This algorithm type is called from Bulk Contact Creation Batch. 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.
Algorithm Entity	Bulk contact creation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.BulkContactCreationAlgoComp
Parameters	NA
Detailed Design	This algorithm will be invoked from bulk contact creation batch from where the hard parameter values are set. The algorithm will call business service 'C1-GenerateCorrespondence'. 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

Description	This algorithm is used for Cross Strategy Action Matrix.
Detailed Description	<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>
Algorithm Entity	Case Type- CSAM
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.CrossStrategyActionMatrixAlgorithm
Parameters	<p>Name: CheckStatus Required (Yes/No): N Description: Y - Case types with Status N - Case types without status</p>
Detailed Design	<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"> ■ Close the case: 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. ■ Hold the case: 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". <p>This algorithm should also get triggered during case association process.</p>

Table 5–22 Cross Strategy Action Matrix: Sample Algorithm

Algorithm Name	C1-CSAMY
Parameters	<p>Name: CheckStatus Value: 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

Description	This algorithm is used to update last payment date and amount in account extension table.
Detailed Description	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.
Algorithm Entity	Customer class - FT Freeze
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.LastPaymentDtAmtUpdateAlgorithm
Parameters	NA
Detailed Design	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

Description	This algorithm is used to check if association review is required.
Detailed Description	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. If association Review is not required then transition to specified next status.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckAssociationReview
Parameters	<p>Name: NextStatus Required (Yes/No): N Description: Next Status</p> <p>Name: AssociationReviewRequired Required (Yes/No): Y Description: Association Review Required</p>
Detailed Design	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

Algorithm Name	C1-ASORVCHK
Parameters	<p>Name: NextStatus Value: ASSNEWLSP</p> <p>Name: AssociationReviewRequired Value: 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

Description	This algorithm is used to validate expired default notices.
Detailed Description	<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"> 1. Association Type={P,A}. P=Primary Type Association,A= Primary as well as Secondary type association 2. To Do Type= To Do will be created if validation failure option is N. 3. To Do Role= To Do Role for the specified To Do Type. 4. Validationfailure Option= {Y,N}. If it is Y then case transition will be failed else a To Do will be created.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.DefaultNoticeExpiryCheck
Parameters	<p>Name: associationType Required (Yes/No): Y Description: Association Type</p> <p>Name: validationfailureOption Required (Yes/No): Y Description: Validation Failure Option</p> <p>Name: toDoType Required (Yes/No): N Description: To Do Type</p>
Detailed Design	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

Algorithm Name	C1-DEFNOEXP
Parameters	<p>Name: associationType Value: P</p> <p>Name: validationfailureOption Value: N</p> <p>Name: toDoType Value: 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

Description	This algorithm is used to associate related entities with the case.
Detailed Description	<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"> 1. To Do Role: Specifies the role for the To Do Type created in case of any exception arising in association of accounts. 2. To Do Type: Specifies the To Do Type created in case of any exception arising in association of accounts. 3. Host Id: Specifies the host Id.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.AssociatedAccountsList
Parameters	<p>Name: hostId Required (Yes/No): Y Description: Host Id</p> <p>Name: toDoType Required (Yes/No): Y Description: To Do Type</p>
Detailed Design	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

Algorithm Name	C1-ASSOENTY
Parameters	Name: hostId Value: NGP Name: toDoType Value: 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

Description	This algorithm is used to validate if an active legal case exists at the same time.
Detailed Description	<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"> 1. To Do Role: Specifies the Role for the To Do Type. 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. 3. Case Category: Specifies the case category for the case(LEGL is for Legal Case)
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase
Parameters	Name: Case Category Required (Yes/No): Y Description: Case Category Name: toDoType Required (Yes/No): Y Description: To Do Type
Detailed Design	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

Algorithm Name	C1-ASSOENTY
Parameters	Name: Case Category Value: LEGL

	Name: ToDoType Value: 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

Description	This algorithm is used to assign LSP to the case.
Detailed Description	<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"> 1. Next Status: value can be possible next status example{PREPLGLDOC etc.} 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. 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. 4. Change Allocation Option= Possible values {AUTO_WITH_REVIEW, AUTO_WITHOUT_REVIEW, MANUAL}. AUTO_WITH_REVIEW= System allocation with review option. AUTO_WITHOUT_REVIEW=System allocation without review option. MANUAL=Manual allocation. System will not allocate LSP. 5. New Allocation And Review Option= Possible values {AUTO_WITH_REVIEW,AUTO_WITH_REVIEW_PRVALLOC,AUTO_WITHOUT_REVIEW,MANUAL} AUTO_WITH_REVIEW= System allocation with review option. AUTO_WITH_REVIEW_PRVALLOC=System allocation and review will be required if previous allocation was retained. AUTO_WITHOUT_REVIEW=System allocation without review option. MANUAL=Manual allocation. System will not allocate LSP.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.AssignNewLSP
Parameters	<p>Name: New Allocation And Review Option Required (Yes/No): N Description: New Allocation and Review Option</p> <p>Name: Change Allocation Option</p>

	<p>Required (Yes/No): N Description: Change Allocation Option</p> <p>Name: Reset Document Submission Date Sw Required (Yes/No): N Description: Reset Document Submission Date Switch</p> <p>Name: Previous Allocation Check Required (Yes/No): N Description: Previous Allocation Check</p> <p>Name: Next Status Required (Yes/No): N Description: Next Status</p>
Detailed Design	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

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

Description	This algorithm is used to check the need of approval.
Detailed	This algorithm creates approval request if required based on certain conditions.

Description	<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"> ■ System allocation override by user i.e. user has changed the LSP assigned by the system. Set Approval Reason as "Allocation override". ■ 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". ■ In case approval is required for both the reason, concatenate the approval reasons before sending for approval. <p>If approval is required:</p> <ul style="list-style-type: none"> ■ Transition the case to a specified status defined as the parameter.
Algorithm Entity	Case Type - Enter Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CreateApprovalRequest
Parameters	<p>Name: Exposure Threshold Required (Yes/No): N Description: Exposure Threshold</p> <p>Name: Approval Request Status Required (Yes/No): N Description: Approval Request Status</p> <p>Name: Approved Status Required (Yes/No): N Description: Approved Status</p> <p>Name: Reject Request Status Required (Yes/No): N Description: Reject Request Status</p>
Detailed Design	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

Algorithm Name	C1-ASGNLSP
Parameters	<p>Name: Exposure Threshold Value: 10</p>

	<p>Name: Approval Request Status Value: PENDINGAPP</p> <p>Name: Approved Status Value: WTFRLSPACK</p> <p>Name: Reject Request Status Value: 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

Description	This algorithm is used to resume status from previous LSP.
Detailed Description	This algorithm resumes the previous state stored while changing LSP.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.ResumeStatusLSP
Parameters	NA
Detailed Design	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

Description	This algorithm is used to check submission date.
Detailed Description	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.
Algorithm Entity	Case Auto Transition Validation
Program Type	Java

Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDate
Parameters	<p>Name: nextStatus Required (Yes/No): Y Description: NA</p> <p>Name: changeStatus Required (Yes/No): Y Description: NA</p>
Detailed Design	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

Algorithm Name	CI_CHKSUBDT1C1
Parameters	<p>Name: nextStatus Value: WTFRLSPACK</p> <p>Name: changeStatus Value: 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

Description	Legal Proceedings - Update Status
Detailed Description	<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 REJ=Rejected PNAP=Pending for Approval.</p>
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssign

Name	nment
Parameters	Name: Lsp Assignment Status Required (Yes/No): Y Description: LSP Assignment Status
Detailed Design	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

Algorithm Name	C1-LSPSTATUS
Parameters	Name: : Lsp Assignment Status Value: 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

Description	Legal Proceedings - Update Status
Detailed Description	This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either closed or cancelled.
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssignment
Parameters	Name: Lsp Assignment Status Required (Yes/No): Y Description: NA
Detailed Design	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

Algorithm Name	C1-LSPSTACAN
Parameters	Name: : Lsp Assignment Status Value: 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

Description	Collateral Verification
Detailed Description	<p>This will perform following validations for the collateral with the case:</p> <ul style="list-style-type: none"> ■ 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. ■ If collateral type soft parameter is blank, then above validation should be ignored and Collateral status is set to Not Sold. ■ 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.
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification
Parameters	<p>Name: Collateral Type Required (Yes/No): N Description: Collateral Type</p>
Detailed Design	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

Algorithm Name	C1-VRFYCOLS
Parameters	<p>Name: Collateral Type Value: 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

Description	Account Association for Asset repossession case
Detailed Description	<p>This algorithm will perform following actions:</p> <ul style="list-style-type: none"> ■ It gets all facilities to which this collateral is associated and all accounts for these facilities.

	<ul style="list-style-type: none"> ■ It associates these accounts with the case. <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>
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AccountAssociationForAssetRepossessionCase
Parameters	NA
Detailed Design	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

Description	Customer Association for Asset repossession case
Detailed Description	<p>This algorithm performs the following actions:</p> <ul style="list-style-type: none"> ■ It gets all customers who are the owners for the selected collateral ■ It associates these customers with the case <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>
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CustomerAssociationForAssetRepossessionCase
Parameters	NA

Detailed Design	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

Description	Update Collateral Property
Detailed Description	<p>This algorithm will perform following operations:</p> <ul style="list-style-type: none"> ■ 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. ■ 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.
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralProperty
Parameters	<p>Name: UpdateCollateralProperty Required (Yes/No): Y Description: NA</p>
Detailed Design	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

Description	Close To do's algorithm
Detailed Description	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.
Algorithm Entity	Case Type-Exit Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo
Parameters	Name: To Do Type1

	<p>Required (Yes/No): N Description: To Do Type 1</p> <p>Name: To Do Type2 Required (Yes/No): N Description: To Do Type 2</p> <p>Name: To Do Type3 Required (Yes/No): N Description: To Do Type 3</p> <p>Name: To Do Type4 Required (Yes/No): N Description: To Do Type 4</p> <p>Name: To Do Type5 Required (Yes/No): N Description: To Do Type 5</p>
Detailed Design	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

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

Description	Validations for Mandatory Characteristics
Detailed Description	Subjective Validations for Mandatory Characteristics
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics
Parameters	<p>Name: ReferenceCharacteristicsValue Required (Yes/No): Y Description: Reference Characteristics Value</p> <p>Name: ReferenceCharacteristic Required (Yes/No): Y Description: Reference Characteristic Type</p> <p>Name: CaseCharacteristics1 Required (Yes/No): N Description: Case Characteristics</p> <p>Name: CaseCharacteristics2 Required (Yes/No): N Description: Case Characteristics</p> <p>Name: CaseCharacteristics3 Required (Yes/No): N Description: Case Characteristics</p> <p>Name: CaseCharacteristics4 Required (Yes/No): N Description: Case Characteristics</p> <p>Name: CaseCharacteristics5 Required (Yes/No): N Description: Case Characteristics</p>
Detailed Design	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

Algorithm Name	C1-CHARVALS
Parameters	Name: ReferenceCharacteristicsValue Value: Type of Possession
	Name: ReferenceCharacteristic Value: Voluntary Possession
	Name: CaseCharacteristics1 Value: Vacancy Date
	Name: CaseCharacteristics2 Value: Vacancy Possession Indemnity Policy Reference
	Name: CaseCharacteristics3 Value: Property Surrender Letter Reference
	Name: CaseCharacteristics4 Value: Property Surrender Letter Reference
	Name: CaseCharacteristics5 Value:

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

Description	Validations for Mandatory Characteristics
Detailed Description	Subjective Validations for Mandatory Characteristics
Algorithm Entity	Case Type-Exit Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics
Parameters	Name: ReferenceCharacteristicsValue Required (Yes/No): Y Description: NA

	<p>Name: ReferenceCharacteristic Required (Yes/No): Y Description: NA</p> <p>Name: CaseCharacteristics1 Required (Yes/No): N Description: NA</p> <p>Name: CaseCharacteristics2 Required (Yes/No): N Description: NA</p> <p>Name: CaseCharacteristics3 Required (Yes/No): N Description: NA</p> <p>Name: CaseCharacteristics4 Required (Yes/No): N Description: NA</p> <p>Name: CaseCharacteristics5 Required (Yes/No): N Description: NA</p>
Detailed Design	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

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

	Value: Name: CaseCharacteristics4 Value: Name: CaseCharacteristics5 Value:
--	---

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

Description	Update Collateral Status in the host
Detailed Description	<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>
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost
Parameters	<p>Name: To Do Type Required (Yes/No): Y Description: To Do Type</p> <p>Name: Collateral Status Required (Yes/No): Y Description: Collateral Status</p>
Detailed Design	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

Algorithm Name	C1-UPCOLLSTX
Parameters	<p>Name: To Do Type Value: C1-TD-UC</p> <p>Name: Collateral Status Value: 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

Description	Initiate collateral valuation
Detailed Description	<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) <= 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"> ■ A To-do of same to-do type is already open for the case ■ A To-do of same to-do type was closed within past "Y" days <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>
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation
Parameters	<p>Name: To Do Type Required (Yes/No): Y Description: To Do Type</p> <p>Name: Days Since Closure Of Last To Do Required (Yes/No): Y Description: Days Since Closure Of Last To Do</p> <p>Name: Assessment Expiry Days Required (Yes/No): Y Description: Assessment Expiry Days</p>
Detailed Design	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

Algorithm Name	C1-COLLVALX
Parameters	<p>Name: To Do Type Value: C1-TD-UC</p>

	Name: Days Since Closure Of Last To Do Value: 5 Name: Assessment Expiry Days Value: 5
--	--

5.32 Validation Settlement: C1-VALSET

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

Table 5–58 Validation Settlement: C1-VALSET

Description	Validation Settlement
Detailed Description	<p>This algorithm will perform following actions: Before completing the asset repossession case, the below validations should be done for the case:</p> <ul style="list-style-type: none"> ■ Collateral should have a settlement date ■ Realization status for the collateral should be Complete <p>Transition to completed status will fail if above validations fail.</p>
Algorithm Entity	Case Type-Exit Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateralSettlementStatus
Parameters	Name: Realization Status Required (Yes/No): Y Description: Realization Status
Detailed Design	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

Algorithm Name	C1-VALSET
Parameters	Name: Realization Status Value: REALIZATION_COMPLETE

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

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

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

Description	PTP Active Algorithm
Detailed Description	<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>
Algorithm Entity	PTP Active Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm
Parameters	<p>Name: contactTypeForLetter Required (Yes/No): No Description: Contact Type for Letter generation</p> <p>Name: contactClassForLetter Required (Yes/No): No Description: Contact Class for letter generation</p> <p>Name: contactMethodForLetter Required (Yes/No): No</p>

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

Table 5–63 PTP Active Algorithm: Sample Algorithm

Algorithm Name	C1-PTPKEPT
Parameters	<p>Name: contactTypeForLetter Value: OVERDUE</p> <p>Name: contactClassForLetter Value: CCC</p> <p>Name: contactMethodForLetter Value: OTBL</p> <p>Name: contactTypeForSMS Value: OVERDUE</p> <p>Name: contactClassForSMS Value: CCC</p> <p>Name: contactMethodForSMS Value: 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

Description	PTP Kept Algorithm
--------------------	--------------------

Detailed Description	This algorithm is used to perform additional processing when the status of a PTP becomes Kept. Customer Contacts can be generated via this algorithm. Contact Class, method and type have to be specified.
Algorithm Entity	PTP Kept Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPKeptForNgpAlgorithm
Parameters	<p>Name: contactTypeForLetter Required (Yes/No): No Description: Contact Type for Letter generation</p> <p>Name: contactClassForLetter Required (Yes/No): No Description: Contact Class for letter generation</p> <p>Name: contactMethodForLetter Required (Yes/No): No Description: Contact Method for Letter generation</p> <p>Name: contactTypeForSMS Required (Yes/No): No Description: Contact Type for SMS</p> <p>Name: contactClassForSMS Required (Yes/No): No Description: Contact Class for SMS</p> <p>Name: contactMethodForSMS Required (Yes/No): No Description: Contact Method for SMS</p>
Detailed Design	This algorithm invokes GenerateContactForPTP service, which creates the contact (generate Letter or SMS) when PTP moves to Kept state.

Table 5–65 PTP Active Algorithm: Sample Algorithm

Algorithm Name	C1-PTPKEPT
Parameters	<p>Name: contactTypeForLetter Value: OVERDUE</p> <p>Name: contactClassForLetter Value: CCC</p>

	<p>Name: contactMethodForLetter Value: OTBL</p> <p>Name: contactTypeForSMS Value: OVERDUE</p> <p>Name: contactClassForSMS Value: CCC</p> <p>Name: contactMethodForSMS Value: 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

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

	<p>Name: contactMethodForSMS Required (Yes/No): No Description: Contact Type for SMS generation</p> <p>Name: contactClassForSMS Required (Yes/No): No Description: Contact Class for SMS generation</p> <p>Name: contactMethodForSMS Required (Yes/No): No Description: Contact Method for SMS generation</p>
Detailed Design	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

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

	<p>Required (Yes/No): No Description: Primary Key name of defined BO.</p>
Parameters	<p>Name: Input B O Name1 Required (Yes/No): Yes Description: 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>Name: Input B O Name2 Required (Yes/No): No Description: 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>Name: Input B O Name3 Required (Yes/No): No Description: 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>Name: Input B O Name4 Required (Yes/No): No Description: 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>Name: Input B O Name5 Required (Yes/No): No Description: 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>Name: Bo Fields Required (Yes/No): Yes Description: Comma separated BO fields of defined BO names.</p> <p>Name: Rule Fact Codes Required (Yes/No): Yes Description: Comma separated fact codes for rule to be executed. BO Fields and Rule Fact codes should be defined in the same order.</p> <p>Name: Pre Populated Rule Facts Algorithm Code Required (Yes/No): No Description: 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>
Detailed Design	<p>This algorithm is used to populate rule facts from Business object (BO). 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

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

	Name: Pre Populated Rule Facts Algorithm Code Value:
--	---

5.38 Borrower Centric Case Lifecycle

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

Table 5–69 Borrower Level: C1-ASSODELAC

Description	Associate new delinquent account of the customer
Detailed Description	Associate delinquent accounts where the customer is the main customer to the case.
Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssociateDelinquentAccount
Parameters	NA
Detailed Design	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

Algorithm Name	C1-BRWRSW_Y
Parameters	Name: Customer Level Switch Name Value: BRRWR_SW Name: Switch Value Value: Y

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

Table 5–71 Borrower Level : C1-BRWTRNDF

Algorithm Name	C1-BRWTRNDF
Parameters	Name: Wait Days Value: 0

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

Table 5–72 Borrower Level : C1-BRWRSW_N

Algorithm Name	C1-BRWRSW_N
Parameters	Name: Customer Level Switch Name Value: BRRWR_SW Name: Switch Value Value: 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

Description	Person Address Update - Pre Processing
Detailed Description	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.
Algorithm Entity	Business Object -Pre-Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPreProcess
Parameters	NA
Detailed Design	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

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

Description	Person Contact Point Update - Pre Processing
Description Description	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.
Algorithm Entity	Business Object - Pre Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.ContactPreferencePreProcess
Parameters	NA
Detailed Design	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

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

	<p>Name: Case Type Required (Yes/No): Yes Description: Case Type</p> <p>Name: Consider Enterprise Id Required (Yes/No): Yes Description: Enterprise Id</p>
Detailed Design	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

Description	Pull all the non delinquent accounts of the customer into collections- Enter Processing
Detailed Description	<p>Processing steps are as below:</p> <ul style="list-style-type: none"> ■ Pull all Not in Collections accounts into OB Collections (from OBP) where the associated customer is one of the borrower. ■ If Account Relationships = MC consider only the accounts where the customer is primary owner. If Account Relationships = FO consider all accounts where the customer is a financial owner. If Account Relationship = All consider all accounts where the customer is a financial or non-financial owner. ■ 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. Possible Values of Account Relationships MC, FO, ALL Possible Values for Consider Enterprise Id Yes/No
Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyPullNonDelinquentAcc
Parameters	<p>Name: Account Relationships (MC,FO,ALL) Required (Yes/No): Yes Description: Account Relationships Name: Consider Enterprise Id (Yes/No)</p>

	Required (Yes/No): Yes Description: Enterprise Id
Detailed Design	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

Description	Associate all accounts to the case where customer is a primary borrower.
Detailed Description	Associate all accounts to the case where customer is a primary borrower For the primary customer associated with the case: <ul style="list-style-type: none"> ■ 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). ■ Shortlist the accounts that are not yet associated with the case. ■ Associate the shortlisted accounts with the case. Consider Enterprise Id value should be "YES" or "NO"
Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyAssociateAcc
Parameters	Name: Consider Enterprise Id Required (Yes/No): Yes Description: Consider Enterprise Id
Detailed Design	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

Description	Exclude all the associated accounts from Dialer- Enter Processing
Detailed Description	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. For all the accounts associated with the case: <ul style="list-style-type: none"> ■ Call the Dialer Exclusion Service to exclude the accounts from feed to Dialer

Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyExcludeAccDir
Parameters	NA
Detailed Design	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

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

	<p>Required (Yes/No): No Description: Exclude Collateral Types</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	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

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

	<p>Name: Charge Off Threshold D P D Required (Yes/No): Yes Description: Charge Off Threshold D P D</p> <p>Name: Secured Accounts Required (Yes/No): Yes Description: Secured Accounts</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	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

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

Description	Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring
Detailed Description	<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: <PTP Type> broken for account <Account Number>. Days since plan broken <Days Since PTP Broken>.</p> <p>Set Display Date of the case to current business date.</p> <p>Values of Validation Date: POSTING DATE, SYSTEM DATE</p>
Algorithm Entity	Case Type -Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorPaymentPlan
Parameters	<p>Name: P T P Type Required (Yes/No): Yes Description: P T P Type</p> <p>Name: Days Since P T P Broken Required (Yes/No): Yes Description: Days Since P T P Broken</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	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

Description	Notify the Bankruptcy Specialist if the Liquidation reaches a specific status.
Detailed Description	<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 <Account Number>; Collateral <Collateral Code> has reached status</p>

	<Case Status> Set Display Date of the Bankruptcy Case to Business Date
Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorAssetLiquidation
Parameters	Name: Validation Date Required (Yes/No): Yes Description: Validation Date
Detailed Design	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

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

Description	Determine in which status the case should proceed for Bankruptcy Treatment - Post Processing
Detailed Description	<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 Then Transition to Manage Chapter 7 Bankruptcy Status</p> <p>If Bankruptcy Chapter = Chapter 13 Then Transition to Manage Chapter 13 Bankruptcy Status</p> <p>If Bankruptcy Chapter = Chapter other than 7 or 13 Then Transition to Other Bankruptcy Status</p> <p>Bankruptcy Chapter Field = "FC" or "CC" Where "FC" = Filing Chapter and "CC"=Convert to chapter</p>
Algorithm Entity	Result Type -Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.DetermineBankruptcyTreatment
Parameters	<p>Name:Bankruptcy Chapter Field Required (Yes/No): Yes Description: Bankruptcy Chapter Field</p> <p>Name:Manage Chapter7 Bankruptcy Status Required (Yes/No): Yes Description: Manage Chapter7 Bankruptcy Status</p> <p>Name:Manage Chapter13 Bankruptcy Status Required (Yes/No): Yes Description: Manage Chapter13 Bankruptcy Status</p> <p>Name:Other Bankruptcy Status Required (Yes/No): Yes Description: Other Bankruptcy Status</p>
Detailed Design	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-VLDBC DATA

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

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

	Required (Yes/No): No Description: Dynamic Panel 5 Fields Name: Case Characteristics Required (Yes/No): Description: Case Characteristics
Detailed Design	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

Description	Notify Bankruptcy Specialist when a Payment Plan status becomes Kept
Detailed Description	Create Notification Notification: <PTP Type> Kept for account <Account Number>. Set Display Date of the case to current business date.
Algorithm Entity	Business Object -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyPaymentPlanKept
Parameters	NA
Detailed Design	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

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

Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyTaskCompletion
Parameters	NA
Detailed Design	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

Description	Joint Bankruptcy - Associate other customers to the Bankruptcy case
Detailed Description	Associate additional customers specified on the UI that exist in OB Collections. (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)
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.JointBankruptcyAssociateCust
Parameters	NA
Detailed Design	Associate additional customers specified on the UI that exist in OB Collections. (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

Description	Algorithm to create Pay Plan for a Bankruptcy Case
Detailed Description	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 : <ol style="list-style-type: none"> 1. CI_BKPTCY_PAY_PLAN_INFO 2. CI_BKPTCY_PAY_PLAN_DTLS
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BankruptcyPayPlanCreation
Parameters	NA

Detailed Design	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

Description	Algorithm to close Arrearage Pay Plan for a Bankruptcy Case
Detailed Description	<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"> 1. CI_BKPTCY_PAY_PLAN_INFO 2. CI_BKPTCY_PAY_PLAN_DTLS 3. CI_BKPTCY_PAY_PLAN_SCHED <p>Associate additional customers specified on the UI that exist in OB Collections. (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>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BkptcyPayPlanClosure
Parameters	NA
Detailed Design	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

Description	Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days
Detailed Description	<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"> Arrearage Plan Threshold Days Arrearage Plan Threshold Amount Confirmed Plan Threshold Days Confirmed Plan Threshold Amount Notification Date Type <p>Notification is generated as -></p>

	<Arrearage/Confirmed Plan> amount for Account Number <account no> of <currency symbol> <overdue amount> is overdue by <overdue no of days> Days
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Banckruptcy MonitorArrearagePlanNotification
Parameters	<p>Name: Arrearage Plan Threshold Days Required (Yes/No): Yes Description: Arrearage Plan Threshold Days</p> <p>Name: Arrearage Plan Threshold Amount Required (Yes/No): Yes Description: Arrearage Plan Threshold Amount</p> <p>Name: Confirmed Plan Threshold Days Required (Yes/No): Yes Description: Confirmed Plan Threshold Days</p> <p>Name: Confirmed Plan Threshold Amount Required (Yes/No): Yes Description: Confirmed Plan Threshold Amount</p> <p>Name: Notification Date Type Required (Yes/No): Yes Description: Notification Date Type</p>
Detailed Design	<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"> 1. Arrearage Plan Threshold Days 2. Arrearage Plan Threshold Amount 3. Confirmed Plan Threshold Days 4. Confirmed Plan Threshold Amount 5. Notification Date Type <p>Notification is generated as -> <Arrearage/Confirmed Plan> amount for Account Number <account no> of <currency symbol> <overdue amount> is overdue by <overdue no of days> Days</p>

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

Description	Set or Reset Account level Warning Indicator for Bankruptcy
Detail	This Algorithm Set or Reset the Account level Warning Indicators of all the associated accounts

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

Description	Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing
Detailed Description	On creation of task check if task is already allocated to a member. 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.
Algorithm	TO DO Type - Post Processing

Entity	
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.vendor.VendorManagementAutomaticTaskAllocation
Parameters	NA
Detailed Design	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

Description	Hardship Entity Association Pending State - Enter Processing
Detailed Description	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.
Algorithm Entity	Case Type -Enter Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipAssociation
Parameters	NA
Detailed Design	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

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

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

Description	Park accounts with small balances to a separate status - Monitoring
Detailed Description	<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 <= Small Balance Threshold And Net Arrear Amount > 0 Then transition to small balance status.</p> <p>Net Arrear Amount = (Overdue Amount - Unclear Amount)</p> <p>If Use Overdue Amount = Yes then use Overdue Amount instead of Net Arrear Amount in the calculations.</p> <p>Set Re-Allocation Switch = Y for the case post case transition.</p> <p>Possible Values : Use Overdue Amount : Y,N</p>
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ParkSmallBalanceAccounts
Parameters	<p>Name: Small Balance Threshold Required (Yes/No): No Description: Small Balance Threshold</p> <p>Name: Small Balance Status Required (Yes/No): No Description: Small Balance Status</p> <p>Name: Use Overdue Amount Required (Yes/No): No Description: Use Overdue Amount</p>
Detailed Design	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

Description	Transition to skip tracking status if no telephone number exists for any of the account holder - Enter Processing
Detailed Description	<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.
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracing
Parameters	<p>Name: Contact Points Required (Yes/No): No Description: Contact Points</p> <p>Name: Skip Tracing Status Required (Yes/No): No Description: Skip Tracing Status</p>
Detailed Design	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

Description	Transition to Suspended status based on Account and Party Risk Indicators - Monitoring
Detailed Description	<p>If the Account has one of the Account Risk Indicators specified in the parameter Transition to Suspended status. Create a task if Task Type has been mentioned and assign it to the Specified Queue Set Re-Allocation Switch = Y for the case post case transition. Set Suspend Reason = Risk Indicator Exit.</p> <p>If either of the financial owners have one of the Party Indicators mentioned in the parameter than transition to Suspended status. Create a task if Task Type has been mentioned and assign it to the Specified Queue Set Re-Allocation Switch = Y for the case post case transition. Set Suspend Reason = Risk Indicator 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. Create a task if Task Type has been mentioned and assign it to the Specified Queue. Set Re-Allocation Switch = Y for the case post case transition. Set Alternate Contact Flag = Y Set Alternate Contact Reason = Risk Indicator If case already in Contact Alternate status don't initiate any activities. Exit.</p>
Algorithm Entity	Case Type-Auto Transition
Program Type	Java

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

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

Description	Check if the contact cap has reached for the case 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 >= 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>
Detailed Description	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.
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ValidateContactCap
Parameters	<p>Name: Contact Cap Required (Yes/No): No Description: Contact Cap</p> <p>Name: Contact Cap Duration (X) Required (Yes/No): No Description: Contact Cap Duration (X)</p> <p>Name: Contact Hold Days (Y) Required (Yes/No): No Description: Contact Hold Days (Y)</p> <p>Name: Contact Methods Required (Yes/No): No Description: Contact Methods</p> <p>Name: Hold Reason Required (Yes/No): No Description: Hold Reason</p> <p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

Table 5–102 Schedule Contact - Monitoring C1-ECSC

Description	Schedule Contact - Monitoring
Detailed Description	Schedule Contact for the case as per intensity If case is not on Hold

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

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

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

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

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

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

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

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

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

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

Description	Resume Contact from Small Balance- Monitoring
Detailed Description	<p>.This algorithm is used to resume contact from small balance status. If Net Arrear Amount > Small Balance Threshold Then transition the case to Contact RM Status if RM exists and Contact RM status has been configured Contact Alternate Status If Contact Alternate Flag = Y Else Contact Status Set Re-Allocation Switch = Y for the case post case transition 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) Possible Value: Overdue Amount : Y,N
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactfromSmallBalance
Parameters	<p>Name: Contact Status Required (Yes/No): No Description: Contact Status</p> <p>Name: Contact RM Status Required (Yes/No): No Description: Contact RM Status</p> <p>Name: Contact Alternate Status Required (Yes/No): No Description: Contact Alternate Status</p> <p>Name: Use Overdue Amount Required (Yes/No): No Description: Use Overdue Amount</p> <p>Name: Small Balance Threshold (Yes/No): No Description: Small Balance Threshold</p>
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

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

Description	Determine Contact Intensity and Contact Intensity Review Date -Monitoring
Detailed Description	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.
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.DetermineContactIntensity

Parameters	<p>Name: Contact Intensity Rule Required (Yes/No): No Description: Contact Intensity Rule</p> <p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>
Detailed Design	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

Description	Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing
Detailed Description	<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 <Result Type> 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"> ■ Create Task of given Task Type and assign it to the give Task Queue if Task Type is configured. ■ Map the created task with the Follow up Id of the Follow Up that created the task. ■ Set Re-Allocation Switch = Y if Re-Allocate = Y ■ 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) ■ Task Creation Logic: If Task For = Account Create Task on the primary associated account on the case If Task For = Customer Create Task on the primary associated customer of the case If Task For = Case Create Task on the case If Task For = Admin Create Admin level Task Note: Task For is the mandatory characteristic at Task Level Task For = Customer is an invalid configuration for Account level Case and vice versa <p>Possible Values of Re-Allocate Switch and Copy Characteristics to Case are : Y/N "Event Name" and "Action Flag" fields are introduced to update Cease_Desist\Contact_Alternate\Dispute Flags, where:- "Event Name" will be provided depending on the FLAG which you need to update. So, it can have one of the values:- Event Name :- "CEASE_DESIST" Event Name :- "CONTACT_ALT"</p>

	<p>Event Name :- "DISPUTE" And "Action Flag" value will be SET\RESET. To set Cease_Desist\Contact_Alternate\Dispute Flags to "Y", provide Action Flag :- "SET". To set Cease_Desist\Contact_Alternate\Dispute Flags "N", provide Action Flag :- "RESET". To transit case to other status from contact alternate status and remove contact alternate details set Contact Alternate Status to contact alternate case status. Contact Alternate Case Status soft parameter usage is as follow : When user want to stop contact alternate details on an account in that scenario :</p> <ul style="list-style-type: none"> ■ 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 ■ 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.
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseTransitionandTraskCreationPostProcessingAlgo
Parameters	<p>Name: Case Status Required (Yes/No): No Description: Case Status</p> <p>Name: Valid Current Status Required (Yes/No): No Description: Valid Current Status</p> <p>Name: Task Type Required (Yes/No): No Description: Task Type</p> <p>Name: Queue Required (Yes/No): No Description: Queue</p> <p>Name: Re-Allocate Switch Required (Yes/No): No Description: Re-Allocate Switch</p>

	<p>Name: Copy Characteristics to Case Required (Yes/No): No Description: Copy Characteristics to Case</p> <p>Name: Event Name Required (Yes/No): No Description: Event Name</p>
Parameters (Cont.)	<p>Name: Action Flag Required (Yes/No): No Description: Action Flag</p> <p>Name: Contact Alternate Case Status Required (Yes/No): No Description: Contact Alternate Case Status</p>
Detailed Design	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

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

	<p>Description: Valid Current Status</p> <p>Name: Case Status Required (Yes/No): No Description: Case Status</p> <p>Name: Task Type Required (Yes/No): Yes Description: Task Type</p> <p>Name: Re-Allocate Required (Yes/No): Yes Description: Re-Allocate</p>
Detailed Design	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

Description	Resume Collections- Result Type - Post Processing
Detailed Description	<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>
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeCollectionsPostProcessingAlgo
Parameters	<p>Name: Contact Status Required (Yes/No): No Description: Contact Status</p> <p>Name: Contact RM Status Required (Yes/No): No Description: Contact RM Status</p> <p>Name: Contact Alternate Status</p>

	<p>Required (Yes/No): No Description: Contact Alternate Status</p> <p>Name: Re-Allocate Required (Yes/No): No Description: Re-Allocate</p>
Detailed Design	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

Description	Create case on Follow up - Post Processing
Detailed Description	<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>
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseCreationonFollowupPostProcessingAlgo
Parameters	<p>Name: Case Type Required (Yes/No): Yes Description: Case Type</p> <p>Name: Queue Type Required (Yes/No): No Description: Queue Type</p>
Detailed Design	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

Description	Hold Case - Post Processing
Detailed Description	<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>

Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.HoldCasePostProcessingAlgo
Parameters	<p>Name: Hold Period Characteristic Type Required (Yes/No): No Description: Hold Period Characteristic Type</p> <p>Name: Hold Period Required (Yes/No): No Description: Hold Period</p> <p>Name: Hold Reason Characteristic Type Required (Yes/No): No Description: Hold Reason Characteristic Type</p> <p>Name: Hold Reason Required (Yes/No): No Description: Hold Reason</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	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

Description	Update Case Level Data when a case enters a new status - Enter Processing
Detailed Description	<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>
Algorithm Entity	Case Status - Enter Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateCaseData
Parameters	<p>Name: Char Type - 1 Required (Yes/No): No Description: Char Type - 1</p>

	<p>Name: Char Type - 2 Required (Yes/No): No Description: Char Type - 2</p> <p>Name: Char Type - 3 Required (Yes/No): No Description: Char Type - 3</p> <p>Name: Char Type - 4 Required (Yes/No): No Description: Char Type - 4</p> <p>Name: Char Type - 5 Required (Yes/No): No Description: Char Type - 5</p> <p>Name: Char Value - 1 Required (Yes/No): No Description: Char Value - 1</p> <p>Name: Char Value - 2 Required (Yes/No): No Description: Char Value - 2</p> <p>Name: Char Value - 3 Required (Yes/No): No Description: Char Value - 3</p> <p>Name: Char Value - 4 Required (Yes/No): No Description: Char Value - 4</p> <p>Name: Char Value - 5 Required (Yes/No): No Description: Char Value - 5</p>
Detailed Design	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

Description	This algorithm will transition the case status to the Suspension status if Cease and Desist = Y
--------------------	---

Detailed Description	Additional algorithm in Pending Status: Enter Processing to transition to Suspend Status if Cease and Desist = Y.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CeaseDesistAccountSuspension
Parameters	Name: Suspension State Required (Yes/No): Yes Description: Suspension State
Detailed Design	Additional algorithm in Pending Status: Enter Processing to transition to Suspend Status if Cease and Desist = Y.
Detailed Design	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

Description	Algorithm is used for scheduling call
Detailed Description	<p>This algorithm is used to fulfil request by customer to collector for calling at specific time.</p> <ul style="list-style-type: none"> ■ 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. ■ 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.
Algorithm Entity	Result Type - Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.algorithms.ScheduleCallPostProcessingAlgorithm
Parameters	Name: Online Dialer Inclusion Required (Yes/No): No Description: Online Dialer Inclusion Name: Preferred Time Char Required (Yes/No): Yes Description: Preferred Time Char Name: Validation Date Required (Yes/No): Yes Description: Validation Date

Detailed Design	<p>This algorithm is used to fulfil request by customer to collector for calling at specific time.</p> <ul style="list-style-type: none"> ■ 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. ■ 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.
------------------------	--

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

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

5.45 Asset Repossession

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

Description	Validate Collateral - Enter Status Validation
Detailed Description	<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>

	<p>The collateral in not already sold or under repossession."</p> <p>Collateral Type: It should be allow any collateral type as a parameter. (For eg: AUTOMOBILE,PROPERTY etc)</p> <p>Collateral Category :Category of collateral(For eg: Vehicle)</p> <p>Both parameter accept comma separated values. CI_COLLATERAL table have both columns.</p>
Algorithm Entity	Case Status - Enter Status Validation
Program Type	Java
Program Name	om.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateral
Parameters	<p>Name: Collateral Type Required (Yes/No): No Description: Collateral Type</p> <p>Name: Collateral Category Required (Yes/No): No Description: Collateral Category</p>
Detailed Design	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

Description	Validate if Demand Letter and Acceleration letter have been sent - Enter Status Validation
Detailed Description	<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"> 1. Demand Letter Template Code - Demand Letter Template Code 2. Acceleration Letter Template Code - Acceleration Letter Template Code 3. Number Of Days in which Demand Letter or Acceleration Letter send - number of days 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 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"
Algorithm Entity	Case Status - Enter Status Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ValidateDemandLetterandAccelerationLetter
Parameters	<p>Name: Demand Letter Template Code Required (Yes/No): No Description: Demand Letter Template Code</p> <p>CodeName: Acceleration Letter Template Code Required (Yes/No): No Description: Acceleration Letter Template Code</p> <p>Name: Number Of Days in which Demand Letter or Acceleration Letter send Required (Yes/No): No Description: Number Of Days in which Demand Letter or Acceleration Letter send</p> <p>Name: Only PrimaryAccount Switch Required (Yes/No): No Description: Only PrimaryAccount Switch</p> <p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>
Detailed Design	Validate if Demand Letter and Acceleration letter have been sent

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

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

Name	ustAssRepo
Parameters	No Parameters
Detailed Design	Associate all financial owners on the associated accounts to the Repossession case.

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

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

Description	Monitor if Demand letter and Acceleration letter have been sent on the account.
Detailed Description	<p>If DL Template Code has been mentioned validate if Demand letter has been sent and current date > Demand Letter Expiry Date. If AL Template Code has been mentioned validate if Acceleration letter has been sent and the current date > Acceleration letter Expiry Date. 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. If both are true transition the case to Repossession Referred Status. Parameter Description as follows :</p>

	<ol style="list-style-type: none"> 1. Demand Letter Template Code - Demand Letter Template Code 2. Acceleration Letter Template Code - Acceleration Letter Template Code 3. Repossession Referred Status - Repossession referred status code 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 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
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.MonitorDemandLetterandAccelerationLetterExpiry
Parameters	<p>Name: Demand Letter Template Code Required (Yes/No): No Description: Demand Letter Template Code</p> <p>Name: Acceleration Template Code Required (Yes/No): No Description: Acceleration Template Code</p> <p>Name: Repossession Referred Status Required (Yes/No): No Description: Repossession Referred Status</p> <p>Name: Primary Account Sw Required (Yes/No): No Description: Primary Account Sw</p> <p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>
Detailed Design	Monitor if Demand letter and Acceleration letter have been sent on the account.

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

Description	Auto Approval Check for Repossession
Detailed Description	<p>If the Auto- Approval Rule returns true the case will be transitioned to the Approved status. 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"> ■ Collateral Type ■ Collateral Category ■ Repossession Reason ■ Outstanding Amount ■ Overdue Amount ■ Days Past Due ■ Last Payment Date ■ Last Payment Amount ■ Estimated Realization Amount ■ Deficiency Balance ■ Number of accounts associated with the collateral
Algorithm Entity	Case Status - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutoApprovalCheckforRepossession
Parameters	<p>Name: Auto Approval Rule Required (Yes/No): Yes Description: Auto Approval Rule</p> <p>Name: Approved Status Required (Yes/No): Yes Description: Approved Status</p> <p>Name: Task Type Required (Yes/No): Yes Description: Task Type</p> <p>Name: Queue</p>

	Required (Yes/No): Yes Description: Queue
Detailed Design	<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

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

Description	Automatic task creation for vendors
Detailed Description	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.
Algorithm Entity	Case Type - Enter Status

Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AutoTaskCreationForVendor
Parameters	<p>Name: Service Type Required (Yes/No): Yes Description: Service Type</p> <p>Name: Task Type Required (Yes/No): Yes Description: Task Type</p> <p>Name: Queue Required (Yes/No): Yes Description: Queue</p>
Detailed Design	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

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

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

Description	Automatic sending of Redemption letters
Detailed Description	<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"> 1. Contact Class - Contact class 2. Contact Type - Contact type 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 4. Validation Date - Possible values are SYSTEM DATE and POSTING DATE
Algorithm Entity	Case Status - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutomaticSendingofRedemptionLetters
Parameters	<p>Name: Contact Class Required (Yes/No): Yes Description: Contact Class</p> <p>Name: Contact Type Required (Yes/No): Yes Description: Contact Type</p> <p>Name: Primary Account Sw Required (Yes/No): No Description: Primary Account Sw</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	<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

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

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

	<p>Required (Yes/No): No Description: Dynamic Panel Two Fields</p> <p>Name: Dynamic Panel Three Required (Yes/No): No Description: Dynamic Panel Three</p> <p>Name: Dynamic Panel Three Fields Required (Yes/No): No Description: Dynamic Panel Three Fields</p> <p>Name: Dynamic Panel Four Required (Yes/No): No Description: Dynamic Panel Four</p> <p>Name: Dynamic Panel Four Fields Required (Yes/No): No Description: Dynamic Panel Four Fields</p>
Parameters(Contd.)	<p>Name: Dynamic Panel Five Required (Yes/No): No Description: Dynamic Panel Five</p> <p>Name: Dynamic Panel Five Fields Required (Yes/No): No Description: Dynamic Panel Five Fields</p> <p>Name: caseCharacteristics Required (Yes/No): No Description: Case Characteristics</p> <p>Name: Previous Status Transition Required (Yes/No): Yes Description: 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.

Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete
Parameters	Name: Next Status Required (Yes/No): Yes Description: Next Status
Detailed Design	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

Description	Send Repossession Alert to Vendor
Detailed Description	<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>
Algorithm Entity	Case Status - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionAssignmentAlert
Parameters	<p>Name: Contact Class Required (Yes/No): No Description: Contact Class</p> <p>Name: Contact Type Required (Yes/No): No Description: Contact Type</p> <p>Name: Service Type Required (Yes/No): No Description: Service Type</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>
Detailed Design	<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

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

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

	Description: Validate Date
Detailed Design	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

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

Table 5–134 Adhoc Characteristic Value Validation Algorithm PASTDATE_VAL

Description	Result Characteristic Value Date field Validation
Detailed Description	This algorithm is used to validate format enter by user for result characteristics during follow up. Validation Date: Validation Date will validate and compare the date with user provided date. 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>
Algorithm Entity	Characteristic Type - Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RedemptionClosureRedemptionClearDate
Parameters	<p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p> <p>Name: Stored Date Format Required (Yes/No): Yes Description: Stored Date Format</p> <p>Name: Date Format2 Required (Yes/No): No Description: Date Format2</p>
Detailed Design	<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. 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

Description	Redemption Clear Date Value Date field Calculation
Detailed Description	This algorithm is used to calculate the Redemption Clear Date. 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.
Algorithm Entity	Result Type - Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDateCal
Parameters	Name: Characteristic Type Code Required (Yes/No): No Description: Characteristic Type Code
Detailed Design	This algorithm is used to calculate the Redemption Clear Date. 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

Description	Update Review Date for associated accounts
Detailed Description	<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"> ■ Set Review Date - This will set the Review Date for the account ■ Remove Review Date - This will remove the Review date from the account <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"> ■ Yes: System will update existing account review date i.e. in case a review date is already present, system will override the same ■ 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

Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateReviewDate
Parameters	<p>Name: Override Flag Value Required (Yes/No): Yes Description: Override Flag Value</p> <p>Name: Days Offset Required (Yes/No): Yes Description: Days Offset</p> <p>Name: Update Type Required (Yes/No): Yes Description: Update Type</p>

Table 5–137 Case Monitoring CS-MONITOR

Description	Case Monitoring
Detailed Description	<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"> 1. Create a To Do, for a given To Do type. 2. Re-Allocate the case to a different Queue. 3. Set Prompt Days. 4. Transition to another Status.
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.CaseAutoTransition
Parameters	<p>Name: Next Status Required (Yes/No): No Description: Next Status</p> <p>Name: Work List Required (Yes/No): No Description: Work List</p> <p>Name: Reallocate Switch</p>

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

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

Description	Update warning indicator for the customer
Detailed Description	<p>This process will update the warning indicator for the customer</p> <ul style="list-style-type: none"> ■ Update activity i.e. set or remove the warning will also be defined as parameter to this process ■ Warning indicator to be set or removed will be set as parameter to this process ■ 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 <p>Call the service form host to update the warning indicator. Please give following values for the below parameters: Association Type : P (Primary) and A (Primary and Secondary) Update Type : S (Set) and R (Remove)</p>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateWarningIndicator
Parameters	<p>Name: Association Type Required (Yes/No): Yes Description: Override Association Type</p> <p>Name: Warning Indicator Type Required (Yes/No): Yes Description: Warning Indicator Type</p> <p>Name: Update Type Required (Yes/No): Yes Description: Update Type</p>

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

Description	Transition to Default Next Status
Detailed Description	<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"> 1. Next Status - The next status to which the case will be transitioned. 2. Next Transition Condition - Mention the transition condition for the next status.
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.TransitionDefaultNextStatus
Parameters	<p>Name: Next Status Required (Yes/No): No Description: Next Status</p> <p>Name: Next Transition Condition Required (Yes/No): No Description: Next Transition Condition</p>

Table 5–140 Set Account Warning Indicator C1-ACTCSWGD

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

5.47 Derived Field

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

Description	Timezone derivation field update algorithm
Detailed Description	This algorithm will update timezone of a person if it is blank

Algorithm Entity	Timezone derivcation Algorithm Spot
Program Type	Java (Converted)
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.TimeZoneDerivationAlgorithm
Parameters	NA
Detailed Design	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

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

Table 5–143 Validate Task Completion C1-VALTASKEKX

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

Description	Automatic Task Creation when case enters a particular status
Detailed Description	<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>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.tasks.algo.AutomaticTaskCreatiomn
Parameters	<p>Name: Task Type1 Required (Yes/No): Yes Description: Task Type</p> <p>Name: Queue1 Required (Yes/No): Yes Description: Queue</p> <p>Name: Task Type2 Required (Yes/No): No Description: Task Type</p> <p>Name: Queue2 Required (Yes/No): No Description: Queue</p> <p>Name: Task Type3 Required (Yes/No): No Description: Task Type</p> <p>Name: Queue3 Required (Yes/No): No Description: Queue</p> <p>Name: Task Type4 Required (Yes/No): No Description: Task Type</p> <p>Name: Queue4 Required (Yes/No): No Description: Queue</p>

Parameters (Cont.)	Name: Task Type5 Required (Yes/No): No Description: Task Type Name: Queue5 Required (Yes/No): No Description: Queue
Detailed Design	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

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

5.50 Legal Vendor Allocation C1-LGLVNDRAL

Table 5–146 Legal vendor Allocation C1-LGLVNDRAL

Descripti on	Legal vendor Allocation
Detailed Descripti on	Legal vendor Allocation
Algorithm Entity	Vendor Service Type Allocation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.LspVendorAllocati onAlgorithm
Paramete rs	NA

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

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

5.51 Extend Expiry Date C1-EXT-EXPDT

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

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

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

5.53 Create To Do C1-TO-DO

Table 5–150 Create To Do C1-TO-DO

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

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

5.54 Check customer eligibility C1-CHKCUST

Table 5–151 Check customer eligibility C1-CHKCUST

Description	Check customer eligibility
Detailed Description	<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"> ■ If ownership type parameter is set to "financial owner" <ul style="list-style-type: none"> • Get all financially responsible customers for the account. • For each customer, system should call the rule engine to check for customer eligibility. ■ If ownership type parameter is set to "primary" <ul style="list-style-type: none"> • System should call the rule engine to check for primary customers eligibility. <p>Customers' facts should be used for rule engine decision.</p> <p>For each call</p> <ul style="list-style-type: none"> ■ Rule will return output as "Validation Status". Possible values can be "Success" OR "Failure". ■ If validation status = Failed, process should return result as validation failed. <ul style="list-style-type: none"> • Check Validation failure option <ul style="list-style-type: none"> ○ Validation failure option = Fail case creation/transition. Case should not get created or should not transition status. ○ 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). ■ If validation status = Success, process should return result as validation successful. <p>Parameters:</p> <ul style="list-style-type: none"> ■ Ownership Type - Ownership type can be FINANCIAL_OWNER(Financial Owner) or

	<p>PRIMARY(Primary Owner).</p> <ul style="list-style-type: none"> ■ 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. ■ 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). ■ Validation Failure Transition Status - Case transition status in case of validation failure. ■ Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status. ■ Cancel Reason Char Value - Characteristic value for the defined characteristic type.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckCustomerEligibility
Parameters	<p>Name: Cancel Reason Char Value Required (Yes/No): No Description: Cancel Reason Char Value</p> <p>Name: Cancel Reason Char Type Required (Yes/No): No Description: Cancel Reason Char Type</p> <p>Name: Validation Failure Transition Status Required (Yes/No): No Description: Validation Failure Transition Status</p> <p>Name: Validation Failure Option Required (Yes/No): Yes Description: Validation Failure Option</p> <p>Name: Rule Id Required (Yes/No): Yes Description: Rule Id</p> <p>Name: Ownership Type Required (Yes/No): Yes Description: Ownership Type</p>

5.55 Capture Hardship Approval Date C1-HARAP-DT

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

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

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

5.57 Perform Validation for Collateral C1-VRFYCOL

Table 5–154 Perform Validation for Collateral C1-VRFYCOL

Description	Perform Validation for Collateral
Detailed Description	This algorithm type will perform below validations for the collateral with the case

Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification
Parameters	Name: CaseCategory Required (Yes/No): Yes Description: Case Category Code

5.58 Check target account eligibility C1-CHKTRGT

Table 5–155 Check target account eligibility C1-CHKTRGT

Description	Check target account eligibility
Detailed Description	<ul style="list-style-type: none"> ■ System should call the rule engine for eligibility check. Output of rule engine will be "Validation Status" <ul style="list-style-type: none"> • If validation status is "Success" <ul style="list-style-type: none"> ◦ Set set-off status as "Pending" ◦ Compute maximum amount allowed to Debit = Target account Balance - Minimum residual amount ◦ Clear the values in the "Exclude Target Till Date" and "Exclude Reason" fields, if populated • If validation status is "Fail" <ul style="list-style-type: none"> ◦ Set set-off status for target account as "Not eligible" ◦ "Exclude Reason" should be set as "Not Eligible" ◦ Get offset days for exclude reason from the look-up ◦ Set "Exclude target till" date for the target account to current business days + offset day. ◦ If no offset days are returned, "Exclude target till" date should not be updated ■ 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"> • 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)
Algorithm Entity	Case Type - Enter Status
Program Type	Java

Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckTargetAccountEligibility
Parameters	<p>Name: Cancel Reason Char Value Required (Yes/No): No Description: Cancel Reason Char Value</p> <p>Name: Cancel Reason Char Type Required (Yes/No): No Description: Cancel Reason Char Type Code</p> <p>Name: Validation Failure Transition Status Required (Yes/No): Yes Description: Validation Failure Transition Status Code</p> <p>Name: Minimum Residual Amount Required (Yes/No): Yes Description: Minimum Residual Amount</p> <p>Name: Rule Id Required (Yes/No): Yes Description: Rule Id</p>

5.59 Approval check for set-off transaction C1-ROSOAPPR

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

Description	Approval check for set-off transaction
Detailed Description	<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"> ■ Asset classification = Value set as parameter for the process ■ Accrual status = Value set as parameter for the process ■ Sum of Debit Amounts for all target accounts >= Specified threshold <p>Based on whether approval is required or not, transition the case to a status as set in the parameters of the process.</p>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RosoApprovalCheck
Parameters	Name: Total Debit Amount Threshold

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

5.60 Get target accounts C1-GETTRGT

Table 5–157 Get target accounts C1-GETTRGT

Description	Get target accounts
Detailed Description	<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"> ■ 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. ■ Same owners: Indicates that all owners of delinquent accounts are one the savings account / term deposit and there is no additional owner ■ 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). <p>Parameters:</p> <ul style="list-style-type: none"> ■ Validation Failure Transition Status - Case transition status in case of validation failure. ■ Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status. ■ Cancel Reason Char Value - Characteristic value for the defined characteristic type. ■ Casa Account Type Identifier List - Comma separated savings account (CASA)

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

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

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

Description	To Do Completion for case
Detailed Description	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.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.GenericToDoCompletionForCase
Parameters	<p>Name: Do Not Complete To Do Type Characteristic Type Required (Yes/No): No Description: Do Not Complete To Do Type Characteristic Type Code</p>

	Name: Do Not Complete To Do Type Characteristic Value Required (Yes/No): No Description: 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

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

5.64 Check Default Notice for Voluntary possession C1-CHKDFLT

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

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

Name	
Parameters	Name: Check Expiry Status Required (Yes/No): Yes Description: Check Expiry Status

5.65 Check Submission Date C1-CHKSUBDT2

Table 5–162 Check Submission Date C1-CHKSUBDT2

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

5.66 Update Financial Hardship flag C1-FNHRD-FLG

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

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

5.67 Result Type Case Transition Algorithm C1-RTCT

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

Description	Result Type Case Transition Algorithm
Detailed Description	<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"> ■ 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. ■ 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.
Algorithm Entity	Result Type – Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypeCaseTransitionAlgo
Parameters	<p>Name: Output Status Required (Yes/No): Yes Description: Output Status</p> <p>Name: Input Status Required (Yes/No): No Description: 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

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

Parameter s	Name: Case Category Required (Yes/No): Yes Description: Case Category
--------------------	--

5.69 Check Deceased status for the customer C1-CHKDCD

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

Descripti on	Check Deceased status for the customer
Detailed Descripti on	For the customer for whom the deceased case is being initiated check if, Deceased warning indicator is already set OR An active deceased case is present If either of above is true, case creation should fail.
Algorith m Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.CheckDeceas edStatusForCustomer
Paramet ers	Name: Case Category Required (Yes/No): Yes Description: Case Category

5.70 Associated accounts with deceased customer case C1-DCDACCTS

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

Descripti on	Associated accounts with deceased customer case
Detailed Descripti on	For the primary customer associated with the case. Get all accounts where this customer is primary owner and the accounts are in collections. Associated those accounts with the case.
Algorith m Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.AssociatedAcc WithDeceasedCustomer
Paramet ers	NA

5.71 Execute Fund Transfer C1-FUNDTRFR

Table 5–168 Execute Fund Transfer C1-FUNDTRFR

Description	Execute Fund Transfer
Detailed Description	<p>This process will execute the fund transfer. This should follow below steps for "each" target account where debit amount specified is > 0 and set-off status = "Pending"</p> <ul style="list-style-type: none"> ■ Execute a payment transfer transaction from Target account to the delinquent account. ■ If transaction is successful, set set-off status = "Success" for this target account ■ If transaction is not successful, set set-off status = "Fail" for this target account <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"> ■ If yes, transition the case to status as set in the parameter "Execution Success Status" ■ 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. <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>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ExecuteFundTransfer
Parameters	<p>Name: Execution Success Status Required (Yes/No): Yes Description: Execution Success Status Code</p> <p>Name: Execution Failure Status Required (Yes/No): Yes Description: Execution Failure Status</p> <p>Name: Cancel Reason Char Type</p>

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

Description	Algorithm to save previous state's status code.
Detailed Description	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.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.SavePreviousStatus
Parameters	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

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

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

5.74 Update Collection Warning Indicator C1-UPD-WRIND

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

Description	Update Collection Warning Indicator
Detailed Description	<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"> 1. Warning Indicator Type. 2. Warning Indicator Value 3. Rule Type Code 4. Collection Column To Be Updated 5. Set In Collections On Related Accounts 6. Exception Transition Condition
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateCollectionPartyWarningIndicator
Parameters	Name: Warning Indicator Type Required (Yes/No): Yes Description: Hardship Warning Indicator Type Name: Warning Indicator Value Required (Yes/No): Yes Description: Warning Indicator Value Name: Rule Type Code Required (Yes/No): No Description: Rule Type Code Name: Collection Column To Be Updated

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

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

5.76 Assign Applicable Relief Type C1-RELIF-TYP

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

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

Name	ApplicableReliefTypes
Parameters	Name: Rule Type Code Required (Yes/No): Yes Description: Rule Type Code

5.77 Create Customer Contact for Resultype Algo C1-CREATCC

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

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

Description	Calculate an expiry date when entering case status
Detailed Description	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
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.CalculateCaseStatusExpiryDate
Parameters	Name: Number of Days Required (Yes/No): Yes

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

5.79 Create Customer Contact C1-CUST-CONT

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

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

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

5.81 Validate Hardship Expiry Date. C1-VAL-FHEXP

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

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

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

5.82 Update Account in collections flag C1-ACTINCOL

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

Description	Update Account in collections flag
Detailed Description	<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"> ■ If no, set in-collections flag to "N" for the account ■ If yes. No updates should be done
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateAccountInCollectionFlag
Parameters	<p>Name: relationshipType</p> <p>Required (Yes/No): Yes</p> <p>Description: Relationship_Type</p> <p>Name: sourceHostId</p> <p>Required (Yes/No): Yes</p> <p>Description: 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

Description	Associate related entities with the case
Detailed Description	Associate related entities with the case
Algorithm Entity	Case Type – Enter Status

Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CaseAssociationForAssetRepossessionCase
Parameters	<p>Name: Customer Association Required (Yes/No): Yes Description: Customer Association</p> <p>Name: Account Association Required (Yes/No): Yes Description: Account Association</p>

5.84 Revalidate target account C1-REVALTRGT

Table 5–181 Revalidate target account. C1-REVALTRGT

Description	Revalidate target account
Detailed Description	<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"> ■ 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). ■ It is possible that target account balance got updated after user had entered the debit amounts. System should refresh balance from host. ■ Re-compute maximum amount which can be debited for each target account For each of the target account with set-off status as "Pending". ■ If maximum amount which can be debited is < Debit amount specified by the user then <ul style="list-style-type: none"> • Set set-off status and exclude reason as "Not eligible". • Skip rest of the processing and move to next target account ■ Call Rule engine to validate the account, which will output "Success" or "Failure". <ul style="list-style-type: none"> • If for any of the account validation status = "Failure", <ul style="list-style-type: none"> ◦ Set set-off status and exclude reason as "Not eligible". <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"> ■ There are no target accounts in pending status. Go to cancel set-off step ■ Sum of Debit amounts of target account > Overdue amount for delinquent account. In this case check the "Excess debit" option

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

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

5.85 Initiate LMI C1-INITLMIS

Table 5–182 Initiate LMI. C1-INITLMIS

Description	Initiate LMI
Detailed	Initiate LMI

Description	
Algorithm Entity	Case Type – Exit Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.Initiate_LMIProcess
Parameters	<p>Name: No LMI Option Required (Yes/No): Yes Description: No LMI Option</p> <p>Name: LMI Insurer Code Required (Yes/No): Yes Description: LMI Insurer Code</p> <p>Name: Initiate LMI Options Required (Yes/No): Yes Description: Initiate LMI Options</p> <p>Name: LMI Case Type Required (Yes/No): Yes Description: LMI Case Type</p> <p>Name: Balance Threshold Required (Yes/No): Yes Description: Balance Threshold</p>

5.86 NGP Collection case creation algorithm C1-COLLCASE

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

Description	NGP Collection case creation algorithm
Detailed Description	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.
Algorithm Entity	Collection Class Overdue Rules – Overdue Monitor Rule
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.CollectionCaseCreationOverdueMonitorRuleAlgo
Parameters	<p>Name: New Case Creation Rule Id Required (Yes/No): Yes</p>

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

5.87 Stop Contract Algorithm C1-CONTSTOP

Table 5–184 Stop Contract Algorithm. C1-CONTSTOP

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

5.88 Check for existing Hardship C1-CHKHRDSHP

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

Description	Check for existing Hardship
Detailed Description	<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, It will proceed with case creation.</p> <p>This checks for an existing Hardship application for the party.</p>
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckExistingHardship
Parameters	<p>Name: Hardship Case Type Feature Config</p> <p>Required (Yes/No): Yes</p> <p>Description: Hardship Case Type Feature Config</p>

5.89 Algorithm for contact processing C1-CNTCT

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

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

5.90 Check application expiry date C1-CHKEXP

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

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

5.91 New Customer Contact Creation Algorithm C1-CCCREATE

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

Description	New Customer Contact Creation Algorithm
Detailed Description	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.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.correspondence.CustomerContactCreation
Parameters	Name: Customer Contact Class

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

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

5.93 Transition case on a date on a case characteristic C1-TRANS DT

Table 5–190 Transition case on a date on a case characteristic. C1-TRANS DT

Description	Transition case on a date on a case characteristic
Detailed Description	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
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.TransitionToNextStatusOnDate
Parameters	<p>Name: Characteristic Type Required (Yes/No): Yes</p>

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

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

5.95 Mark accounts for strategy review C1-REVIW-ACT

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

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

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

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

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

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

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

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

Parameters	NA
-------------------	----

5.98 Check for Operational Relief Types C1-OP-RT

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

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

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

5.99 Auto-Approval Check C1-FH-AUTOAP

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

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

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

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

5.101 Update Party Warning Indicator C1-UPD-PRTWI

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

Description	Update Party Warning Indicator
Detailed Description	<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>
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdatePartyWarningIndicator
Parameters	<p>Name: Warning Indicator Type Required (Yes/No): Yes Description: Warning Indicator Type</p> <p>Name: Warning Indicator Value Required (Yes/No): Yes Description: Warning Indicator Value</p> <p>Name: Rule Type Code Required (Yes/No): No Description: Rule Type Code</p> <p>Name: Exception Transition Condition Required (Yes/No): No Description: 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

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

5.103 Collection - Entity Activity Population. C1-ENTACTPOP

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

Description	Collection - Entity Activity Population
Detailed Description	<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"> 1. Entity Type : Person/Account for which activity is getting created (e.g. Case can be created on Person as well as Account) 2. Entity Id : Person/Account Id 3. ModeOfOperation: Add/Update/Delete/Cancel 4. HostEntityId: Activity Entity Id (e.g PTP/CC/Follow-up/Case Id) 5. HostEntityName: PTP/CC/FOLLOWUP/CASE
Algorithm Entity	Installation – Entity Activity Populate
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.PopulateAccountActivityAlgo
Parameters	NA

5.104 Cancel Hardship Application C1-CXLFH

Table 5–201 Cancel Hardship Application. C1-CXLFH

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

5.105 Perform Payment Transfer for ROSO C1-ROSOPMTXR

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

Description	Perform Payment Transfer for ROSO
Detailed Description	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).
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.rightofSetOff.PerformPaymentXferForROSO
Parameters	<p>Name : Exception Transition Condition Required (Yes/No): Yes Description: Exception Transition Condition</p> <p>Name: Host Code Required (Yes/No): Yes Description: Host Code</p> <p>Name : Failed Leg Notification To Do Type Required (Yes/No): Yes Description: 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

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

5.107 Create RMB Entities from Host Data C1-VCREATE

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

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

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

5.108 Populate Activity Table For Notes Creation C1-NTACTIVITY

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

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

5.109 Suspend Activity for Account Pre Processing C1-SPATACPRE

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

Description	Suspend Activity for Account Pre Processing
Detailed Description	Suspend Activity for Account Pre Processing
Algorithm Entity	Business Object – Pre-Processing

Program Type	Java
Program Name	com.splwg.ccb.domain.collection.suspendActivity.SuspendActivityPreProcessing_Impl
Parameters	NA

5.110 Sample TAM Algorithm Type C1-TAMALG

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

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

5.111 Cancel Approval Request C1-CANAPPR

Table 5–208 Cancel Approval Request. C1-CANAPPR

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

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

5.112 Update Customer in collections flag C1-CUSINCOL

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

Description	Update Customer in collections flag
Detailed Description	<p>This algorithm will set or reset in-collections flag for the customer in core banking While setting the Flag</p> <ul style="list-style-type: none"> ■ Skip the process if flag is already set ■ If not, set the in-collection flag for the customer to "Y" in the host. <p>While resetting the Flag,</p> <ul style="list-style-type: none"> ■ Skip the process if flag is already reset ■ System should check for all cases to which the customer is associated (as primary or secondary entity). ■ 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. ■ If none of the cases for the customer are from those categories, set the in-collection
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateCustomerInColl Flg
Parameters	<p>Name: Update Type Required (Yes/No): Yes Description: Update Type</p> <p>Name: Case Category1 Required (Yes/No): No Description: Case Category1 Code</p> <p>Name: Case Category2 Required (Yes/No): No Description: Case Category2 Code</p>

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

5.113 Set Display Date C1-SETDSPDT

Table 5–210 Set Display Date. C1-SETDSPDT

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

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

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

Type	
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.DefaultNextStatusAutoTransition
Parameters	Name: Wait Days Required (Yes/No): Yes Description: 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

Description	Check current cases on account for exclusion
Detailed Description	<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"> ■ Get all active cases for the account. Account can be primary or secondary entity for that case. ■ Get case categories for all these cases ■ If the case category for any of the cases is from the exclusion list, validation should fail. ■ Check Validation failure option <ul style="list-style-type: none"> • Validation failure option = Fail case creation/transition. Case should not get created or should not transition status • 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) ■ 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. <p>Parameters:</p> <ul style="list-style-type: none"> ■ Cancel Reason Char Type: Characteristic type to set as case characteristic if validation failure option is transition status. ■ Cancel Reason Char Value: Characteristic value for the defined characteristic type. ■ Validation Failure Transition Status: Case transition status in case of validation failure. ■ 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).
Algorithm Entity	Case Type – Enter Status
Program	Java

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

Description	Update Collateral Status in the host
Detailed Description	Update Collateral Status in the hostdate of entry into current status is >= specified number of days.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost
Parameters	<p>Name: To Do Type Required (Yes/No): Yes Description: To Do Type</p> <p>Name: Collateral Status Required (Yes/No): Yes Description: Collateral Status</p>

5.117 Initiate collateral valuation C1-COLLVAL

Table 5–214 Initiate collateral valuation. C1-COLLVAL

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

Description	Mandatory characteristics check for Asset Repo
Detailed Description	Mandatory characteristics check for Asset Repo
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics
Parameters	<p>Name: Reference Characteristic Value Required (Yes/No): Yes</p>

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

5.119 Update Collateral Status in the Host C1-UPCOLLSTS

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

Description	Update Collateral Status in the Host
Detailed Description	Update Collateral Status in the Host
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost
Parameters	<p>Name: To Do Type Required (Yes/No): Yes Description: To Do Type</p>

	<p>Name: Collateral Status</p> <p>Required (Yes/No): Yes</p> <p>Description: Collateral Status</p>
--	---

5.120 Set exclusion date for delinquent account. C1-EXCLROSO

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

Description	Set exclusion date for delinquent account
Detailed Description	<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 <= 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"> ■ Get the value for the specified char type ■ This char type should be used to get the offset days from the Lookup for set-off exclusion days ■ Set-off exclusion date should be set as current business days + offset days. ■ If mapping for the reason is not found, default value for offset days should be used. <p>If Cancel Reason char type parameters is blank but Reason code is provided</p> <ul style="list-style-type: none"> ■ Get the corresponding offset days from the lookup for the Reason code ■ Set-off exclusion date should be set as current business days + offset days ■ If mapping for the reason is not found, default value for offset days should be used. <p>Parameters:</p> <ul style="list-style-type: none"> ■ Default Offset - Number of days to add to the set-off exclusion date. ■ Reason Code - Code to fetch offset days from lookup. ■ Cancel Reason - Characteristic type code to fetch offset days from lookup.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.UpdateSetoffExclusionDate
Parameters	<p>Name: Default Offset</p> <p>Required (Yes/No): Yes</p> <p>Description: Default Offset</p>

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

5.121 Cancel Set-off C1-CANROSO

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

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

5.122 Complete Set-off C1-COMPROSO

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

Description	Complete Set-off
Detailed Description	<p>This algorithm transitions the case to complete. Processing Logic will be as below</p> <ul style="list-style-type: none"> ■ Validate that at least one of the target account has set-off status = "Success" and Reversed Flag = "N". ■ If above validation fails transition to complete should not be allowed and To-do of given To-do Type should be created.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CompleteSetoff
Parameters	NA

5.123 Reverse Set-off C1-REVROSO

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

Description	Reverse Set-off
Detailed Description	<p>This algorithm transitions the case to Reversed status. Processing Logic will be as below</p> <ul style="list-style-type: none"> ■ Validate below for each target account <ul style="list-style-type: none"> • Set-off status is not "Success" or If set-off status is "Success" then Reversed Flag should be "Y". • There should be at least one account with Reversed Flag as "Y". ■ If above validation fails, transition to this status should not be allowed.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ReverseSetoff
Parameters	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

Description	Algorithm type for update case id for Insurance
Detailed Description	Algorithm type for update case id for Insurance
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateInsuranceCaseDetails
Parameters	NA

5.125 Case Creation on enter processing C1-CCOENTER

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

Description	Case Creation on enter processing
Detailed	This Algorithm will create a new case for the given Case Type on enter processing.

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

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

5.127 Action category Validation algorithm C1-ACTCAT

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

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

5.128 Action Type Algorithm Type C1-ACTTYP

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

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

5.129 Case Type Status Mapping Algorithm Type C1-CASETYMP

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

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

5.130 Collection - Close Processing Algorithm C1-CCALG

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

Description	Collection - Close Processing Algorithm
Detailed Description	<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"> ■ It will update the financial balance of the Contract to zero through an adjustment.

	<ul style="list-style-type: none"> ■ 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 ■ 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 <p>The following parameters are available and are required:</p> <ul style="list-style-type: none"> ■ Adjustment Type used for the adjustment created by this algorithm. ■ Cancellation Reason Code used while canceling Active PTPs ■ 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. ■ 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.
Algorithm Entity	Collection – Collection Case Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.CollectionClosingAlgo
Parameters	<p>Name: Adjustment Type Required (Yes/No): Yes Description: Adjustment Type</p> <p>Name: Cancellation Reason Code for PTP Required (Yes/No): Yes Description: Cancellation Reason Code for PTP</p> <p>Name: Is Closing Required Flag (Y/N) Required (Yes/No): Yes Description: Is Closing Required Flag</p> <p>Name: Final Default Case Status Required (Yes/No): Yes Description: 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

Description	Algorithm type for case list update
--------------------	-------------------------------------

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

5.132 Copy Characteristics Algorithm C1-COPYCHAR

Table 5–229 Copy Characteristics Algorithm. C1-COPYCHAR

Description	Copy Characteristics Algorithm
Detailed Description	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".
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CopyCharacteristicsOnCaseCreate
Parameters	<p>Name: Case Category Required (Yes/No): Yes Description: Case Category</p> <p>Name: Characteristics List Required (Yes/No): No Description: Characteristics List</p>

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

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

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

	<ul style="list-style-type: none"> • 'Permission to Call' is Yes And • Current Time is within the State level Acceptable Time Limits And • Current Time is within the preferred times of the Customer And • Current Date is not within the Do Not Disturb Dates <p>Else it is 'Red'.</p>
Algorithm Entity	Installation – Contact Information
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.collectionLandingPage.ContactInformationCallAdviceAlgo
Parameters	NA

5.134 Task Case Mapping Validation Algorithm C1-TCVAL

Table 5–231 Task Case Mapping Validation Algorithm. C1-TCVAL

Description	Task Case Mapping Validation Algorithm
Detailed Description	<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>
Algorithm Entity	Task Type Case Type Validation Algorithm Spot
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepoDateValidation
Parameters	<p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p> <p>Name: Case Type Code Required (Yes/No): Yes Description: Case Type Code</p> <p>Name: Task Type Code Required (Yes/No): Yes Description: 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

Description	Monitoring Algorithm For Dispute Resolved to exit from Dispute Status.
Detailed Description	<p>This Monitoring Algorithm exit the Dispute status and move into Contact Status 'x' days after the resolution date. Also it set Dispute Flag at account level to N when case exits this status. Below are the soft parameter example:</p> <ul style="list-style-type: none"> ■ Validation Date: This Validation Date will validate and compare the date with New Dispute Resolved Date (Date obtained after adding no. of grace days). It's value can be SYSTEM DATE or POSTING DATE. This is mandatory parameter. ■ 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. This is mandatory parameter ■ 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. ■ Contact RM Status: Case will move to Contact Relation Manager status if relation manager exist for that account and Contact Rm status exist ■ 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 ■ Characteristic Code : Characteristic Type Code to be referred while registering dispute resolution follow up to set new Risk Indicator on Recovery Account. ■ Characteristic Value : Characteristic Value Code to be referred while registering dispute resolution follow up to set new Risk Indicator on Recovery Account. ■ Risk Indicator Code : Sets the Risk Indicator Code on Recovery Account based on Characteristic Code and Characteristic Value mentioned while Dispute Resolution.
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateDisputeMonitor
Parameters	<p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p>

	<p>Name: No Of Grace Days Required (Yes/No): Yes Description: No Of Grace Days</p> <p>Name: Contact Status Required (Yes/No): No Description: Contact Status</p> <p>Name: Contact RM Status Required (Yes/No): No Description: Contact RM Status</p> <p>Name: Contact Alternate Status Required (Yes/No): No Description: Contact Alternate Status</p> <p>Name: Characteristic Code Required (Yes/No): No Description: Characteristic Code</p> <p>Name: Characteristic Value Required (Yes/No): No Description: Characteristic Value</p> <p>Name: Risk Indicator Code Required (Yes/No): No Description: Risk Indicator Code</p>
--	---

5.136 SLA Parameters validation algorithm C1-SLAPARAM

Table 5–233 SLA Parameters validation algorithm. C1-SLAPARAM

Description	SLA Parameters validation algorithm
Detailed Description	SLA Parameters validation algorithm created for Recovery 2.6.2 release. This algorithm to be called along with C1-SLABO.
Algorithm Entity	Business Object – Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.vendor.SLAParametersPostProcessAlgo
Parameters	NA

5.137 Case Group add validation algorithm C1-CGVAL

Table 5–234 Case Group add validation algorithm. C1-CGVAL

Description	Case Group add validation algorithm
Detailed Description	Case Group add validation algorithm
Algorithm Entity	Business Object – Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseGroup.CaseGroupValidationAlgorithm
Parameters	NA

5.138 Collection - Get Strategy Algorithm C1-COLGS

Table 5–235 Collection - Get Strategy Algorithm. C1-COLGS

Description	Collection - Get Strategy Algorithm
Detailed Description	<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"> ■ Rule Type (defined in the input parameter) ■ Case Type (if any) ■ Days Past Due ■ Overdue Amount ■ Collection Type
Algorithm Entity	Collection – Get Strategy Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.GetStrategyAlgo
Parameters	<p>Name: Rule Type</p> <p>Required (Yes/No): Yes</p> <p>Description: 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

Description	Create/Move Collection Strategy Cases for Account
Detailed Description	<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"> ■ 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. ■ Is Closing required - Flag indicate whether case closing is required or not (Y/N) ■ Collection Closing Algorithm - This is the algorithm code for Collection - Close Processing Algorithm ■ Get Strategy Algorithm - This is the algorithm code for Collection - Get Strategy Algorithm
Algorithm Entity	Collection Class Overdue Rules– Overdue Monitor Rule
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.CaseOverdueMonitorRuleAlgo
Parameters	<p>Name: Final Default Case Status Required (Yes/No): Yes Description: Final Default Case Status</p> <p>Name: Is Closing Required Flag (Y/N) Required (Yes/No): Yes Description: Is Closing Required Flag (Y/N)</p> <p>Name: Collection Closing Algorithm Code Required (Yes/No): Yes Description: Collection Closing Algorithm Code</p> <p>Name: Get Strategy Algorithm Code Required (Yes/No): Yes Description: 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

Description	Collection - Case SA Update for Manual Creation
Detailed Description	This Algorithm will update Case SA table for Manual Case Creation
Algorithm Entity	Business Object – Audit
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.CaseSaUpdateBoAuditAlgo
Parameters	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

Description	Promise to Pay - Additional Grace Days Sample Algo
Detailed Description	<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>
Algorithm Entity	Installation – Additional Grace Days
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.payPlan.AdditionalGraceDaysCalculationAlgorithm
Parameters	<p>Name: Additional Grace Days</p> <p>Required (Yes/No): No</p> <p>Description: Additional Grace Days</p>

5.142 Promise to Pay Threshold Percentage C1-PPTHRESH

Table 5–239 Promise to Pay Threshold Percentage. C1-PPTHRESH

Description	Promise to Pay Threshold Percentage
Detailed Description	<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"> ■ Promise to Pay ID ■ Total Amount Paid towards the promise to pay ■ Date (Business Date - Grace Days)

	<ul style="list-style-type: none"> Array of Promise to Pay Scheduled Payments balance <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>
Algorithm Entity	Installation – Payment Threshold Percentage
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.payPlan.PaymentThresholdPercentageCalculationAlgorithm
Parameters	<p>Name: Threshold Percentage</p> <p>Required (Yes/No): No</p> <p>Description: 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

Description	Result type Post Processing Case Transition Algo.
Detailed Description	<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"> 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 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.
Algorithm Entity	Result Type - Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcCaseTransAlgo
Parameters	<p>Name: Output Status</p> <p>Required (Yes/No): Yes</p>

	<p>Description: Output Status</p> <p>Name: Input Status</p> <p>Required (Yes/No): No</p> <p>Description: Input Status</p>
--	---

5.144 Inbound Customer algorithm C1-IN-CUST

Table 5–241 Inbound Customer algorithm. C1-IN-CUST

Description	Inbound Customer algorithm
Detailed Description	<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"> ■ Retrieve the XML message containing the customer information, which stored on the FACT MO. ■ Read the XML and determine if the action is to add a new customer or update an existing customer. ■ It may create a combination of Person, Account, Contract, Contract, or Adjustment, depending on what was contained in the XML. ■ If all objects are created successfully it will transition the lifecycle to the 'Created' status ■ Else if any of the objects experienced an error while processing it will transition the lifecycle to the "Rejected" status. <p>It has two parameters , both optional.</p> <ul style="list-style-type: none"> ■ Account Id Type identifies the Account Identifier Type used to locate the account in ORMB. ■ Person Id Type identifies the Person Identifier Type used to locate the person in ORMB.
Algorithm Entity	Business Object Status - Enter
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.inboundCustomer.CreateEntityAlgo
Parameters	<p>Name: Account Id Type Required (Yes/No): No Description: Account Id Type</p> <p>Name: Person Id Type Required (Yes/No): No Description: 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

Description	Result Type Pre-processing Algorithm Type
Detailed Description	Result Type Pre-processing Algorithm Type
Algorithm Entity	Result Type – Pre Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePreProcAlgo
Parameters	NA

5.146 Result Type Post-processing Algorithm Type C1-RSTPOST

Table 5–243 Result Type Post-processing Algorithm Type. C1-RSTPOST

Description	Result Type Post-processing Algorithm Type
Detailed Description	Result Type Post-processing Algorithm Type
Algorithm Entity	Result Type – Pre Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcAlgo
Parameters	<p>Name: contactType Required (Yes/No): Yes Description: Contact Type</p> <p>Name: contactClass Required (Yes/No): Yes Description: Contact Class</p> <p>Name: contactMethod Required (Yes/No): Yes Description: Contact Method</p>

5.147 Characteristic Type :Validate Date Field (Custom) C1-ADHDATE

Table 5–244 Characteristic Type :Validate Date Field (Custom). C1-ADHDATE

Description	Characteristic Type :Validate Date Field (Custom)
--------------------	---

Detailed Description	<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>
Algorithm Entity	Characteristic Type – Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CustomAdhocDateValidationAlgComp
Parameters	NA

5.148 Algorithm Type for Dialer Results Upload C1-DLRRSUPLD

Table 5–245 Algorithm Type for Dialer Results Upload. C1-DLRRSUPLD

Description	Algorithm Type for Dialer Results Upload
Detailed Description	Algorithm Type for Dialer Results Upload
Algorithm Entity	Dialer Result Upload – Account, Customer, CaseType
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.dialerResultUpload.DialerResultUploadAlgo
Parameters	NA

5.149 Algorithm for Hardship case creation activity C1-CRTHDSP

Table 5–246 Algorithm for Hardship case creation activity. C1-CRTHDSP

Description	Algorithm for Hardship case creation activity
Detailed	This Algorithm is responsible for making a Hardship Case entry on the Party, when the

Description	Hardship case is created.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCaseCreationActivity
Parameters	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

Description	This Algorithm is used to abort Approval work item
Detailed Description	<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"> ■ caseStatusExclusionList: Comma separated list of case status for which approval work item shouldn't be aborted. ■ Composite Name: Fully qualified approval class name. ■ Instance Title: Approval instance work item title prefix. <ul style="list-style-type: none"> • Example: Input parameters and it's applicable value for ROSO Process, ■ Composite Name: com.ofss.fc.workflow.process.ROSOProcessForApproval ■ Instance Title: ROSO_CASE_Value of the above parameters is dependent upon the SOA approval work flow.
Algorithm Entity	Case Type – Exit Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AbortApprovalWorkItemsAlgo
Parameters	<p>Name: Composite Name Required (Yes/No): Yes Description: Composite Name</p> <p>Name: Instance Title Required (Yes/No): Yes Description: Instance Title</p>

	Name: Case Status Exclusion List Required (Yes/No): No Description: 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

Description	Cancel Process Approval Request:Financial Hardship
Detailed Description	<p>This algorithm will cancel all pending approval requests for the case. Example for parameter values for hardship Process:</p> <ul style="list-style-type: none"> ■ Composite Name: com.ofss.fc.workflow.process.FinancialHardshipProcessForApproval ■ Instance Title: FINANCIAL_HARDSHIP_CASE_ <p>Value of the above parameters depends upon the SOA approval work flow.</p>
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelFinancialHardshipApprovalReqAlgo
Parameters	<p>Name: Composite Name Required (Yes/No): Yes Description: Composite Name</p> <p>Name: Instance Title Required (Yes/No): Yes Description: Instance Title</p>

5.152 Hardship Characteristic Association C1-FHCHARASC

Table 5–249 Hardship Characteristic Association. C1-FHCHARASC

Description	Hardship Characteristic Association
Detailed Description	Hardship Characteristic Association

Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCharAssociation
Parameters	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

Description	Pre-Populated system facts to be used for Rule
Detailed Description	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. 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).
Algorithm Entity	Rule Execution – Pre Populated Rule Facts
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.PrePopulatedSystemFacts
Parameters	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

Description	Assign Batch level TODOs(task) to a queue.
Detailed Description	Assign Batch level TODOs(task) to a queue.
Algorithm Entity	To Do Type – To Do Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.AssignTaskToQueueAlgorithm
Parameters	NA

5.155 Validate Date Field :Custom C1-ADHV-DTD

Table 5–252 Validate Date Field :Custom. C1-ADHV-DTD

Descripti	Validate Date Field :Custom
------------------	-----------------------------

on	
Detailed Description	<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>
Algorithm Entity	Characteristic Type – Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.ProductAdhocDateValidationAlgComp
Parameters	<p>Name: From Date Required (Yes/No): No Description: From Date</p> <p>Name: To Date Required (Yes/No): No Description: To Date</p> <p>Name: Date Format1 (Stored Format) Required (Yes/No): Yes Description: Date Format1 (Stored Format)</p> <p>Name: Date Format2 Required (Yes/No): No Description: Date Format2</p> <p>Name: Date Format3 Required (Yes/No): No Description: Date Format3</p> <p>Name: Date Format4 Required (Yes/No): No Description: Date Format4</p>

	<p>Name: Date Format5 Required (Yes/No): No Description: Date Format5</p> <p>Name: Date Format6 Required (Yes/No): No Description: Date Format6</p>
--	---

5.156 Characteristic Date field Validation C1-CHARDTVAl

Table 5–253 Characteristic Date field Validation. C1-CHARDTVAl

Description	Characteristic Date field Validation
Detailed Description	<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>
Algorithm Entity	Characteristic Type – Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.CharAdhocDateValidation
Parameters	<p>Name: From Date Required (Yes/No): No Description: From Date</p> <p>Name: To Date Required (Yes/No): No Description: To Date</p> <p>Name: Date Format1 (Stored Format)</p>

	<p>Required (Yes/No): Yes Description: Date Format1 (Stored Format)</p> <p>Name: Date Format2 Required (Yes/No): No Description: Date Format2</p> <p>Name: Date Format3 Required (Yes/No): No Description: Date Format3</p> <p>Name: Date Format4 Required (Yes/No): No Description: Date Format4</p> <p>Name: Date Format5 Required (Yes/No): No Description: Date Format5</p> <p>Name: Date Format6 Required (Yes/No): No Description: Date Format6</p> <p>Name: Business Date Validation Required Required (Yes/No): No Description: Business Date Validation Required</p>
--	--

5.157 Retry Case in Error C1-RCASEE

Table 5–254 Retry Case in Error C1-RCASEE

Description	Retry Case in Error
Detailed Description	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.
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.RetryCaseInErrorForHardshipApp
Parameters	Name: Retry Case Status Code Required (Yes/No): No

	Description: Retry Case Status Code Name: Max Retries Required (Yes/No): No Description: Max Retries
--	---

5.158 Allocate Queue for Customer Level Case C1-ALLOCQUE

Table 5–255 Allocate Queue for Customer Level Case. C1-ALLOCQUE

Description	Allocate Queue for Customer Level Case
Detailed Description	Allocate Queue for Customer Level Case. Only Queue Allocation would be done. User Allocation would be skipped for customer level cases.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.PerformQueueAllocation
Parameters	Name: Queue Code Required (Yes/No): Yes Description: Queue Code

5.159 Person Address – Collection C1-PERADDRC

Table 5–256 Person Address – Collection. C1-PERADDRC

Description	Person Address – Collection
Detailed Description	This Algorithm is a reference implementation for consulting. This algorithm will be used for validating Person address as per requirement.
Algorithm Entity	Business Object – Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.PersonCollectionAddressValidation
Parameters	NA

5.160 Person Contact Point Update - Post Processing C1-PERCONTPP

Table 5–257 Person Contact Point Update - Post Processing. C1-PERCONTPP

Description	Person Contact Point Update - Post Processing
Detailed Description	This is a reference implementation of Post processing Algo. Customization team can utilize this hook
Algorithm Entity	Collection Contact Preference – Post Process
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot
Parameters	NA

5.161 Update Self Serve Flag Algorithm C1-SELFERVE

Table 5–258 Update Self Serve Flag Algorithm. C1-SELFERVE

Description	Update Self Serve Flag Algorithm
Detailed Description	Action -soft parameter mentioned in algorithm type which will update the self_ serve flag to Y or N. If Action = Set make Self Serve Flag = Y If Action = Reset make Self Serve Flag = N
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.UpdateSelfServeFlag
Parameters	Name: Action Required (Yes/No): Yes Description: 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

Description	Create Task for Self Serve Request for Assistance transaction
Detailed Description	This algorithm will be used to Create Task post Follow Up.
Algorithm Entity	Result Type – Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.actionObject.actionHistory.FollowUpResultTaskAlgo

Parameters	Name: Task For Required (Yes/No): Yes Description: Task For
	Name: Task Type Required (Yes/No): Yes Description: Task Type
	Name: Task Queue Required (Yes/No): Yes Description: Task Queue

5.163 Transition to Under Settlement Offer Status

Table 5–260 Transition to Under Settlement Offer Status. C1-TSOF

Description	Transition to Under Settlement Offer Status if Active Settlement Offer Flag is Yes
Detailed Description	Transition the case to Under Settlement Offer Status if Active Settlement Offer Flag is Yes. Parameters: <ol style="list-style-type: none"> Under Settlement Offer Status: Value should be "Under Settlement Offer Status Code". Case will be transitioned to this Status. Reallocate Switch : Value could be "Y" on "N"
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.TransitionToUnderSettlement Offer
Parameters	Name: Under Settlement Offer Status Required (Yes/No): No Description: Under Settlement Offer Status Name: Reallocate Switch Required (Yes/No): No Description: Reallocate Switch

5.164 Create PTP Validation Algorithm

Table 5–261 Create PTP Validation Algorithm. C1-VPTP

Description	Create PTP Validation Algorithm
--------------------	---------------------------------

Detailed Description	This algorithm performs below validations before creating PTP: 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.
Algorithm Entity	Case Type – Enter Status Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.ValidatePTPForSettlementOffer
Parameters	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

Description	Under Settlement Offer Status Monitor Algorithm for Settlement Expiry/Offer Acceptance/Rejection
Detailed Description	<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 > 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>
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.recovery.internalLifecycle.algorithms.MonitorSettlementOffer
Parameters	<p>Name: Task Type</p> <p>Required (Yes/No): No</p> <p>Description: Task Type</p> <p>Name: Task Queue</p>

	<p>Required (Yes/No): No Description: Task Queue</p> <p>Name: Contact Status Required (Yes/No): No Description: Contact Status</p> <p>Name: Contact Alt Status Required (Yes/No): No Description: Contact Alt Status</p> <p>Name: Reallocate Switch Required (Yes/No): No Description: Reallocate Switch</p> <p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>
--	---

5.166 Algorithm to identify Partial Charge off Accounts

Table 5–263 Algorithm to identify Partial Charge off Accounts BATCHJOBPARM

Description	Algorithm to identify Partial Charge off Accounts
Detailed Description	<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"> 1. Host - Host Code 2. Collateral Type - Collateral Type Codes to be Considered for Assessment 3. Product Group Code - Account Product Group Codes to be Considered for Assessment 4. Bankruptcy Risk Indicator Code - Bankruptcy Risk Indicator Code to be Considered for Party 5. Deceased Risk Indicator Code - Deceased Risk Indicator Code to be Considered for Party
Algorithm Entity	Generic Batch Job Parameters Algorithm Spot
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.partialChargeOffBatchJobParamsNGP
Parameters	Name: Host

	<p>Required (Yes/No): Yes Description: Host</p> <p>Name: Collateral Type Required (Yes/No): Yes Description: Collateral Type</p> <p>Name: Product Group Code Required (Yes/No): Yes Description: Product Group Code</p> <p>Name: Bankruptcy Risk Indicator Code Required (Yes/No): Yes Description: Bankruptcy Risk Indicator Code</p> <p>Name: Deceased Risk Indicator Code Required (Yes/No): Yes Description: Deceased Risk Indicator Code</p>
--	---

5.167 Monitor Settlement Offer Payments

Table 5–264 Monitor Settlement Offer Payments C1-MSOP

Description	Monitor Settlement Offer Payments
Detailed Description	<p>If Active Settlement Offer Flag = Y AND Settlement Offer Payment Total < Settlement Amount, then check:</p> <p>If System date > 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>
Algorithm Entity	Case Type – Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.recovery.externalLifecycle.algorithms.MonitorSettlementOfferPaymentsAlgorithm
Parameters	<p>Name: Validation Date Required (Yes/No): No Description: Validation Date</p>

6 Localized Algorithms

6.1 Localized Algorithms

Table 6–1 Case Transition for Active Service Member C1-ACTMEMCHK

Description	Case Transition for Active Service Member
Detailed Description	<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>
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.scra.algorithm.ActiveServiceAlgorithm
Parameters	<p>Name: Suspend Status Required (Yes/No): No Description: Suspend Status</p> <p>Name: All Financial Owner Validation Required (Yes/No): Yes Description: All Financial Owner Validation</p> <p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p> <p>Name: Dependent Validation Required (Yes/No): Yes Description: Dependent Validation</p> <p>Name: Suspend Reason Characteristics Required (Yes/No): No Description: Suspend Reason Characteristics</p>
Detailed Design	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

Description	Block Repossession - Enter Status
Detailed Description	Verify if repossession needs to be blocked as per SCRA regulations
Algorithm Entity	Case Status - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ActiveMilitaryServiceCheckonAssociatedCustomers
Parameters	<p>Name: Validation Date Required (Yes/No): Yes Description: Validation Date</p> <p>Name: Repossession Block Period Required (Yes/No): No Description: Repossession Block Period</p>
Detailed Design	Verify if repossession needs to be blocked as per SCRA regulations

Table 6–3 Metro 2 Reporting - Account Status Code post Liquidation C1- ASCLIQU

Description	Metro 2 Reporting - Account Status Code post Liquidation
Detailed Description	<p>If Repossession Reason = Voluntary Surrender If Account Status Condition = Consumer not responsible for Remaining Balance/ No Deficiency Balance Set Account Status Code = 95;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance' Set Account Status Code = 95;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full Set Account Status Code = 61; 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' Set Account Status Code = 96;</p> <p>If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full Set Account Status Code = 63;</p> <p>Additionally record the Repossession Date and the Last Payment Date in each of the scenarios. Data to be logged: (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. Char Values are: CNRBND,CRBAP,CRRB</p>
Algorithm Entity	Result Type – Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithm s. Metro2AcctStatusCodePostLiquidationPostProcessing
Parameters	<p>Name: Account status Code Char Required (Yes/No): Yes Description: Account status Code Char</p> <p>Name: Voluntary Surrender Code Required (Yes/No): Yes Description: Voluntary Surrender Code</p>
Detailed Design	Metro 2 Reporting - Account Status Code post Liquidation

Table 6–4 Metro 2 Reporting - Account Status Code C1- ASCREPO

Descri ption	Metro 2 Reporting - Account Status Code
Detail ed Descri ption	<p>If Repossession Reason = Voluntary Surrender Set Account Status Code = Account Status Code for Voluntary Surrender Special Comment Code = Special Comment Code for Voluntary Surrender Else Set Account Status Code = Account Status Code for Normal Repossession Special Comment Code = Special Comment Code for Normal Repossession (If multiple accounts associated with the case, the Account Status Code should be set for all</p>

	associated accounts)
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatusCodeEnterProcessingAlgo
Parameters	<p>Name: Special Comment Code for Normal Repossession Required (Yes/No): No Description: Special Comment Code for Normal Repossession</p> <p>Name: Special Comment Code for Voluntary Surrender Required (Yes/No): No Description: Special Comment Code for Voluntary Surrender</p> <p>Name: Voluntary Surrender Code Required (Yes/No): Yes Description: Voluntary Surrender Code</p> <p>Name: Account Status Code for Normal Repossession Required (Yes/No): Yes Description: Account Status Code for Normal Repossession</p> <p>Name: Account Status Code for Voluntary Surrender Required (Yes/No): Yes Description: Account Status Code for Voluntary Surrender</p>
Detailed Design	Metro 2 Reporting - Account Status Code

Table 6–5 Metro 2 Reporting - Compliance condition code C1- COMCODE

Description	Metro 2 Reporting - Compliance condition code
Detailed Description	Set the Compliance Condition Code sent to Credit Bureau with the value selected in the characteristic given in the parameter.
Algorithm Entity	Result Type - Post Processing Algorithm
Program Type	Java

Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.Metro2ComplianceCodePostProcessingAlgo
Parameters	Name: Compliance Condition Code Char Required (Yes/No): Yes Description: Compliance Condition Code Char
Detailed Design	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

Description	Metro 2 Reporting - Marking Account as Close
Detailed Description	The logic is incorporated for Metro Algorithm only if an Account is close then it should be marked as Close
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2CheckForOpenStatusEnterProcessing
Parameters	NA
Detailed Design	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

Description	Metro 2 Reporting - Consumer Information Indicator
Detailed Description	Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInformationIndicator
Parameters	Name: Chapter7 CII Code Required (Yes/No): Yes

	<p>Description: Chapter7 CII Code</p> <p>Name: Chapter11 CII Code Required (Yes/No): Yes Description: Chapter11 CII Code</p> <p>Name: Chapter12 CII Code Required (Yes/No): Yes Description: Chapter12 CII Code</p> <p>Name: Chapter13 CII Code Required (Yes/No): Yes Description: Chapter13 CII Code</p> <p>Name: Other CII Code Required (Yes/No): Yes Description: Other CII Code</p>
Detailed Design	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

Description	Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post Discharge
Detailed Description	<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 = <Chapter12 CII Code> for Chapter 12</p> <p>Report CII = <Chapter13 CII Code>for Chapter 13</p>
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInfoIndiChap13PostDis
Parameters	<p>Name:No Confirmed Plan CII Code Required (Yes/No): Yes Description: No Confirmed Plan CII Code</p> <p>Name: Chapter12 CII Code Required (Yes/No): Yes Description: Chapter12 CII Code</p>

	Name: Chapter13 CII Code Required (Yes/No): Yes Description: Chapter13 CII Code
Detailed Design	If any associated secured account without confirmed plan on it report CII as per No Confirmed Plan CII Code parameter. Else Report CII = <Chapter12 CII Code> for Chapter 12 Report CII = <Chapter13 CII Code>for Chapter 13

Table 6–9 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1-CGCLC

Description	Credit Grantor Cannot Locate Consumer
Detailed Description	Automatically Set in Skip Tracing Status - Enter Processing. Set for all borrowers on the account. Parameter CII CODE - Mandatory
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CreditGrantorCannotLocateConsumer
Parameters	Name: Cii Code Required (Yes/No): Yes Description: Cii Code
Detailed Design	Credit Grantor Cannot Locate Consumer

Table 6–10 Metro 2 Reporting - Consumer Now Located (Removes previously reported T Indicator) C1-CNLREM

Description	Consumer Now Located (Removes previously reported T Indicator)
Detailed Description	Skip Tracing Actions New Result: Metro 2: Consumer Information Indicator Chars: Party Id (Adhoc) CII (Predefined Values: (T,U) Post Processing: Set the given CII Code for the party id provided.

Algorithm Entity	Result Type - Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ConsumerNowLocated
Parameters	<p>Name: Party Id Char Required (Yes/No): Yes Description: Party Id Char</p> <p>Name: Cii Char Required (Yes/No): Yes Description: Cii Char</p>
Detailed Design	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

Descripti on	Set DPD and Outstanding amount to all associated accounts
Detailed Descripti on	Record the DPD and the Outstanding Balance at account level if number of cases associated with the account of given case type < 2
Algorith m Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.SetDPDOut standingAmount
Paramete rs	<p>Name: Bankruptcy Case Type Required (Yes/No): No Description: Bankruptcy Case Type</p>
Detailed Design	Set DPD and Outstanding amount to all associated accounts

7 Feeder Services

Feeder tables in Oracle Banking Collections act as an additional layer to validate incoming data pulled from the host. Since Oracle Banking Collections has its own architecture and framework, incoming data from any host is validated as per Oracle Banking 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
AccountHardshipDtIsFeederApplicationService	AccountHardshipDtIsFeederResponse update(SessionContext sessionContext, AccountFeederHardshipDtIsWrapperDTO accountFeederHardshipDtIsWrapperDTO) 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 accountArrearFeederWrapperDTO) 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 update(SessionContext sessionContext,AccountWarningIndFeederWrapperDTO accountWarningIndFeederWrapperDTO) 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, 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, FeederPerEmpProfile WrapperDTO feederPerEmpProfileWrapperDTO) throws FatalException	This service adds or updates party employment details fields in the feeder table. It handles add, update and delete operations.	srcHostId, hostCustomerNb r, 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, hostCustomerNb r, 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, hostCustomerNb r, 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 p_ PaymentTrackerDetailsWrapperDTO) 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"> ■ Format: YYYYMMDD. ■ For example, 10-DEC-1982 is represented as 10121982 ■ 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.
N (Numeric 0-9)	Number field: Numeric values must be right justified and zero filled.
Amount field	<ul style="list-style-type: none"> ■ Values must be right justified and zero filled. ■ Must not contain alpha characters, dollar signs, commas, plus and minus signs, decimal point or spaces.
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"> ■ alpha field, must not start with a space or be space filled ■ alphanumeric field, must not start with a space or be space filled ■ numeric field, must not start with a space or be space filled and must not be zero filled, except where specified ■ date field, must not be zero filled.
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"> ■ alpha field, must be space filled ■ alphanumeric field, must be space filled

Data Type	Format
	<ul style="list-style-type: none"> ■ numeric, must be zero filled ■ date field, must be zero filled

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