Oracle® Banking Platform Collections

Interface Specification Guide

Release 2.7.1.0.0

F16199-01

March 2019

Oracle Banking Platform Collections Interface Specification Guide, Release 2.7.1.0.0

F16199-01

Contents

Preface **26**

Audience 26

Documentation Accessibility 26

Organization of the Guide 26

Related Documents 27

Conventions 27

1 Introduction **29**

2 System Overview **31**

3 Staging Area **33**

3.1 Feeder Tables 33

3.1.1 Account Data 33

3.1.1.1 Account Details 33

3.1.1.2 Account Arrears Details 44

3.1.1.3 Account Hardship Details 46

3.1.1.4 Account Repayment Schedule 47

3.1.1.5 Account Warning Indicator 48

3.1.1.6 Minimum Amount Due on Bill (MAD) 49

3.1.1.7 Payment Tracker Details 50

3.1.2 Party Data 51

3.1.2.1 Party Account Relationship 51

3.1.2.2 Party Details 53

3.1.2.3 Party Address Details 57

3.1.2.4 Party Employment Details 59

3.1.2.5 Party Identification Details 61

3.1.2.6 Party Name Details 63

3.1.2.7 Party Contact Preference Details 64

3.1.2.8 Party Warning Indicators 67

3.1.2.9 Service Member History Details 68

3.1.3 Collateral Data 70

3.1.3.1 Collateral Details 70

3.1.3.2 Collateral Charge Details 72

3.1.3.3 Collateral Entity Mapping 73

3.1.3.4 Collateral Guarantor Mapping 74

3.1.3.5 Collateral Owner Mapping 75

3.1.3.6 Collateral Vehicle Mapping 76

3.1.4 Insurance Data 77

3.1.4.1 Insurance Details 77

3.1.5 Payment Data 79

3.1.5.1 Online Payment Records 79

3.1.6 IRS Reporting 80

3.2 Interfacing Tables 81

3.2.1 Agency or Vendor Upload 81

3.2.1.1 Upload Followup Table Details 81

3.2.1.2 Upload Result Table Details 83

3.2.1.3 Upload PTP Table Details 84

3.2.1.4 Upload PTP Schedule Table Details 85

3.2.2 Dialer Results Upload 86

3.2.2.1 Upload Dialer Result Table Details 86

3.2.3 Account Dialer Extract 90

3.2.3.1 Dialer Extract Account Table Details 90

3.2.3.2 Dialer Extract Contact Table Details 92

3.3 OBP Views 93

3.3.1 Main Account Views 93

3.3.2 Account Updateable Views 94

3.3.3 Hardship Views 94

3.3.4 Party Views 94

3.3.5 LCM / Collateral Views 94

4 Algorithms 95

4.1 Stop Contract: C1-CURENTITY 95

4.2 Cure Account: C1-FINCOLL 95

4.3 Queue Allocation: C1-ALLOCQUEU 96

4.4 Update Customer Switch: C1-CUSTSW 97

4.5 Update Legal/Repo Switch: C1-LEREPOCT 98

4.6 User Allocation - Round Robin: C1-USRALCRR 99

4.7 User Allocation - % Based: C1-USRALCPR 100

4.8 Vendor Allocation - Round Robin: C1-VENALCRR 102

4.9 Vendor Allocation - % Based: C1-VENALCPR 103

4.10 Bulk Contact Creation: C1-BLKCNTCRE 105

4.11 Cross Strategy Action Matrix: C1-CSAM 105

4.12 Last Payment for Account: C1-PAYDTAMTU 106

4.13 Association Review Check: C1-ASORVCHK 107

4.14 Validate Expired Default Notice: C1-DEFNOEXP 108

4.15 Associate Related Entity: C1-ASSOENTY 109

4.16 Validate Legal Case Exists: C1-CHKLGL 110

4.17 Assign New LSP: C1-ASGNLSP 111

4.18 Check Approval Requirement: C1-APPRCHK 112

4.19 Resume Status from Previous LSP: C1-RESSTATUS 114

4.20 Check Submission Date: CI_CHKSUBDT1 114

4.21 Update LSP (CLOS): C1-LSPSTATUS 115

4.22 Update LSP (CANCEL): C1-LSPSTACAN 116

4.23 Collateral Verification: C1-VRFYCOLS 116

4.24 Account Association for Asset Repossession Case: C1-ARSACCTS 117

4.25 Customer Association for Asset Repossession Case: C1-ARSCUSTS 118

4.26 Update Collateral Property: C1-UPCOLPROP 119

4.27 Close To do's Algorithm: C1-CLSTODO 119

4.28 Validations for Mandatory Characteristics: C1-CHARVALS 120

4.29 Validations for Mandatory Characteristics: C1-CHARVALA 122

4.30 Update Collateral Status in the Host: C1-UPCOLLSTX 124

4.31 Initiate Collateral Valuation: C1-COLLVALX 125

4.32 Validation Settlement: C1-VALSET 126

4.33 Initiate LMI Process: C1-INITLMI 126

4.34 PTP Active Algorithm: C1-PTPACTIVE 128

4.35 PTP Kept Algorithm: C1-PTPKEPT 129

4.36 PTP Broken Algorithm: C1-BRKPTPNGP 131

4.37 Rule facts populating algorithm: C1-BRLSR 132

4.38 Borrower Centric Case Lifecycle 135

4.39 Update Collection Address on Borrower Panel 136

4.40 Update Collection Contact Point 136

4.41 Bankruptcy Process 137

4.42 Task - Automatic Allocation of tasks to Vendors 152

4.43 Hardship - Associate Accounts of Main Customer 153

4.44 Early Collection 153

4.45 Asset Repossession 172

4.46 Miscellaneous 187

4.47 Derived Field 190

4.48 Task 190

4.49 Event Manager 192

4.50 Legal Vendor Allocation C1-LGLVNDRAL 193

4.51 Extend Expiry Date C1-EXT-EXPDT 193

4.52 Capture Case Status Update Date/Time C1-CASE-STAT 194

4.53 Create To Do C1-TO-DO 195

4.54 Check customer eligibility C1-CHKCUST 195

4.55 Capture Hardship Approval Date C1-HARAP-DT 197

4.56 Algorithm that will interface with Rule Engine. C1-RULEADAPT 198

4.57 Perform Validation for Collateral C1-VRFYCOL 198

4.58 Check target account eligibility C1-CHKTRGT 198

4.59 Approval check for set-off transaction C1-ROSOAPPR 200

4.60 Get target accounts C1-GETTRGT 200

4.61 Update status of relief to Expired in Hardship C1-UPDHDSTAT 202

4.62 To Do Completion for case C1-TO-DO-END 203

4.63 Update Marketing Consent flag C1-MKT-FLG 203

4.64 Check Default Notice for Voluntary possession C1-CHKDFLT 204

4.65 Check Submission Date C1-CHKSUBDT2 204

4.66 Update Financial Hardship flag C1-FNHRD-FLG 205

4.67 Result Type Case Transition Algorithm C1-RTCT 205

4.68 Algorithm to see if case is running before closing C1-CHKCASE 206

4.69 Check Deceased status for the customer C1-CHKDCD 206

4.70 Associated accounts with deceased customer case C1-DCDACCTS 207

4.71 Execute Fund Transfer C1-FUNDTRFR 207

4.72 Algorithm to save previous state's status code C1-SAVPRESTA 208

4.73 Attach case type from feature config attach to BO C1-ATCHCS 209

4.74 Update Collection Warning Indicator C1-UPD-WRIND 209

4.75 Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSO 210

4.76 Assign Applicable Relief Type C1-RELIF-TYP 211

4.77 Create Customer Contact for Resultype Algo C1-CREATCC 211

4.78 Calculate an expiry date when entering case status C1-CSEXPDT 212

4.79 Create Customer Contact C1-CUST-CONT 212

4.80 Transition to Next Status x days before expiry C1-NXT-BX-DY 213

4.81 Validate Hardship Expiry Date. C1-VAL-FHEXP 214

4.82 Update Account in collections flag C1-ACTINCOL 214

4.83 Associate related entities with the case C1-ARSENTITY 215

4.84 Revalidate target account C1-REVALTRGT 215

4.85 Initiate LMI C1-INITLMIS 218

4.86 NGP Collection case creation algorithm C1-COLLCASE 219

4.87 Stop Contract Algorithm C1-CONTSTOP 219

4.88 Check for existing Hardship C1-CHKHRDSHP 220

4.89 Algorithm for contact processing C1-CNTCT 220

4.90 Check application expiry date C1-CHKEXP 220

4.91 New Customer Contact Creation Algorithm C1-CCCREATE 221

4.92 Removes a case characteristic on case status exit C1-REMCSCH 221

4.93 Transition case on a date on a case characteristic C1-TRANSDT 222

4.94 Set Account Nxt Credit Review Date to current date C1-NXTRVWDT 222

4.95 Mark accounts for strategy review C1-REVIW-ACT 223

4.96 Wait Time Out (in days) C1-WAIT-DAYS 223

4.97 Validate Hardship Application inputs C1-V-FH-APP 224
4.98 Check for Operational Relief Types C1-OP-RT 224
4.99 Auto-Approval Check C1-FH-AUTOAP 225
4.100 Apply Hardship Relief Types for accounts in Host C1-FH-EVAL 226
4.101 Update Party Warning Indicator C1-UPD-PRTWI 226
4.102 Transition to Next status when all reliefs are app C1-RAPP 227
4.103 Collection - Entity Activity Population. C1-ENTACTPOP 228
4.104 Cancel Hardship Application C1-CXLFH 228
4.105 Perform Payment Transfer for ROSO C1-ROSOPMTXR 229
4.106 Validate ROSO Target Account inputs C1-RS-VALIN 229
4.107 Create RMB Entities from Host Data C1-VCREATE 230
4.108 Populate Activity Table For Notes Creation C1-NTACTVITY 231
4.109 Suspend Activity for Account Pre Processing C1-SPATACPRE 231
4.110 Sample TAM Algorithm Type C1-TAMALG 231
4.111 Cancel Approval Request C1-CANAPPR 232
4.112 Update Customer in collections flag C1-CUSINCOL 232
4.113 Set Display Date C1-SETDSPDT 233
4.114 Transition to Default next status after N Days C1-TRNDFLT 234
4.115 Check current cases on account for exclusion C1-EXCLCASE 234
4.116 Update Collateral Status in the host C1-UPCOLLST 236
4.117 Initiate collateral valuation C1-COLLVAL 236
4.118 Mandatory characteristics check for Asset Repo C1-CHARVAL 237
4.119 Update Collateral Status in the Host C1-UPCOLLSTS 238
4.120 Set exclusion date for delinquent account. C1-EXCLROSO 238
4.121 Cancel Set-off C1-CANROSO 239
4.122 Complete Set-off C1-COMPROSO 240

4.123 Reverse Set-off C1-REVROSO 240
4.124 Algorithm type for update case id for Insurance C1-UPCASFINS 241
4.125 Case Creation on enter processing C1-CCOENTER 241
4.126 Collection - Case Creation On Exit of Status C1-CCOE 241
4.127 Action category Validation algorithm C1-ACTCAT 242
4.128 Action Type Algorithm Type C1-ACTTYP 242
4.129 Case Type Status Mapping Algorithm Type C1-CASETYMP 242
4.130 Collection - Close Processing Algorithm C1-CCALG 243
4.131 Algorithm type for case list update C1-CASELIST 244
4.132 Copy Characteristics Algorithm C1-COPYCHAR 244
4.133 Call Advice - Red/Green logic calculation C1-CALADVICE 245
4.134 Task Case Mapping Validation Algorithm C1-TCVAL 245
4.135 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status C1-DISMON 246
4.136 SLA Parameters validation algorithm C1-SLAPARAM 248
4.137 Case Group add validation algorithm C1-CGVAL 248
4.138 Collection - Get Strategy Algorithm C1-COLGS 248
4.139 Create/Move Collection Strategy Cases for Account C1-COLOMR 249
4.140 Collection - Case SA Update for Manual Creation C1-CSAUPD 250
4.141 Promise to Pay - Additional Grace Days Sample Algo C1-PPADDLGRD .250
4.142 Promise to Pay Threshold Percentage C1-PPTHRESH 251
4.143 Result type Post Processing Case Transition Algo C1-RTPCC 252
4.144 Inbound Customer algorithm C1-IN-CUST 252
4.145 Result Type Pre-processing Algorithm Type C1-RSTPRE 253
4.146 Result Type Post-processing Algorithm Type C1-RSTPOST 254
4.147 Characteristic Type :Validate Date Field (Custom) C1-ADHDATE 254
4.148 Algorithm Type for Dialer Results Upload C1-DLRRSUPLD 255

4.149 Algorithm for Hardship case creation activity C1-CRTHDSP 255

4.150 This Algorithm is used to abort Approval work item C1-ABORTAPP 256

4.151 Cancel Process Approval Request:Financial Hardship C1-CANFHAPPR 256

4.152 Hardship Characteristic Association C1-FHCHARASC 257

4.153 Pre-Populated system facts to be used for Rule C1-PPSF 258

4.154 Assign Batch level TODOs(task) to a queue C1-ASGNTASK 258

4.155 Validate Date Field :Custom C1-ADHV-DTD 258

4.156 Characteristic Date field Validation C1-CHARDTVAL 260

4.157 Retry Case in Error C1-RCASEE 261

4.158 Allocate Queue for Customer Level Case C1-ALLOCQUE 262

4.159 Person Address – Collection C1-PERADDRC 262

4.160 Person Contact Point Update - Post Processing C1-PERCONTPP 262

4.161 Update Self Serve Flag Algorithm C1-SELFSERVE 263

4.162 Create Task for Self Serve Request for Assistance transaction C1-FLWRTSK 263

5 Localized Algorithms 265

5.1 Localized Algorithms 265

6 Feeder Services 273

7 Dialer Webservice Integration 281

7.1 Generic Data Type 281

7.2 Summary 282

7.3 Interface 282

7.4 Service Management 283

7.5 Request Message Details 283

7.6 Header Record 284

7.7 Detail Record 284

7.8 Translation Rules 284

7.9 Response Message Details 285

7.10 Customer Information 285

7.11 Constraints 285

List of Tables

Table 3–1 Account Details 33
Table 3–2 Account Arrears Details 44
Table 3–3 Account Hardship Details 46
Table 3–4 Account Repayment Schedule 47
Table 3–5 Account Warning Indicator 48
Table 3–6 Minimum Amount Due Bill 49
Table 3–7 Payment Tracker Details 50
Table 3–8 Account Party Relationship 51
Table 3–9 Party Details 54
Table 3–10 Party Address Details 57
Table 3–11 Party Employment Details 59
Table 3–12 Party Identification Details 61
Table 3–13 Party Name Details 63
Table 3–14 Party Contact Preference Details 65
Table 3–15 Party Warning Indicators 68
Table 3–16 Service Member History Details 69
Table 3–17 Collateral Details 70
Table 3–18 Collateral Charges Details 72
Table 3–19 Collateral Entity Mapping 73
Table 3–20 Collateral Guarantor Mapping 74
Table 3–21 Collateral Owner Mapping 75
Table 3–22 Collateral Vehicle Mapping 76
Table 3–23 Insurance Details 77
Table 3–24 Online Payment 79

Table 3–25 IRS Report Data 81
Table 3–26 Upload Followup Table 81
Table 3–27 Upload Result Table 83
Table 3–28 PTP Upload data 84
Table 3–29 PTP Schedule Upload data 85
Table 3–30 Dialer Result Upload data 86
Table 3–31 Account Dialer Extract data. 90
Table 3–32 Dialer Extract Contact data. 92
Table 4–1 Stop Contract: C1-CURENTITY 95
Table 4–2 Cure Account: C1-FINCOLL 95
Table 4–3 Cure Account: Sample Algorithm 96
Table 4–4 Queue Allocation: C1-ALLOCQUEU 96
Table 4–5 Queue Allocation: Sample Algorithm 97
Table 4–6 Update Customer Switch: C1-CUSTSW 97
Table 4–7 Update Customer Switch: Sample Algorithm 97
Table 4–8 Update Legal/Repo Switch: C1-LEREPOCT 98
Table 4–9 Update Legal/Repo Switch: Sample Algorithm 98
Table 4–10 User Allocation - Round Robin: C1-USRALCRR 99
Table 4–11 User Allocation - % Based: C1-USRALCPR 100
Table 4–12 User allocation percentage of Queue Q1 100
Table 4–13 Calculations for allocating cases 100
Table 4–14 Final Distribution 101
Table 4–15 User allocation percentage of Queue Q2 101
Table 4–16 Calculations for allocating cases 101
Table 4–17 Final distribution in cases 102
Table 4–18 Vendor Allocation - Round Robin: C1-VENALCRR 102

Table 4–19 Vendor Allocation - % Based: C1-VENALCRR 104
Table 4–20 Bulk Contact Creation: C1-BLKCNTCRE 105
Table 4–21 Cross Strategy Action Matrix: C1-CSAM 106
Table 4–22 Cross Strategy Action Matrix: Sample Algorithm 106
Table 4–23 Last Payment for Account: C1-PAYDTAMTU 107
Table 4–24 Association Review Check: C1-ASORVCHK 107
Table 4–25 Association Review Check: Sample Algorithm 108
Table 4–26 Validate Expired Default Notice: C1-DEFNOEXP 108
Table 4–27 Validate Expired Default Notice: Sample Algorithm 109
Table 4–28 Associate Related Entity: C1-ASSOENTY 109
Table 4–29 Associate Related Entity: Sample Algorithm 110
Table 4–30 Validate Legal Case Exists: C1-CHKLGL 110
Table 4–31 Validate Legal Case Exists: Sample Algorithm 110
Table 4–32 Assign New LSP: C1-ASGNLSP 111
Table 4–33 Assign New LSP: Sample Algorithm 112
Table 4–34 Check Approval Requirement: C1-APPRCHK 112
Table 4–35 Check Approval Requirement: Sample Algorithm 113
Table 4–36 Resume Status from Previous LSP: C1-RESSTATUS 114
Table 4–37 Check Submission Date: CI_CHKSUBDT1 114
Table 4–38 Check Submission Date: Sample Algorithm 115
Table 4–39 Update LSP (CLOS): C1-LSPSTATUS 115
Table 4–40 Update LSP (CLOS): Sample Algorithm 116
Table 4–41 Update LSP (CANCEL): C1-LSPSTACAN 116
Table 4–42 Update LSP (CANCEL): Sample Algorithm 116
Table 4–43 Collateral Verification: C1-VRFYCOLS 117
Table 4–44 Collateral Verification: Sample Algorithm 117

Table 4–45 Account Association for Asset Repossession Case: C1-ARSACCTS .117

Table 4–46 Customer Association for Asset Repossession Case: C1-ARSCUSTS 118

Table 4–47 Update Collateral Property: C1-UPCOLPROP 119

Table 4–48 Close To do's Algorithm:C1-CLSTODO 119

Table 4–49 Close To do's Algorithm: Sample Algorithm 120

Table 4–50 Validations for Mandatory Characteristics:C1-CHARVALS 121

Table 4–51 Validations for Mandatory Characteristics: Sample Algorithm 122

Table 4–52 Validations for Mandatory Characteristics: C1-CHARVALA 122

Table 4–53 Validations for Mandatory Characteristics: Sample Algorithm 123

Table 4–54 Update Collateral Status in the Host: C1-UPCOLLSTX 124

Table 4–55 Update Collateral Status in the Host: Sample Algorithm 124

Table 4–56 Initiate Collateral Valuation: C1-COLLVALX 125

Table 4–57 Initiate Collateral Valuation: Sample Algorithm 125

Table 4–58 Validation Settlement: C1-VALSET 126

Table 4–59 Validation Settlement: Sample Algorithm 126

Table 4–60 Initiate LMI Process: C1-INITLMI 127

Table 4–61 Initiate LMI Process: Sample Algorithm 128

Table 4–62 PTP Active Algorithm: C1-PTPACTIVE 128

Table 4–63 PTP Active Algorithm: Sample Algorithm 129

Table 4–64 PTP Kept Algorithm: C1-PTPKEPT 129

Table 4–65 PTP Active Algorithm: Sample Algorithm 130

Table 4–66 PTP Broken Algorithm: C1-BRKPTPNGP 131

Table 4–67 Rule Facts Populating Algorithm: C1-BRLSR 132

Table 4–68 Sample Algorithm 134

Table 4–69 Borrower Level: C1-ASSODELAC 135

Table 4–70 Borrower Level : C1-BRWRSW_Y 135

Table 4–71 Borrower Level : C1-BRWRTRNDF 135

Table 4–72 Borrower Level : C1-BRWRSW_N 136

Table 4–73 Person Address Update -Pre-Processing: C1-PADDPRE 136

Table 4–74 Collection Address Post Processing: C1-PERADDPP 136

Table 4–75 Person Contact Point Update - Pre Processing: C1-PCONTPRE 137

Table 4–76 Check if Special Case Already Exist on the Customer- Enter Processing: C1-CKSPLCASE 137

Table 4–77 Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullNDAcc 138

Table 4–78 Associate all accounts to the case where customer is a primary borrower- Enter Processing: C1-ASSCTEACC 139

Table 4–79 Exclude all the associated accounts from Dialer- Enter Processing: C1-ExcAccDlr 139

Table 4–80 Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing: C1-IniCltVal 140

Table 4–81 Monitor if any of the associated account need to be charged off and monitor delinquency- Monitoring: C1-MTRCRGDQY 141

Table 4–82 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG 142

Table 4–83 Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring: C1-MTRPYMPLN 143

Table 4–84 Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD 143

Table 4–85 Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring: C1-MTRHRNGDT 144

Table 4–86 Determine in which status the case should proceed for Bankruptcy Treatment- Post Processing C1-DTMBKTRTM 145

Table 4–87 Validate if appropriate Case Details have been entered by the user- Post Processing C1-VLDBCDATA 146

Table 4–88 Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK 148

Table 4–89 Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP 148

Table 4–90 Joint Bankruptcy - Associate other customers to the Bankruptcy case C1-ASSCUSTJB 149

Table 4–91 Create Pay Plan for a Bankruptcy Case - Enter Processing: C1-CRTATP 149

Table 4–92 Close Pay Plan for a Bankruptcy Case - Exit Processing: C1-CLDATP 150

Table 4–93 Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days - Monitoring: C1-MTRARPLNT 150

Table 4–94 Set or Reset Account level Warning Indicator for Bankruptcy - Enter Processing: C1-SETWI 151

Table 4–95 Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type - Post Processing C1-TSKVNDR 152

Table 4–96 Hardship - Associate Accounts of Main Customer - Enter Processing C1-HARASOPND 153

Table 4–97 Transition to Contact Statuses - Monitoring C1-ECIC 153

Table 4–98 Park Small Balance Accounts - Monitoring C1-ECPSBA 155

Table 4–99 Initiate Skip Tracking - No Telephone Number- Enter Processing C1-ECISTNTN 155

Table 4–100 Initiate Skip Tracking - No Telephone Number- Monitoring C1-ECTTSS 156

Table 4–101 Validate Contact Cap- Monitoring C1-ECVCC 157

Table 4–102 Schedule Contact - Monitoring C1-ECSC 158

Table 4–103 Initiate Skip Tracing - Wrong Telephone Number- Monitoring C1-ECISTITN 159

Table 4–104 Transition to Under Resolution Status- Monitoring C1-ECTTURS 160

Table 4–105 Resume Contact From Under Resolution- Monitoring C1-ECRCFUR 161

Table 4–106 Resume Contact from Small Balance- Monitoring C1-ECRCSB 162

Table 4–107 Determine Contact Intensity - Monitoring C1-ECDCI 163

Table 4–108 Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing C1-CTRANTCRE 163

Table 4–109 Refer to Supervisor- Result Type - Post Processing C1-ECRTS 165

Table 4–110 Resume Collections- Result Type - Post Processing C1-RESCOLL . 166

Table 4–111 Create case on Follow up- Result Type - Post Processing C1-CRETCSFL 167

Table 4–112 Hold Case- Result Type - Post Processing C1-HOLDCASE 168

Table 4–113 Set Case Data- Result Type - Enter Processing C1-ECUPCASE 169

Table 4–114 This algorithm will transition the case status to the Suspension status if Cease and Desist = Y C1-CSETRANS 170

Table 4–115 Algorithm is used for scheduling call C1-SCHCALL 170

Table 4–116 Reset WI in the host C1-RESETWISCHCALL 171

Table 4–117 Validate Collateral - Enter Validation C1-VALDCOLL 172

Table 4–118 Validate Demand Letter and Acceleration Letter - Enter Validation C1-VALIDDLAL 172

Table 4–119 Associate Customers in Repossession Case - Enter Validation C1-ASSOCUST 174

Table 4–120 Bankruptcy Check on Associate Customers - Enter Status C1-CHKBKPTCY 174

Table 4–121 Monitor if Demand letter and Acceleration letter have been sent on the account. C1-MNTRDLAL 175

Table 4–122 Auto Approval Check for Repossession C1-REPOAPRV 176

Table 4–123 Repossession Setup Complete C1-RSTUPCMPL 177

Table 4–124 Automatic task creation for vendors C1-AUTOTASKC 178

Table 4–125 Notify Repossession Specialist on Task Completion C1-NOTRSTSK 178

Table 4–126 Automatic sending of Redemption letters C1-REDEMPLTR 179

Table 4–127 Monitor for Redemption Proceeds C1-REDEPROC 180

Table 4–128 Validate if appropriate Case Details have been entered by the user and transition C1-VALDATAPR 180

Table 4–129 Monitor for Liquidation Setup Complete C1-LIQSETCMP 182

Table 4–130 Send Repossession Alert to Vendor C1-REPOASAL 182

Table 4–131 Extract Algorithm Repossession Assignment C1-REPEMTEMP 183

Table 4–132 Monitor Redemption Clear Date C1-REDCLRDT 183

Table 4–133 Result Post Processing Algorithm for Approvals C1-RAPRVRSLT ...184

Table 4–134 Adhoc Characteristic Value Validation Algorithm PASTDATE_VAL .185

Table 4–135 Result Post Processing Algorithm for Redemption Clear Date C1-RDEEMDATE 186

Table 4–136 Update Review Date for associated accounts C1-UPDRVWDT 187

Table 4–137 Case Monitoring CS-MONITOR 187

Table 4–138 Update warning indicator for the customer C1-UPDWARN 188

Table 4–139 Transition to Default Next Status C1-TRAN-STAT 189

Table 4–140 Set Account Warning Indicator C1-ACTCSWGID 190

Table 4–141 Timezone derivation field update algorithm C1-TZDRFLD 190

Table 4–142 Validate Task Completion (Case Closure) C1-VALTASKCM 190

Table 4–143 Validate Task Completion C1-VALTASKEX 191

Table 4–144 Automatic Task Creation when case enters a particular status C1-CREATTASK 191

Table 4–145 Set Account Warning Indicator 192

Table 4–146 Legal vendor Allocation C1-LGLVNDRAL 193

Table 4–147 Validate Extended Expiry Date C1-VAL-EXPDT 193

Table 4–148 Extend Expiry Date C1-EXT-EXPDT 193

Table 4–149 Capture Case Status Update Date/Time C1-CASE-STAT 194

Table 4–150 Create To Do C1-TO-DO 195

Table 4–151 Check customer eligibility C1-CHKCUST 195

Table 4–152 Capture Hardship Approval Date C1-HARAP-DT 197

Table 4–153 Algorithm that will interface with Rule Engine. C1-RULEADAPT 198

Table 4–154 Perform Validation for Collateral C1-VRFYCOL 198

Table 4–155 Check target account eligibility C1-CHKTRGT 198

Table 4–156 Approval check for set-off transaction C1-ROSOAPPR 200

Table 4–157 Get target accounts C1-GETTRGT 200

Table 4–158 Update status of relief to Expired in Hardship C1-UPDHDSTAT 202

Table 4–159 To Do Completion for caseC1-TO-DO-END 203

Table 4–160 Update Marketing Consent flag C1-MKT-FLG 203

Table 4–161 Check Default Notice for Voluntary possession C1-CHKDFLT 204

Table 4–162 Check Submission Date C1-CHKSUBDT2 204

Table 4–163 Update Financial Hardship flag C1-FNHRD-FLG 205

Table 4–164 Update Financial Hardship flag C1-FNHRD-FLG 205

Table 4–165 Algorithm to see if case is running before closing. C1-CHKCASE 206

Table 4–166 Check Deceased status for the customer .C1-CHKDCD 206

Table 4–167 Associated accounts with deceased customer case.C1-DCDACCTS207

Table 4–168 Execute Fund Transfer C1-FUNDTRFR 207

Table 4–169 Algorithm to save previous state's status code.C1-SAVPRESTA 208

Table 4–170 Attach case type from feature config attach to BO. C1-ATCHCS 209

Table 4–171 Update Collection Warning Indicator. C1-UPD-WRIND 209

Table 4–172 Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSOD 210

Table 4–173 Assign Applicable Relief Type C1-RELIF-TYP 211

Table 4–174 Create Customer Contact for Resultype Algo. C1-CREATCC 211

Table 4–175 Calculate an expiry date when entering case status. C1-CSEXPDT .212

Table 4–176 Create Customer Contact. C1-CUST-CONT 212

Table 4–177 Transition to Next Status x days before expiry C1-NXT-BX-DY 213

Table 4–178 Validate Hardship Expiry Date. C1-VAL-FHEXP 214

Table 4–179 Update Account in collections flag. C1-ACTINCOL 214

Table 4–180 Associate related entities with the case. C1-ARSENTITY 215

Table 4–181 Revalidate target account. C1-REVALTRGT 215

Table 4–182 Initiate LMI. C1-INITLMIS 218

Table 4–183 NGP Collection case creation algorithm. C1-COLLCASE 219

Table 4–184 Stop Contract Algorithm. C1-CONTSTOP 219

Table 4–185 Check for existing Hardship. C1-CHKHRDSHP 220

Table 4–186 Algorithm for contact processing. C1-CNTCT 220

Table 4–187 Check application expiry date. C1-CHKEXP 220

Table 4–188 New Customer Contact Creation Algorithm. C1-CCCREATE 221

Table 4–189 Removes a case characteristic on case status exit. C1-REMCSCH .. 221

Table 4–190 Transition case on a date on a case characteristic. C1-TRANSDT ... 222

Table 4–191 Set Account Nxt Credit Review Date to current date. C1-NXTRVWDT 222

Table 4–192 Mark accounts for strategy review. C1-REVIW-ACT 223

Table 4–193 Wait Time Out (in days). C1-WAIT-DAYS 223

Table 4–194 Validate Hardship Application inputs. C1-V-FH-APP 224

Table 4–195 Check for Operational Relief Types .C1-OP-RT 224

Table 4–196 Auto-Approval Check. C1-FH-AUTOAP 225

Table 4–197 Apply Hardship Relief Types for accounts in Host. C1-FH-EVAL 226

Table 4–198 Update Party Warning Indicator. C1-UPD-PRTWI 226

Table 4–199 Transition to Next status when all reliefs are app. C1-RAPP 227

Table 4–200 Collection - Entity Activity Population. C1-ENTACTPOP 228

Table 4–201 Cancel Hardship Application. C1-CXLFH 228

Table 4–202 Perform Payment Transfer for ROSO. C1-ROSOPMTXR 229

Table 4–203 Validate ROSO Target Account inputs C1-RS-VALIN 229

Table 4–204 Create RMB Entities from Host Data. C1-VCREATE 230

Table 4–205 Populate Activity Table For Notes Creation. C1-NTACTVITY 231

Table 4–206 Suspend Activity for Account Pre Processing C1-USRALCRR 231
Table 4–207 Sample TAM Algorithm Type. C1-TAMALG 231
Table 4–208 Cancel Approval Request. C1-CANAPPR 232
Table 4–209 Update Customer in collections flag. C1-CUSINCOL 232
Table 4–210 Set Display Date. C1-SETDSPDT 233
Table 4–211 Transition to Default next status after N Days. C1-TRNDFLT 234
Table 4–212 Check current cases on account for exclusion. C1-EXCLCASE 234
Table 4–213 Update Collateral Status in the host. C1-UPCOLLST 236
Table 4–214 Initiate collateral valuation. C1-COLLVAL 236
Table 4–215 Mandatory characteristics check for Asset Repo. C1-CHARVAL 237
Table 4–216 Update Collateral Status in the Host. C1-UPCOLLSTS 238
Table 4–217 Set exclusion date for delinquent account. C1-EXCLROSO 238
Table 4–218 Cancel Set-off. C1-CANROSO 239
Table 4–219 Complete Set-off. C1-COMPROSO 240
Table 4–220 Reverse Set-off. C1-REVROSO 240
Table 4–221 Algorithm type for update case id for Insurance. C1-UPCASFINS 241
Table 4–222 Case Creation on enter processing. C1-CCOENTER 241
Table 4–223 Collection - Case Creation On Exit of Status. C1-CCOE 241
Table 4–224 Action category Validation algorithm. C1-ACTCAT 242
Table 4–225 Action Type Algorithm Type. C1-ACTTYP 242
Table 4–226 Case Type Status Mapping Algorithm Type C1-CASETYMP 242
Table 4–227 Collection - Close Processing Algorithm. C1-CCALG 243
Table 4–228 Algorithm type for case list update. C1-CASELIST 244
Table 4–229 Copy Characteristics Algorithm. C1-COPYCHAR 244
Table 4–230 Call Advice - Red/Green logic calculation. C1-CALADVICE 245
Table 4–231 Task Case Mapping Validation Algorithm. C1-TCVAL 245

Table 4–232 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status. C1-DISMON 246

Table 4–233 SLA Parameters validation algorithm. C1-SLAPARAM 248

Table 4–234 Case Group add validation algorithm. C1-CGVAL 248

Table 4–235 Collection - Get Strategy Algorithm. C1-COLGS 248

Table 4–236 Create/Move Collection Strategy Cases for Account. C1-COLOMR .. 249

Table 4–237 Collection - Case SA Update for Manual Creation. C1-CSAUPD 250

Table 4–238 Promise to Pay - Additional Grace Days Sample Algo. C1-PPADDLGRD 250

Table 4–239 Promise to Pay Threshold Percentage. C1-PPTHRESH 251

Table 4–240 Result type Post Processing Case Transition Algo. C1-RTPCC 252

Table 4–241 Inbound Customer algorithm. C1-IN-CUST 252

Table 4–242 Result Type Pre-processing Algorithm Type. C1-RSTPRE 253

Table 4–243 Result Type Post-processing Algorithm Type. C1-RSTPOST 254

Table 4–244 Characteristic Type :Validate Date Field (Custom). C1-ADHDATE .. 254

Table 4–245 Algorithm Type for Dialer Results Upload. C1-DLRRSUPLD 255

Table 4–246 Algorithm for Hardship case creation activity. C1-CRTHDSP 255

Table 4–247 This Algorithm is used to abort Approval work item. C1-ABORTAPP 256

Table 4–248 Cancel Process Approval Request:Financial Hardship. C1-CANFHAPPR 256

Table 4–249 Hardship Characteristic Association. C1-FHCHARASC 257

Table 4–250 Pre-Populated system facts to be used for Rule. C1-PPSF 258

Table 4–251 Assign Batch level TODOs(task) to a queue. C1-ASGNTASK 258

Table 4–252 Validate Date Field :Custom. C1-ADHV-DTD 258

Table 4–253 Characteristic Date field Validation. C1-CHARDTVAL 260

Table 4–254 Retry Case in Error C1-RCASEE 261

Table 4–255 Allocate Queue for Customer Level Case. C1-ALLOCQUE 262

Table 4–256 Person Address – Collection. C1-PERADDRC 262

Table 4–257 Person Contact Point Update - Post Processing. C1-PERCONTPP 262

Table 4–258 Update Self Serve Flag Algorithm. C1-SELFSERVE 263

Table 4–259 Create Task for Self Serve Request for Assistance transaction. C1-FLWRTSK 263

Table 5–1 Case Transition for Active Service Member C1-ACTMEMCHK 265

Table 5–2 Active Military Check on Associated Customers - Enter Validation C1-BLOCKREPO 266

Table 5–3 Metro 2 Reporting - Account Status Code post Liquidation C1-ASCLIQU 266

Table 5–4 Metro 2 Reporting - Account Status Code C1- ASCREPO 267

Table 5–5 Metro 2 Reporting - Compliance condition code C1- COMCODE 268

Table 5–6 Metro 2 Reporting - Marking Account as Close C1- CFOSEP 269

Table 5–7 Metro 2 Reporting - Consumer Information Indicator C1- CONINFOIN 269

Table 5–8 Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post Discharge C1- CIIPSTDIS 270

Table 5–9 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1-CGCLC 271

Table 5–10 Metro 2 Reporting - Consumer Now Located (Removes previously reported T Indicator) C1-CNLREM 271

Table 5–11 Metro 2 Reporting - Set DPD and Outstanding amount to all associated accounts C1-SETDPD 272

Table 6–1 Feeder Services 273

Table 7–1 Generic Data Type 281

Table 7–2 Summary 282

Table 7–3 Interface 282

Table 7–4 Service Management 283

Table 7–5 Detail Record 284

Table 7–6 Customer information 285

Preface

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

This preface contains the following topics:

- Audience
- Documentation Accessibility
- Organization of the Guide
- Related Documents
- Conventions

Audience

This document is intended for the following audience:

- IT Deployment Team
- Consulting Staff
- Administrators

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/us/corporate/accessibility/index.html.

Access to Oracle Support

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

Organization of the Guide

This document contains:

Chapter 1 Introduction

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

Chapter 2 System Overview

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

Chapter 3 Staging Area

This chapter provides details of the feeder tables.

Chapter 4 Algorithms

This chapter outlines the pre-shipped algorithm details.

Chapter 5 Localized Algorithms

This chapter provides a list of Localized algorithm details.

Chapter 6 Feeder Services

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

Chapter 7 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 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.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1 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.

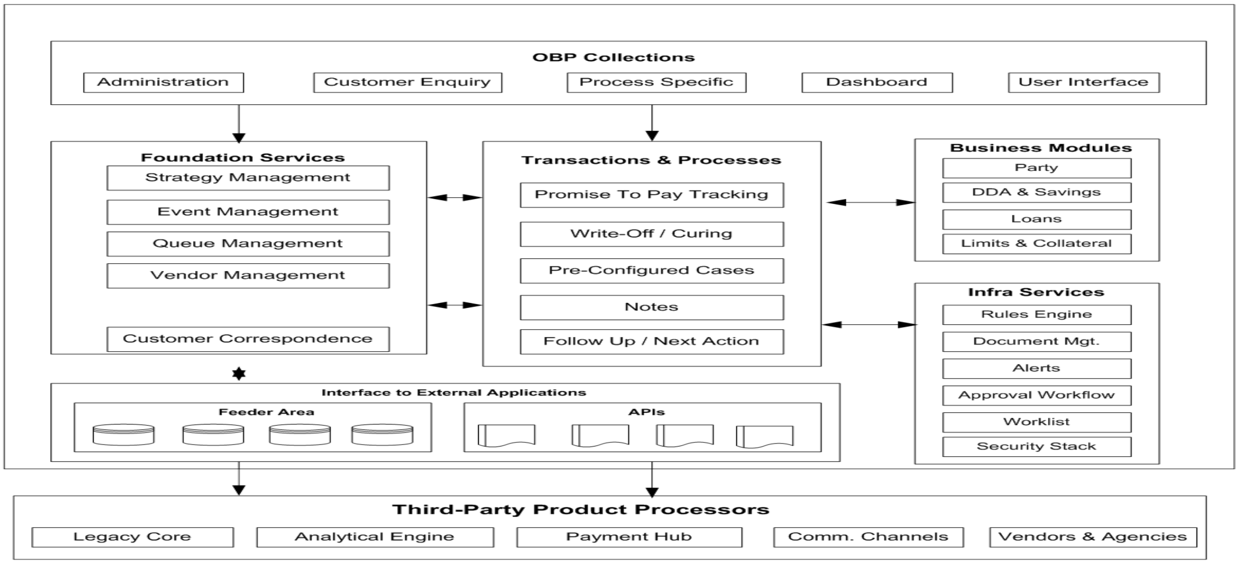
The feeder services exposed by Collections are invoked when changes in data take place in OBP. These services bring modified data into staging tables before batch processes these and update collections tables.

2 System Overview

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

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

Figure 2–1 System Overview



3 Staging Area

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

3.1 Feeder Tables

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

3.1.1 Account Data

This section provides information on the tables related to accounts.

3.1.1.1 Account Details

Table Name: Account Details (CI_FDR_ACCT)

Description: This table holds account related data from host.

Table 3–1 Account Details

Field Name	Description	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

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

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

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

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_DELINQUENCY
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_DELINQUENT_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

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

3.1.1.2 Account Arrears Details

Table Name: Account Arrear Details (CI_FDR_ACCT_ARS)

Description: This table holds account arrears data from host.

Table 3–2 Account Arrears Details

Field Name	Description	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

3.1.1.3 Account Hardship Details

Table Name: Account Hardship Details (CI_FDR_ACCT_HARDSHIP_DTLS)

Description: This table holds account hardship data from host.

Table 3–3 Account Hardship Details

Field Name	Description	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

3.1.1.4 Account Repayment Schedule

Table Name: Account Repayment Schedule (CI_FDR_REPAYMENT_SCH)

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

Table 3–4 Account Repayment Schedule

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Date	Date when the installments are to be recovered		DATE	10	Y	INSTALLMENT_DT
Amount	Installment amount		NUMBER	36,18	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

3.1.1.5 Account Warning Indicator

Table Name: Account Warning Indicator (CI_FDR_ACCT_WARNING_IND)

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

Table 3–5 Account Warning Indicator

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ ACCT_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ ID

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

3.1.1.6 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 3–6 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

3.1.1.7 Payment Tracker Details

Table Name: Payment Tracker Details (CI_FDR_PAY_TRACKER_DETLS)

Description: This table tracks payments done on an account.

Table 3–7 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

3.1.2 Party Data

This section provides information on the tables related to party.

3.1.2.1 Party Account Relationship

Table Name: Party Account Relationship (CI_FDR_ACCT_PER)

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

Table 3–8 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

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

3.1.2.2 Party Details

Table Name: Party Details (CI_FDR_PER)

Description: This table holds party data from host.

Table 3–9 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

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

3.1.2.3 Party Address Details

Table Name: Party Address Details (CI_FDR_PER_ADDR)

Description: This table holds party address data from host.

Table 3–10 Party Address Details

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

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

3.1.2.4 Party Employment Details

Table Name: Party Employment Details (CI_FDR_PER_EMPLOYMENT_PROF)

Description: This table holds party employment details from host.

Table 3–11 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

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

3.1.2.5 Party Identification Details

Table Name: Party Identification Details (CI_FDR_PER_ID)

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

Table 3–12 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

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

3.1.2.6 Party Name Details

Table Name: Party Name Details (CI_FDR_PER_NAME)

Description: This table holds party name details from host.

Table 3–13 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

3.1.2.7 Party Contact Preference Details

Table Name: Party Contact Preference Details (CI_FDR_CONTACT_PREF)

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

Table 3–14 Party Contact Preference Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Party ID	Party ID as stored in Host		VARCHAR2	40	Y	HOST_CUST_ NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Determinant Value	Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features.		VARCHAR2	60	Y	DETERMINANT_ VALUE
Contact Point	Type of Contact Point	Mobile, Landline, Email, and so on.	VARCHAR2	10	Y	CONTACT_ POINT_TYPE
Purpose			VARCHAR2	120	N	PURPOSE_TXT
Value	Contact Point Value, for example, if Contact Point is Mobile then provide mobile number, if Email then provide email ID		VARCHAR2	400	N	CONTACT_ VALUE
Contact Type		Home, Work, Others	VARCHAR2	10	Y	CONTACT_ PREF_TYPE
Start Date	Start date for using this contact point and type		DATE	10	N	START_DT
End Date	End date for using this contact point and type		DATE	10	N	END_DT
Time From (weekdays)	Start Time for contacting on weekdays	In hundred hour format (for example, 1800 for 6:00 PM)	NUMBER	10,0	N	WKDAY_FROM_ TM

Field Name	Description	Value	Data Type	Length	Required	Column Name
Time To (weekdays)	End Time for contacting on weekdays	In hundred hour format (for example, 1800 for 6:00 PM)	NUMBER	10,0	N	WKDAY_TO_TM
Time From (weekends)	Start Time for contacting on weekends	In hundred hour format (for example, 1800 for 6:00 PM)	NUMBER	10,0	N	WKEND_FROM_TM
Time To (weekends)	End Time for contacting on weekends	In hundred hour format (for example, 1800 for 6:00 PM)	NUMBER	10,0	N	WKEND_TO_TM
Preference Frequency	Preferred Frequency of contact		NUMBER	20	N	PREFERENCE_FREQUENCY
Primary Contact Point	Primary Contact Point Flag		VARCHAR2	10	N	FDR_PRIMARY_SW
Status	Status - if Active or Dormant		VARCHAR2	60	Y	STATUS
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert 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

3.1.2.8 Party Warning Indicators

Table Name: Party Warning Indicators (CI_FDR_PARTY_WARNING_IND)

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

Table 3–15 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

3.1.2.9 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 3–16 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

3.1.3 Collateral Data

This section provides information on the tables related to collaterals.

3.1.3.1 Collateral Details

Table Name: Collateral Details (CI_FDR_COLLATERAL)

Description: This table holds collateral data from host.

Table 3–17 Collateral Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Collateral Code	Collateral Code as stored in host		VARCHAR2	40	Y	COLLATERAL_ CD
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Collateral Type	Type of Collateral		VARCHAR2	50	N	COLLATERAL_ TYPE
Collateral Sub Type	If there are any collateral sub type		VARCHAR2	50	N	COLLATERAL_ SUB_TYPE
Collateral Category	Collateral Category		VARCHAR2	50	N	COLLATERAL_ CAT
Collateral Description	Collateral Description		VARCHAR2	300	N	FDR_ COLLATERAL_ DESCR
Nature	Normal/ Guarantee		VARCHAR2	40	N	COLLATERAL_ NATURE
Collateral Currency	Collateral Currency		VARCHAR2	3	N	COLLATERAL_ CUR
Assessed Value	Market Value		NUMBER	36,18	N	ASSESD_ VALUE
Assessment Date	Date of assessment		DATE	10	N	ASSESD_DT

Field Name	Description	Value	Data Type	Length	Required	Column Name
Bank Value	Book Value		NUMBER	36,18	N	BANK_VALUE
Sold By	This property is required to identify entity which sold the collateral.	Customer (Borrower), Bank, Court	VARCHAR2	255	N	SOLD_BY
Date of Sale	Date on which the collateral was sold		DATE	10	N	SALE_DT
Amount Realized	Gross Sale amount		NUMBER	36,18	N	AMT_ REALIZED
Date of Settlement	Date on which settlement took place		DATE	10	N	SETLMNT_DT
Realization Status	Final status of realization		VARCHAR2	60	N	REALIZATION_ STATUS
Amount Recovered	Gross Sale Amount less Costs incurred for sale of collateral		NUMBER	36,18	N	FDR_AMT_ RECOVERED
Collateral Address Line1	Collateral Address Line1		VARCHAR2	120	N	ADDRESS_ LINE1
Collateral Address Line2	Collateral Address Line2		VARCHAR2	120	N	ADDRESS_ LINE2
Collateral Address Line3	Collateral Address Line3		VARCHAR2	120	N	ADDRESS_ LINE3
Collateral Address Line4	Collateral Address Line4		VARCHAR2	120	N	ADDRESS_ LINE4
City code	City code		VARCHAR2	50	N	CITY_CD
Postal code	Postal code		VARCHAR2	30	N	ZIP_CD
State code	State code		VARCHAR2	6	N	STATE_CD
Country code	Country code		VARCHAR2	30	N	COUNTRY_CD
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert 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

3.1.3.2 Collateral Charge Details

Table Name: Collateral Charge Details (CI_FDR_COLLATERAL_CHRG)

Description: This table holds collateral charges details from host.

Table 3–18 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

3.1.3.3 Collateral Entity Mapping

Table Name: Collateral Entity Mapping (CI_FDR_COLLATERAL_ENTITY)

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

Table 3–19 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

3.1.3.4 Collateral Guarantor Mapping

Table Name: Collateral Guarantor Mapping (CI_FDR_COLLATERAL_GRNTR)

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

Table 3–20 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

3.1.3.5 Collateral Owner Mapping

Table Name: Collateral Owner Mapping (CI_FDR_COLLATERAL_PARTY)

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

Table 3–21 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

3.1.3.6 Collateral Vehicle Mapping

Table Name: Collateral Vehicle Mapping (CI_FDR_COLLATERAL_AUTOMOBILE)

Description: This table holds Vehicle information for the collateral.

Table 3–22 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

3.1.4 Insurance Data

This section provides information on the tables related to insurance.

3.1.4.1 Insurance Details

Table Name: Insurance Details (CI_FDR_INSR_DTLS)

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

Table 3–23 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

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

3.1.5 Payment Data

This section provides information on the tables related to payments.

3.1.5.1 Online Payment Records

Table Name: Online Payment (CI_FDR_PAYMENTS)

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

Table 3–24 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

3.1.6 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 3–25 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

3.2 Interfacing Tables

This section provides details about the Interfacing tables.

3.2.1 Agency or Vendor Upload

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

3.2.1.1 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 3–26 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

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

3.2.1.2 Upload Result Table Details

Table Name: Vendor/Agency Upload Result Table (CI_VNDR_UPLD_RESULT)

Description: This table holds Result Upload data.

Table 3–27 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

3.2.1.3 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 3–28 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

3.2.1.4 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 3–29 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_

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

3.2.2 Dialer Results Upload

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

3.2.2.1 Upload Dialer Result Table Details

Table Name: Dialer Result Upload Batch (CI_DIALER_RESULTS_UPLOAD)

Description: This table holds Dialer Result Upload data.

Table 3–30 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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 Fields		VARCHAR2	60	Y	UDF25
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		VARCHAR2	60	Y	UDF38

Field Name	Description	Value	Data Type	Length	Required	Column Name
	Fields					
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
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
UDF59	User Defined Fields		VARCHAR2	60	Y	UDF59
UDF60	User Defined Fields		VARCHAR2	60	Y	UDF60
VERSION	Version		NUMBER	5	Y	VERSION

3.2.3 Account Dialer Extract

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

3.2.3.1 Dialer Extract Account Table Details

Table Name: Account Dialer Extract Table (CI_ACCT_DILR_EXTRCT)

Description: This table holds Account Dialer Extract data.

Table 3–31 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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 Amount	Outstanding Amount		NUMBER	36,18	Y	OUTSTANDING_ 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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

3.2.3.2 Dialer Extract Contact Table Details

Table Name: Dialer Extract Contact Table (CI_DIALER_EXTRACTS_CONTACT)

Description: This table holds Dialer Extract Contact data.

Table 3–32 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
Acceptence Start Date	Acceptence Start Date		DATE		N	ACCEPT_START_DTTM
Acceptence End Date	Acceptence 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
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

3.3 OBP Views

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

3.3.1 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

3.3.2 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

3.3.3 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

3.3.4 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

3.3.5 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 Algorithms

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

4.1 Stop Contract: C1-CURENTITY

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

Table 4–1 Stop Contract: C1-CURENTITY

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.

4.2 Cure Account: C1-FINCOLL

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

Table 4–2 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	This algorithm performs the following activities: - 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	**Name:** contactMethods **Required (Yes/No):** Yes **Description:** Contact Methods soft parameter has a comma-separated value of customer

	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 4–3 Cure Account: Sample Algorithm

Algorithm Name	C1-FINCOL
Parameters	**Name:** contactMethods **Value:** SMS, EM

4.3 Queue Allocation: C1-ALLOCQUEU

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

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

Algorithm Name	C1-ALLOCQUEU
Parameters	**Name:** queueAllocationView **Value:** CI_ALLOCATION_MONITOR_VW

4.4 Update Customer Switch: C1-CUSTSW

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

Table 4–6 Update Customer Switch: C1-CUSTSW

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 4–7 Update Customer Switch: Sample Algorithm

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

	Name: Switch Value **Value:** Y

4.5 Update Legal/Repo Switch: C1-LEREPOCT

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

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

Description	This algorithm is used to update Legal and Repo case status on enter processing.
Detailed Description	Legal Repo Switch Name: Specify the Legal or Repo case switch column name of account extension For example, LEGAL_CASE_EXISTS_SW or REPO_CASE_EXISTS_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.RepoAndLegalCaseUpdateAlgorithm
Parameters	**Name:** Legal Repo 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 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, ■ 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 4–9 Update Legal/Repo Switch: Sample Algorithm

Algorithm Name	C1-LEGALSW
Parameters	**Name:** Legal Repo Switch Name **Value:** LEGAL_CASE_EXISTS_SW **Name:** Switch Value **Value:** Y

4.6 User Allocation - Round Robin: C1-USRALCRR

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

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

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	This algorithm receives input as queue code. The computation logic is explained below: ■ 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: • 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.

4.7 User Allocation - % Based: C1-USRALCPR

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

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

<table>
<tr><td><b>Description</b></td><td>This algorithm is used for allocating cases to users or teams in percentage-based method.</td></tr>
<tr><td><b>Detailed Description</b></td><td>This algorithm allocates cases to user or teams in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC).
User Allocation Percentage based algorithm type allocates cases to users on the basis of percentage allocations set during configuration on queue admin.
OverFlow cases will get assigned to Exception User.</td></tr>
<tr><td><b>Algorithm Entity</b></td><td>User Allocation</td></tr>
<tr><td><b>Program Type</b></td><td>Java</td></tr>
<tr><td><b>Program Name</b></td><td>com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPerBasedAllocRoundOff</td></tr>
<tr><td><b>Parameters</b></td><td>NA</td></tr>
<tr><td><b>Detailed Design</b></td><td>
<ul>
<li>Open unallocated cases will be allocated to valid queue users based on corresponding percentage distribution value mentioned in the Queue Details screen.</li>
<li>Algorithm will round off the decimal percent allocated case count to the nearest whole integer value.</li>
</ul>
For example:
User allocation percentage of Queue Q1 is as follows:
<i><b>Table 4–12 User allocation percentage of Queue Q1</b></i>
<table>
<tr><th>User</th><th>Allocation Percentage</th></tr>
<tr><td>U1</td><td>33%</td></tr>
<tr><td>U2</td><td>33%</td></tr>
<tr><td>U3</td><td>34%</td></tr>
</table>
Total unallocated cases = 10
Then, cases will be allocated as per following calculations:
<i><b>Table 4–13 Calculations for allocating cases</b></i>
<table>
<tr><th>User</th><th>Allocation Percentage</th><th>Calculated Case Allocation</th><th>Actual Case Allocation</th></tr>
<tr><td>U1</td><td>33%</td><td>3.3</td><td>3</td></tr>
</table>
</td></tr>
</table>

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 4–14 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 4–15 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 4–16 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

	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). The final distribution in this case will be as follows: *Table 4–17 Final distribution in cases* User | Allocation Percentage | Calculated Case Allocation | Actual Case Allocation U1 | 33% | 3.3 | 3 U2 | 33% | 3.3 | 4 U3 | 34% | 3.4 | 3 ■ If capacities of all queue users are exhausted, unallocated cases will be assigned to the Queue Exception User.

4.8 Vendor Allocation - Round Robin: C1-VENALCRR

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

Table 4–18 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	This algorithm takes input as Queue code. The computation logic for case capacity is as below: ■ 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

	C1 and C2. ■ 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: • 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: • 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.

4.9 Vendor Allocation - % Based: C1-VENALCPR

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

Table 4–19 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	This algorithm takes input as Queue code. The computation logic for case capacity is as below: ■ 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 • 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

	calculation) and then in decreasing order of capacity. ■ 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: • 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.

4.10 Bulk Contact Creation: C1-BLKCNTCRE

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

Table 4–20 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.

4.11 Cross Strategy Action Matrix: C1-CSAM

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

Table 4–21 Cross Strategy Action Matrix: C1-CSAM

Description	This algorithm is used for Cross Strategy Action Matrix.
Detailed Description	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. Parameters : Check Status- It checks the status with which the matrix has to be dealt with. Possible values are "Y" or "N"
Algorithm Entity	Case Type- CSAM
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.CrossStrategyActionMatrixAlgorithm
Parameters	**Name:** CheckStatus **Required (Yes/No):** N **Description:** Y - Case types with Status N - Case types without status
Detailed Design	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. The two possible actions are: ■ **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". This algorithm should also get triggered during case association process.

Table 4–22 Cross Strategy Action Matrix: Sample Algorithm

Algorithm Name	C1-CSAMY
Parameters	**Name:** CheckStatus **Value:** Y

4.12 Last Payment for Account: C1-PAYDTAMTU

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

Table 4–23 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 freezed for payment. Algorithm will update the FT amount and FT date in Account extension table column LAST_PAYMENT_AMT and LAST_PAYMENT_DT.

4.13 Association Review Check: C1-ASORVCHK

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

Table 4–24 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	**Name:** NextStatus **Required (Yes/No):** N **Description:** Next Status **Name:** AssociationReviewRequired **Required (Yes/No):** Y **Description:** Association Review Required
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 4–25 Association Review Check: Sample Algorithm

Algorithm Name	C1-ASORVCHK
Parameters	**Name:** NextStatus **Value:** ASSNEWLSP **Name:** AssociationReviewRequired **Value:** Y

4.14 Validate Expired Default Notice: C1-DEFNOEXP

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

Table 4–26 Validate Expired Default Notice: C1-DEFNOEXP

Descriptio n	This algorithm is used to validate expired default notices.
Detailed Descriptio n	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. 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
Parameter s	**Name:** associationType **Required (Yes/No):** Y **Description:** Association Type **Name:** validationfailureOption **Required (Yes/No):** Y **Description:** Validation Failure Option **Name:** toDoType **Required (Yes/No):** N **Description:** To Do Type
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 4–27 Validate Expired Default Notice: Sample Algorithm

Algorithm Name	C1-DEFNOEXP
Parameters	**Name:** associationType **Value:** P **Name:** validationfailureOption **Value:** N **Name:** toDoType **Value:** C1-TD-DN

4.15 Associate Related Entity: C1-ASSOENTY

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

Table 4–28 Associate Related Entity: C1-ASSOENTY

Descriptio n	This algorithm is used to associate related entities with the case.
Detailed Descriptio n	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: 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
Parameter s	**Name:** hostId **Required (Yes/No):** Y **Description:** Host Id **Name:** toDoType **Required (Yes/No):** Y **Description:** To Do Type
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 4–29 Associate Related Entity: Sample Algorithm

Algorithm Name	C1-ASSOENTY
Parameters	**Name:** hostId **Value:** NGP **Name:** toDoType **Value:** C1-TD-AC

4.16 Validate Legal Case Exists: C1-CHKLGL

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

Table 4–30 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	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: 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 4–31 Validate Legal Case Exists: Sample Algorithm

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

	Name: toDoType **Value:** C1-TD-CL

4.17 Assign New LSP: C1-ASGNLSP

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

Table 4–32 Assign New LSP: C1-ASGNLSP

Description	This algorithm is used to assign LSP to the case.
Detailed Description	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 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	**Name:** New Allocation And Review Option **Required (Yes/No):** N **Description:** New Allocation and Review Option **Name:** Change Allocation Option

	Required (Yes/No): N **Description:** Change Allocation Option **Name:** Reset Document Submission Date Sw **Required (Yes/No):** N **Description:** Reset Document Submission Date Switch **Name:** Previous Allocation Check **Required (Yes/No):** N **Description:** Previous Allocation Check **Name:** Next Status **Required (Yes/No):** N **Description:** Next Status
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 4–33 Assign New LSP: Sample Algorithm

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

4.18 Check Approval Requirement: C1-APPRCHK

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

Table 4–34 Check Approval Requirement: C1-APPRCHK

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

Description	This process will check if LSP assignment needs to be approved, if LSP assignment status = "Pending Approval" Approval would be required if either of below is true: ■ 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. If approval is required: ■ 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	**Name:** Exposure Threshold **Required (Yes/No):** N **Description:** Exposure Threshold **Name:** Approval Request Status **Required (Yes/No):** N **Description:** Approval Request Status **Name:** Approved Status **Required (Yes/No):** N **Description:** Approved Status **Name:** Reject Request Status **Required (Yes/No):** N **Description:** Reject Request Status
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 4–35 Check Approval Requirement: Sample Algorithm

Algorithm Name	C1-ASGNLSP
Parameters	**Name:** Exposure Threshold **Value:** 10

	Name: Approval Request Status **Value:** PENDINGAPP **Name:** Approved Status **Value:** WTFRLSPACK **Name:** Reject Request Status **Value:** ASSNEWLSP

4.19 Resume Status from Previous LSP: C1-RESSTATUS

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

Table 4–36 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.

4.20 Check Submission Date: CI_CHKSUBDT1

This section provides details of the Check Submission Date: CI_CHKSUBDT1 algorithm.

Table 4–37 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	**Name:** nextStatus **Required (Yes/No):** Y **Description:** NA **Name:** changeStatus **Required (Yes/No):** Y **Description:** NA
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 4–38 Check Submission Date: Sample Algorithm

Algorithm Name	CI_CHKSUBDT1C1
Parameters	**Name:** nextStatus **Value:** WTFRLSPACK **Name:** changeStatus **Value:** Y

4.21 Update LSP (CLOS): C1-LSPSTATUS

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

Table 4–39 Update LSP (CLOS): C1-LSPSTATUS

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.
	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.
	If Status = Closed or Cancelled set Assignment End date = Business Date
	Status possible values {CLOS,REJ,CAN,PNAP}
	CLOS=Closed
	REJ=Rejected
	PNAP=Pending for Approval.

Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssig

Name	nment
Parameter s	**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 4–40 Update LSP (CLOS): Sample Algorithm

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

4.22 Update LSP (CANCEL): C1-LSPSTACAN

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

Table 4–41 Update LSP (CANCEL): C1-LSPSTACAN

Descriptio n	Legal Proceedings - Update Status
Detailed Descriptio n	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
Parameter s	**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 4–42 Update LSP (CANCEL): Sample Algorithm

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

4.23 Collateral Verification: C1-VRFYCOLS

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

Table 4–43 Collateral Verification: C1-VRFYCOLS

Description	Collateral Verification
Detailed Description	This will perform following validations for the collateral with the case: ■ 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	**Name:** Collateral Type **Required (Yes/No):** N **Description:** Collateral Type
Detailed Design	It is invoked in the Pending status of the Asset Repossession Process case. It Verifies the collateral associated with account.

Table 4–44 Collateral Verification: Sample Algorithm

Algorithm Name	C1-VRFYCOLS
Parameters	**Name:** Collateral Type **Value:** PROPERTY

4.24 Account Association for Asset Repossession Case: C1-ARSACCTS

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

Table 4–45 Account Association for Asset Repossession Case: C1-ARSACCTS

Description	Account Association for Asset repossession case
Detailed Description	This algorithm will perform following actions: ■ It gets all facilities to which this collateral is associated and all accounts for these facilities.

	- It associates these accounts with the case. Scope of this association is limited to accounts already in collections. This process will not check for any accounts not in collections. This algorithm doesn't have any soft parameter.
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.

4.25 Customer Association for Asset Repossession Case: C1-ARSCUSTS

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

Table 4–46 Customer Association for Asset Repossession Case: C1-ARSCUSTS

Description	Customer Association for Asset repossession case
Detailed Description	This algorithm performs the following actions: - It gets all customers who are the owners for the selected collateral - It associates these customers with the case Scope of this association is limited to customers already in collections. This process will not check for any customers not in collections. This algorithm does not have any soft parameter.
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.

4.26 Update Collateral Property: C1-UPCOLPROP

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

Table 4–47 Update Collateral Property: C1-UPCOLPROP

Description	Update Collateral Property
Detailed Description	This algorithm will perform following operations: ■ 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	**Name:** UpdateCollateralProperty **Required (Yes/No):** Y **Description:** NA
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.

4.27 Close To do's Algorithm: C1-CLSTODO

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

Table 4–48 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

	Required (Yes/No): N **Description:** To Do Type 1 **Name:** To Do Type2 **Required (Yes/No):** N **Description:** To Do Type 2 **Name:** To Do Type3 **Required (Yes/No):** N **Description:** To Do Type 3 **Name:** To Do Type4 **Required (Yes/No):** N **Description:** To Do Type 4 **Name:** To Do Type5 **Required (Yes/No):** N **Description:** To Do Type 5
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 4–49 Close To do's Algorithm: Sample Algorithm

Algorithm Name	C1-ARSCUSTS
Parameters	**Name:** To Do Type1 **Value:** C1-ANA1 **Name:** To Do Type2 **Value:** C1-ANA2 **Name:** To Do Type3 **Value:** **Name:** To Do Type4 **Value:** **Name:** To Do Type5 **Value:**

4.28 Validations for Mandatory Characteristics: C1-CHARVALS

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

Table 4–50 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	**Name:** ReferenceCharacteristicsValue **Required (Yes/No):** Y **Description:** Reference Characteristics Value **Name:** ReferenceCharacteristic **Required (Yes/No):** Y **Description:** Reference Characteristic Type **Name:** CaseCharacteristics1 **Required (Yes/No):** N **Description:** Case Characteristics **Name:** CaseCharacteristics2 **Required (Yes/No):** N **Description:** Case Characteristics **Name:** CaseCharacteristics3 **Required (Yes/No):** N **Description:** Case Characteristics **Name:** CaseCharacteristics4 **Required (Yes/No):** N **Description:** Case Characteristics **Name:** CaseCharacteristics5 **Required (Yes/No):** N **Description:** Case Characteristics
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 4–51 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:**

4.29 Validations for Mandatory Characteristics: C1-CHARVALA

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

Table 4–52 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

	Name: ReferenceCharacteristic **Required (Yes/No):** Y **Description:** NA **Name:** CaseCharacteristics1 **Required (Yes/No):** N **Description:** NA **Name:** CaseCharacteristics2 **Required (Yes/No):** N **Description:** NA **Name:** CaseCharacteristics3 **Required (Yes/No):** N **Description:** NA **Name:** CaseCharacteristics4 **Required (Yes/No):** N **Description:** NA **Name:** CaseCharacteristics5 **Required (Yes/No):** N **Description:** NA
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 4–53 Validations for Mandatory Characteristics: Sample Algorithm

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

	Value: **Name:** CaseCharacteristics4 **Value:** **Name:** CaseCharacteristics5 **Value:**

4.30 Update Collateral Status in the Host: C1-UPCOLLSTX

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

Table 4–54 Update Collateral Status in the Host: C1-UPCOLLSTX

Description	Update Collateral Status in the host
Detailed Description	This process updates the collateral status in the host. The value of status to be set will be passed as parameter to the process. 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. 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.
Algorithm Entity	Case Type-Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost
Parameters	**Name:** To Do Type **Required (Yes/No):** Y **Description:** To Do Type **Name:** Collateral Status **Required (Yes/No):** Y **Description:** Collateral Status
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 4–55 Update Collateral Status in the Host: Sample Algorithm

Algorithm Name	C1-UPCOLLSTX
Parameters	**Name:** To Do Type **Value:** C1-TD-UC **Name:** Collateral Status **Value:** Sold

4.31 Initiate Collateral Valuation: C1-COLLVALX

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

Table 4–56 Initiate Collateral Valuation: C1-COLLVALX

Description	Initiate collateral valuation
Detailed Description	This algorithm works as follows: 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. 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. However, To-do should not be created if: ■ 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 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.
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation
Parameters	**Name:** To Do Type **Required (Yes/No):** Y **Description:** To Do Type **Name:** Days Since Closure Of Last To Do **Required (Yes/No):** Y **Description:** Days Since Closure Of Last To Do **Name:** Assessment Expiry Days **Required (Yes/No):** Y **Description:** Assessment Expiry Days
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 4–57 Initiate Collateral Valuation: Sample Algorithm

Algorithm Name	C1-COLLVALX
Parameters	**Name:** To Do Type **Value:** C1-TD-UC

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

4.32 Validation Settlement: C1-VALSET

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

Table 4–58 Validation Settlement: C1-VALSET

Description	Validation Settlement
Detailed Description	This algorithm will perform following actions: Before completing the asset repossession case, the below validations should be done for the case: ■ Collateral should have a settlement date ■ Realization status for the collateral should be Complete Transition to completed status will fail if above validations fail.
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 4–59 Validation Settlement: Sample Algorithm

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

4.33 Initiate LMI Process: C1-INITLMI

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

Table 4–60 Initiate LMI Process: C1-INITLMI

Description	Initiate LMI Process
Detailed Description	Parameters to the algorithm must be as follows: ■ For Initiate LMI Options: 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: 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.InitiateLMIP
Parameters	**Name:** Balance Threshold **Required (Yes/No):** Y **Description:** NA **Name:** LMI Case Type **Required (Yes/No):** Y **Description:** NA **Name:** Initiate LMI Options **Required (Yes/No):** Y **Description:** NA **Name:** LMI Insurer Code **Required (Yes/No):** Y **Description:** NA **Name:** No LMI Option **Required (Yes/No):** Y **Description:** NA
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 4–61 Initiate LMI Process: Sample Algorithm

Algorithm Name	C1-INITLMI
Parameters	**Name:** Balance Threshold **Value:** 1000 **Name:** LMI Case Type **Value:** C1_LMI **Name:** Initiate LMI Options **Value:** HIGH_INSUR_AMT **Name:** LMI Insurer Code **Value:** QBE **Name:** No LMI Option **Value:** ALL

4.34 PTP Active Algorithm: C1-PTPACTIVE

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

Table 4–62 PTP Active Algorithm: C1-PTPACTIVE

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

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

Table 4–63 PTP Active Algorithm: Sample Algorithm

Algorithm Name	C1-PTPKEPT
Parameters	**Name:** contactTypeForLetter **Value:** OVERDUE **Name:** contactClassForLetter **Value:** CCC **Name:** contactMethodForLetter **Value:** OTBL **Name:** contactTypeForSMS **Value:** OVERDUE **Name:** contactClassForSMS **Value:** CCC **Name:** contactMethodForSMS **Value:** OTBS

4.35 PTP Kept Algorithm: C1-PTPKEPT

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

Table 4–64 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	**Name:** contactTypeForLetter **Required (Yes/No):** No **Description:** Contact Type for Letter generation **Name:** contactClassForLetter **Required (Yes/No):** No **Description:** Contact Class for letter generation **Name:** contactMethodForLetter **Required (Yes/No):** No **Description:** Contact Method for Letter generation **Name:** contactTypeForSMS **Required (Yes/No):** No **Description:** Contact Type for SMS **Name:** contactClassForSMS **Required (Yes/No):** No **Description:** Contact Class for SMS **Name:** contactMethodForSMS **Required (Yes/No):** No **Description:** Contact Method for SMS
Detailed Design	This algorithm invokes **GenerateContactForPTP** service, which creates the contact (generate Letter or SMS) when PTP moves to **Kept** state.

Table 4–65 PTP Active Algorithm: Sample Algorithm

Algorithm Name	C1-PTPKEPT
Parameters	**Name:** contactTypeForLetter **Value:** OVERDUE **Name:** contactClassForLetter **Value:** CCC

	Name: contactMethodForLetter **Value:** OTBL **Name:** contactTypeForSMS **Value:** OVERDUE **Name:** contactClassForSMS **Value:** CCC **Name:** contactMethodForSMS **Value:** OTBS

4.36 PTP Broken Algorithm: C1-BRKPTPNGP

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

Table 4–66 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	**Name:** contactTypeForLetter **Required (Yes/No):** No **Description:** Contact Type for Letter generation **Name:** contactClassForLetter **Required (Yes/No):** No **Description:** Contact Class for letter generation **Name:** contactMethodForLetter **Required (Yes/No):** No **Description:** Contact Method for letter generation **Name:** contactTypeForSMS **Required (Yes/No):** No **Description:** Contact Class for SMS generation

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

4.37 Rule facts populating algorithm: C1-BRLSR

This section provides details of the Rule Facts Populating Algorithm: C1_BRLSR algorithm.

Table 4–67 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	**Name:** Input Key1 **Required (Yes/No):** Yes **Description:** Primary Key name of defined BO. **Name:** Input Key2 **Required (Yes/No):** No **Description:** Primary Key name of defined BO. **Name:** Input Key3 **Required (Yes/No):** No **Description:** Primary Key name of defined BO. **Name:** Input Key4 **Required (Yes/No):** No **Description:** Primary Key name of defined BO. **Name:** Input Key5

	Required (Yes/No): No **Description:** Primary Key name of defined BO.
Parameters	**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. **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. **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. **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. **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.
Parameters	**Name:** Bo Fields **Required (Yes/No):** Yes **Description:** Comma separated BO fields of defined BO names. **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. **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').
Detailed Design	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.

	Also, pre-populated facts are appended to these values, if provided from external algorithm. These populated facts will act as input to defined rule through soft parameter.

Sample Algorithm

Table 4–68 Sample Algorithm

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

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

4.38 Borrower Centric Case Lifecycle

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

Table 4–69 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 4–70 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-BRWRTRNDF algorithm.

Table 4–71 Borrower Level : C1-BRWRTRNDF

Algorithm Name	C1-BRWRTRNDF
Parameters	**Name:** Wait Days **Value:** 0

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

Table 4–72 Borrower Level : C1-BRWRSW_N

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

4.39 Update Collection Address on Borrower Panel

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

Table 4–73 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 4–74 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.

4.40 Update Collection Contact Point

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

Table 4–75 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.

4.41 Bankruptcy Process

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

Table 4–76 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	Check if any active case is present of a given case category or case type on the customer. Processing steps are as below: 1. If only Case Category is specified check if any active case is running on the customer whose 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 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	**Name:** Case Category **Required (Yes/No):** Yes **Description:** Case Category

	Name: Case Type **Required (Yes/No):** Yes **Description:** Case Type **Name:** Consider Enterprise Id **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 Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullNDAcc algorithm.

Table 4–77 Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullNDAcc

Description	Pull all the non delinquent accounts of the customer into collections- Enter Processing
Detailed Description	Processing steps are as below: ■ 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	**Name:** Account Relationships (MC,FO,ALL) **Required (Yes/No):** Yes **Description:** Account Relationships **Name:** Consider Enterprise Id (Yes/No)

	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 4–78 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: ■ 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-ExcAccDlr algorithm.

Table 4–79 Exclude all the associated accounts from Dialer- Enter Processing: C1-ExcAccDlr

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: ■ 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.BankruptcyExcludeAccDlr
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-IniCltVal algorithm.

Table 4–80 Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing: C1-IniCltVal

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	**Name:** Valuation Expiry days **Required (Yes/No):** Yes **Description:** Valuation Expiry days **Name:** Collateral Valuation Task **Required (Yes/No):** Yes **Description:** Collateral Valuation Task **Name:** Administration Queue **Required (Yes/No):** Yes **Description:** Administration Queue **Name:** Exclude Collateral Types

	Required (Yes/No): No **Description:** Exclude Collateral Types **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 Monitor if any of the associated account need to be charged off and monitor delinquency- Monitoring: C1-MTRCRGDQY algorithm.

Table 4–81 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	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. 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. If Secured Accounts = Yes than associated accounts with Secured Switch = Y should also be considered. Monitor Delinquency = "Y" or "N" Monitor Charge Off = "Y" or "N" Secured Accounts = "Y" or "N" 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.BankruptcyMonitorChargeOffDelinquency
Parameters	**Name:** Monitor Delinquency **Required (Yes/No):** Yes **Description:** Monitor Delinquency **Name:** Monitor Charge Off **Required (Yes/No):** Yes **Description:** Monitor Charge Off

	Name: Charge Off Threshold D P D **Required (Yes/No):** Yes **Description:** Charge Off Threshold D P D **Name:** Secured Accounts **Required (Yes/No):** Yes **Description:** Secured Accounts **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 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG algorithm.

Table 4–82 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG

Description	Notify the Bankruptcy Specialist on Hearing Dates- Monitoring
Detailed Description	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 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
Algorithm Entity	Case Type - Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitor341Hearing
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 Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring: C1-MTRPYMPLN algorithm.

Table 4–83 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	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. The PTP Type specified in the parameter should be considered Notification: <PTP Type> broken for account <Account Number>. Days since plan broken <Days Since PTP Broken>. Set Display Date of the case to current business date. Values of Validation Date: POSTING DATE, SYSTEM DATE
Algorithm Entity	Case Type -Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorPaymentPlan
Parameters	**Name:**P T P Type **Required (Yes/No):** Yes **Description:** P T P Type **Name:**Days Since P T P Broken **Required (Yes/No):** Yes **Description:** Days Since P T P Broken **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 Notify the Bankruptcy Specialist if the Liquidation reaches a specific status-Monitoring: C1-MNTRASLQD algorithm.

Table 4–84 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	Notify the Bankruptcy Specialist if the Liquidation reaches a specific status. If for any of the associated account if the liquidation case reaches a specific status than create a notification for the Bankruptcy Specialist. Notification: "Liquidation for Account <Account Number>; Collateral <Collateral Code> has reached status

	<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 4–85 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.Bankruptcy MonitorHearingDate
Parameters	**Name:**Validation Date **Required (Yes/No):** Yes **Description:** Validation Date
Detailed Design	This is a reference implementation of Monitring 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 4–86 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	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 If Bankruptcy Chapter = Chapter 7 Then Transition to Manage Chapter 7 Bankruptcy Status If Bankruptcy Chapter = Chapter 13 Then Transition to Manage Chapter 13 Bankruptcy Status If Bankruptcy Chapter = Chapter other than 7 or 13 Then Transition to Other Bankruptcy Status Bankruptcy Chapter Field = "FC" or "CC Where "FC" = Filing Chapter and "CC"=Convert to chapter
Algorithm Entity	Result Type -Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.DetermineBankruptcyTreatment
Parameters	**Name:**Bankruptcy Chapter Field **Required (Yes/No):** Yes **Description:** Bankruptcy Chapter Field **Name:**Manage Chapter7 Bankruptcy Status **Required (Yes/No):** Yes **Description:** Manage Chapter7 Bankruptcy Status **Name:**Manage Chapter13 Bankruptcy Status **Required (Yes/No):** Yes **Description:** Manage Chapter13 Bankruptcy Status **Name:**Other Bankruptcy Status **Required (Yes/No):** Yes **Description:** Other Bankruptcy Status
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 4–87 Validate if appropriate Case Details have been entered by the user- Post Processing C1-VLDBCDATA

Description	Validate if appropriate Case Details have been entered by the user- Post Processing
Detailed Description	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: ■ ENTITY_NAME,PHONE,EMAIL,FAX_NUMBER,CONTACT_POINT_NAME,CONTACT_POINT_PHONE_NUM,CONTACT_POINT_EMAIL,CONTACT_POINT_FAX Panel Name : bankruptcyProcessingInfoPanel Corresponding Panel Fields : ■ HEARING_DATE,HEARING_LOCATION,LENDER_COLL_VAL_DATE,LENDER_COLL_VAL,DISCHARGE_DATE,DISMISSED_DATE,CHAPTER_CODE,COVERSION_REMARKS,CONVERSION_DATE,HEARING_ADD_INFO Panel Name : bankruptcyDebtorAttorneyPanel Corresponding Panel Fields: ■ FIRM_NAME,PHONE,ENTITY_NAME,DEBTOR_ADDRESS Panel Name : bankruptcyFilingInfoPanel Corresponding Panel Fields : ■ DATE_OF_BNKPT_CASE_FILE,BNKPT_CASE_NUM,COURT,CHAPTER Panel Name : bankruptcyConfirmPlanInformationPanel Corresponding Panel Fields : ■ RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT Panel Name : bankruptcyDebtorProposedPlanInfoPanel Corresponding Panel Fields : ■ RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT,OBJECTION_DATE,OBJECTION_OUTCOME,HEARING_DATE Panel Name : bankruptcyLegalCounselInfoPanel Corresponding Panel Fields : ■ ASSIGNED_DATE,COUNSEL_NAME,CONTACT_POINT_NAME,EMAIL,PHONE,ALTERNATE_PHONE
Algorith	Result Type -Post Processing

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

	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 4–88 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 4–89 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 4–90 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.JointBnkptcyAssociateCust
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 4–91 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 : 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.BkptcyPayPlanCreation
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 4–92 Close Pay Plan for a Bankruptcy Case - Exit Processing: C1-CLDATP

Description	Algorithm to close Arrearage Pay Plan for a Bankruptcy Case
Detailed Description	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 : 1. CI_BKPTCY_PAY_PLAN_INFO 2. CI_BKPTCY_PAY_PLAN_DTLS 3. CI_BKPTCY_PAY_PLAN_SCHED 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.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 4–93 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	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 : Arrearage Plan Threshold Days Arrearage Plan Threshold Amount Confirmed Plan Threshold Days Confirmed Plan Threshold Amount Notification Date Type 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
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BanckruptcyMonitorArreragePlanNotification
Parameters	**Name**: Arrearage Plan Threshold Days **Required (Yes/No)**: Yes **Description**: Arrearage Plan Threshold Days **Name**: Arrearage Plan Threshold Amount **Required (Yes/No)**: Yes **Description**: Arrearage Plan Threshold Amount **Name**: Confirmed Plan Threshold Days **Required (Yes/No)**: Yes **Description**: Confirmed Plan Threshold Days **Name**: Confirmed Plan Threshold Amount **Required (Yes/No)**: Yes **Description**:Confirmed Plan Threshold Amount **Name**: Notification Date Type **Required (Yes/No)**: Yes **Description**:Notification Date Type
Detailed Design	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 : 1. Arrearage Plan Threshold Days 2. Arrearage Plan Threshold Amount 3. Confirmed Plan Threshold Days 4. Confirmed Plan Threshold Amount 5. Notification Date Type 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

Table 4–94 Set or Reset Account level Warning Indicator for Bankruptcy - Enter Processing: C1-SETWI

Description	Set or Reset Account level Warning Indicator for Bankruptcy
Detaile	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,...>

4.42 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 4–95 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.

4.43 Hardship - Associate Accounts of Main Customer

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

Table 4–96 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.

4.44 Early Collection

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

Table 4–97 Transition to Contact Statuses - Monitoring C1-ECIC

Description	Transition to Contact Statuses - Monitoring
Detailed Description	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. Transition to contact state if First Contact Date has reached: If First Contact Date has reached (based on the parameters below) or Account is Direct Debit and Immediate Transition if Direct Debit = Yes/No Transition to Contact RM status if Relationship Manager exists and Contact RM status has been specified . Transition to Contact Alternate status if Contact Alternate Flag = Y and Contact

	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
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateContact
Parameters	**Name**: First Contact Calculation Parameter **Required (Yes/No)**: No **Description**: First Contact Calculation Parameter **Name**: Number Of Days For First Contact **Required (Yes/No)**: No **Description**: Number Of Days For First Contact **Name**: Contact RM Status **Required (Yes/No)**: No **Description**: Contact RM Status **Name**: Contact Alternate Status **Required (Yes/No)**: No **Description**: Contact Alternate Status **Name**: Contact Status **Required (Yes/No)**: No **Description**: Contact Status **Name**: Immediate Transition if Direct Debit: Yes/No **Required (Yes/No)**: No **Description**: Immediate Transition if Direct Debit: Yes/No **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 4–98 Park Small Balance Accounts - Monitoring C1-ECPSBA

Description	Park accounts with small balances to a separate status - Monitoring
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 Park Small Balance Accounts. Park accounts with small balances to a separate status so that no contacts are initiated for the account. If Net Arrear Amount <= Small Balance Threshold And Net Arrear Amount > 0 Then transition to small balance status. Net Arrear Amount = (Overdue Amount - Unclear Amount) If Use Overdue Amount = Yes then use Overdue Amount instead of Net Arrear Amount in the calculations. Set Re-Allocation Switch = Y for the case post case transition. Possible Values : Use Overdue Amount : Y,N
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ParkSmallBalanceAccounts
Parameters	**Name**: Small Balance Threshold **Required (Yes/No)**: No **Description**: Small Balance Threshold **Name**: Small Balance Status **Required (Yes/No)**: No **Description**: Small Balance 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 4–99 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	If no contact points exists then move the case to Skip Tracing status Check if one of the Contact Points as specified in the parameters exists for any of the account holder. If no contact point exists than move the case to Skip Tracing Status.

	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	**Name**: Contact Points **Required (Yes/No)**: No **Description**: Contact Points **Name**: Skip Tracing Status **Required (Yes/No)**: No **Description**: Skip Tracing Status
Detailed Design	This is a reference implementation Enter Processing algorithm. Customization team can utilize this hook.

Table 4–100 Initiate Skip Tracking - No Telephone Number- Monitoring C1-ECTTSS

Description	Transition to Suspended status based on Account and Party Risk Indicators - Monitoring
Detailed Description	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. 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. 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.
Algorithm Entity	Case Type-Auto Transition
Program Type	Java

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

Table 4–101 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

	And the number of successful contacts linked to the case in last X number of days >= Contact Cap Hold the case for Y number of days with the given Hold Reason. Logic for considering successful contacts: All contacts with given contact methods that have Authentication Status = Green Possible Values for Validation Date {SYSTEMDATE,POSTINGDATE}
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	**Name**: Contact Cap **Required (Yes/No)**: No **Description**: Contact Cap **Name**: Contact Cap Duration (X) **Required (Yes/No)**: No **Description**: Contact Cap Duration (X) **Name**: Contact Hold Days (Y) **Required (Yes/No)**: No **Description**: Contact Hold Days (Y) **Name**: Contact Methods **Required (Yes/No)**: No **Description**: Contact Methods **Name**: Hold Reason **Required (Yes/No)**: No **Description**: Hold Reason **Name**: Validation Date **Required (Yes/No)**: No **Description**: Validation Date
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

Table 4–102 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

	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
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ScheduleContact
Parameters	**Name**: Contact Intensity **Required (Yes/No)**: No **Description**: Contact Intensity **Name**: Contact Methods **Required (Yes/No)**: No **Description**: Contact Methods **Name**: Validation Date **Required (Yes/No)**: No **Description**: Validation Date
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

Table 4–103 Initiate Skip Tracing - Wrong Telephone Number- Monitoring C1-ECISTITN

Description	Initiate Skip Tracing - Wrong Telephone Number- Monitoring
Detailed Description	Transition to skip review if 'X' number of consecutive failed contacts ■ If last X number of consecutive contacts has been unsuccessful, transition to Skip Tracing Status. 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
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracingInvalidTelNumber

Parameters	**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 4–104 Transition to Under Resolution Status- Monitoring C1-ECTTURS

Description	Transition to under resolution status.
Detailed Description	Transition to under resolution status if Net Arrear Amount <=0 ▪ 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 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
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToUnderResolutionStatus
Parameters	**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 4–105 Resume Contact From Under Resolution- Monitoring C1-ECRCFUR

Description	Resume Contact From Under Resolution- Monitoring
Detailed Description	Resume Contact From Under Resolution Status - If there is no more active PTP on the account and - If the Net Arrear Amount > 0 Than 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. Net Arrear Amount = (Overdue Amount - Unclear Amount) Use Overdue Amount can be Y/N or Yes/No
Algorithm Entity	Case Type-Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactFromUnderResolution
Parameters	**Name**: Contact Status **Required (Yes/No)**: No **Description**: Contact Status **Name**: Contact RM Status **Required (Yes/No)**: No **Description**: Contact RM Status **Name**: Contact Alternate Status **Required (Yes/No)**: No **Description**: Contact Alternate 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 4–106 Resume Contact from Small Balance- Monitoring C1-ECRCSB

Descriptio n	Resume Contact from Small Balance- Monitoring
Detailed Descriptio n	.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. 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
Parameter s	**Name**: Contact Status **Required (Yes/No)**: No **Description**: Contact Status **Name**: Contact RM Status **Required (Yes/No)**: No **Description**: Contact RM Status **Name**: Contact Alternate Status **Required (Yes/No)**: No **Description**: Contact Alternate Status **Name**: Use Overdue Amount **Required (Yes/No)**: No **Description**: Use Overdue Amount **Name**: Small Balance Threshold

	(Yes/No): No **Description**: Small Balance Threshold
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

Table 4–107 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	**Name**: Contact Intensity Rule **Required (Yes/No)**: No **Description**: Contact Intensity Rule **Name**: Validation Date **Required (Yes/No)**: No **Description**: Validation Date
Detailed Design	This is a reference implementation Monitoring algorithm. Customization team can utilize this hook.

Table 4–108 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	Generic Result Post Processing Algorithm for Case Transition and Task Creation. 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. ▪ 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 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" 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 : ■ 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	**Name**: Case Status **Required (Yes/No)**: No **Description**: Case Status

	Name: Valid Current Status **Required (Yes/No)**: No **Description**: Valid Current Status **Name**: Task Type **Required (Yes/No)**: No **Description**: Task Type **Name**: Queue **Required (Yes/No)**: No **Description**: Queue **Name**: Re-Allocate Switch **Required (Yes/No)**: No **Description**: Re-Allocate Switch **Name**: Copy Characteristics to Case **Required (Yes/No)**: No **Description**: Copy Characteristics to Case **Name**: Event Name **Required (Yes/No)**: No **Description**: Event Name
Parameters (Cont.)	**Name**: Action Flag **Required (Yes/No)**: No **Description**: Action Flag **Name**: Contact Alternate Case Status **Required (Yes/No)**: No **Description**: Contact Alternate Case Status
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

Table 4–109 Refer to Supervisor- Result Type - Post Processing C1-ECRTS

Description	Refer to Supervisor - Result Type - Post Processing
Detailed Description	Supervisor Referral Algorithm ■ 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	**Name**: Valid Current Status **Required (Yes/No)**: No **Description**: Valid Current Status **Name**: Case Status **Required (Yes/No)**: No **Description**: Case Status **Name**: Task Type **Required (Yes/No)**: Yes **Description**: Task Type **Name**: Re-Allocate **Required (Yes/No)**: Yes **Description**: Re-Allocate
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

Table 4–110 Resume Collections- Result Type - Post Processing C1-RESCOLL

Description	Resume Collections- Result Type - Post Processing
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 Resume Collections. 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 = Yes if Re-Allocate = Y Re-Allocate can be Y/N
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeCollectionsPostProcessingAlgo
Parameters	**Name**: Contact Status **Required (Yes/No)**: No **Description**: Contact Status **Name**: Contact RM Status **Required (Yes/No)**: No **Description**: Contact RM Status **Name**: Contact Alternate Status **Required (Yes/No)**: No **Description**: Contact Alternate Status **Name**: Re-Allocate **Required (Yes/No)**: No **Description**: Re-Allocate
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

Table 4–111 Create case on Follow up- Result Type - Post Processing C1-CRETCSFL

Description	Create case on Follow up - Post Processing
Detailed Description	Create Required Case on Follow Up If Account Level Case Type creates case on account, If Customer level Case Type creates case on the main customer of the account. Queue to which the case should be allocated if provided else the case should remain unallocated with Re-Allocation Switch as Y
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseCreationonFollowupPostProcessingAlgo
Parameters	**Name**: Case Type **Required (Yes/No)**: Yes

	Description: Case Type **Name**: Queue Type **Required (Yes/No)**: No **Description**: Queue Type
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

Table 4–112 Hold Case- Result Type - Post Processing C1-HOLDCASE

Description	Hold Case - Post Processing
Detailed Description	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. And 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. Validation Date can be SYSTEMDATE or POSTINGDATE
Algorithm Entity	Result Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.HoldCasePostProcessingAlgo
Parameters	**Name**: Hold Period Characteristic Type **Required (Yes/No)**: No **Description**: Hold Period Characteristic Type **Name**: Hold Period **Required (Yes/No)**: No **Description**: Hold Period **Name**: Hold Reason Characteristic Type **Required (Yes/No)**: No **Description**: Hold Reason Characteristic Type **Name**: Hold Reason **Required (Yes/No)**: No **Description**: Hold Reason **Name**: Validation Date **Required (Yes/No)**: Yes

	Description: Validation Date
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

Table 4–113 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	Set Case Characteristics to specific values provided in algorithm parameters. On entering the value the corresponding characteristic validation algorithm should be triggered. If type is mentioned but value is not than the char type needs to be made blank.
Algorithm Entity	Case Status - Enter Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateCaseData
Parameters	**Name**: Char Type - 1 **Required (Yes/No)**: No **Description**: Char Type - 1 **Name**: Char Type - 2 **Required (Yes/No)**: No **Description**: Char Type - 2 **Name**: Char Type - 3 **Required (Yes/No)**: No **Description**: Char Type - 3 **Name**: Char Type - 4 **Required (Yes/No)**: No **Description**: Char Type - 4 **Name**: Char Type - 5 **Required (Yes/No)**: No **Description**: Char Type - 5 **Name**: Char Value - 1 **Required (Yes/No)**: No **Description**: Char Value - 1 **Name**: Char Value - 2 **Required (Yes/No)**: No **Description**: Char Value - 2

	Name: Char Value - 3 **Required (Yes/No)**: No **Description**: Char Value - 3 **Name**: Char Value - 4 **Required (Yes/No)**: No **Description**: Char Value - 4 **Name**: Char Value - 5 **Required (Yes/No)**: No **Description**: Char Value - 5
Detailed Design	This is a reference implementation of Pre Processing algorithm. Customization team can utilize this hook.

Table 4–114 This algorithm will transition the case status to the Suspension status if Cease and Desist = Y C1-CSETRANS

Descriptio n	This algorithm will transition the case status to the Suspension status if Cease and Desist = Y
Detailed Descriptio n	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
Parameter s	**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 4–115 Algorithm is used for scheduling call C1-SCHCALL

Description	Algorithm is used for scheduling call
Detailed Description	This algorithm is used to fulfil request by customer to collector for calling at specific time. ■ 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	This algorithm is used to fulfil request by customer to collector for calling at specific time. ■ 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 4–116 Reset WI in the host C1-RESETWISCHCALL

Description	Reset WI in the host
Detailed Description	.This algorithm resets WI in the host. ■ 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	**Name**: Account Warning Indicator **Required (Yes/No)**: Yes **Description**: Account Warning Indicator
Detailed Design	This algorithm resets WI in the host.

	- Call the Host Account Warning Indicator Service to set the WI mentioned in the parameter

4.45 Asset Repossession

Table 4–117 Validate Collateral - Enter Validation C1-VALDCOLL

Description	Validate Collateral - Enter Status Validation
Detailed Description	The input collateral is associated with the account on which the repossession case is being created. 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. The collateral status is not 'Sold'. Date of Sale is blank. Error Message: "Repossession cannot be initiated on the collateral. Please check if The collateral type is supported in the repossession process. The collateral in not already sold or under repossession." Collateral Type: It should be allow any collateral type as a parameter.(For eg: AUTOMOBILE,PROPERTY etc) Collateral Category :Category of collateral(For eg: Vehicle) Both parameter accept comma separated values. CI_COLLATERAL table have both columns.
Algorithm Entity	Case Status - Enter Status Validation
Program Type	Java
Program Name	om.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateral
Parameters	**Name**: Collateral Type **Required (Yes/No)**: No **Description**: Collateral Type **Name**: Collateral Category **Required (Yes/No)**: No **Description**: Collateral Category
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 4–118 Validate Demand Letter and Acceleration Letter - Enter Validation C1-VALIDDLAL

Descri	Validate if Demand Letter and Acceleration letter have been sent - Enter Status Validation

ption	
Detailed Description	If DL Template Code has been mentioned validate if Demand Letter has been sent in last X days. If AL Template Code has been mentioned validate if Acceleration Letter has been sent in last X days. If X Days is not specified just check if the letters have been sent on the account. 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. Parameter Description as follows : 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	**Name**: Demand Letter Template Code **Required (Yes/No)**: No **Description**: Demand Letter Template Code **CodeName**: Acceleration Letter Template Code **Required (Yes/No)**: No **Description**: Acceleration Letter Template Code **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 **Name**: Only PrimaryAccount Switch **Required (Yes/No)**: No **Description**: Only PrimaryAccount Switch

	Name: Validation Date **Required (Yes/No)**: No **Description**: Validation Date
Detailed Design	Validate if Demand Letter and Acceleration letter have been sent

Table 4–119 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 Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AssociateCustAssRepo
Parameters	No Parameters
Detailed Design	Associate all financial owners on the associated accounts to the Repossession case.

Table 4–120 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	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)
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 4–121 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	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 : 1. Demand Letter Template Code - Demand Letter Template Code 2. Acceleration Letter Template Code - Acceleration Letter Template Code 3. Reposession 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	**Name**: Demand Letter Template Code **Required (Yes/No)**: No **Description**: Demand Letter Template Code **Name**: Acceleration Template Code **Required (Yes/No)**: No **Description**: Acceleration Template Code

	Name: Reposession Referred Status **Required (Yes/No)**: No **Description**: Reposession Referred Status **Name**: Primary Account Sw **Required (Yes/No)**: No **Description**: Primary Account Sw **Name**: Validation Date **Required (Yes/No)**: No **Description**: Validation Date
Detailed Design	Monitor if Demand letter and Acceleration letter have been sent on the account.

Table 4–122 Auto Approval Check for Repossession C1-REPOAPRV

Description	Auto Approval Check for Repossession
Detailed Description	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. Below facts are used: ■ 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	**Name**: Auto Approval Rule **Required (Yes/No)**: Yes **Description**: Auto Approval Rule **Name**: Approved Status **Required (Yes/No)**: Yes **Description**: Approved Status **Name**: Task Type **Required (Yes/No)**: Yes **Description**: Task Type **Name**: Queue **Required (Yes/No)**: Yes **Description**: Queue
Detailed Design	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.

Table 4–123 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	**Name**: Voluntary Repossession Reason **Required (Yes/No)**: Yes **Description**: Voluntary Repossession Reason **Name**: Voluntary Repossession Status **Required (Yes/No)**: Yes **Description**: Voluntary Repossession Status

	Name: Normal Repossession Status **Required (Yes/No)**: Yes **Description**: Normal Repossession Status
Detailed Design	If Repossession Reason = Voluntary Repossession transition to Repossession In Progress - Voluntary Surrender else transition to Repossession in Progress

Table 4–124 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	**Name**: Service Type **Required (Yes/No)**: Yes **Description**: Service Type **Name**: Task Type **Required (Yes/No)**: Yes **Description**: Task Type **Name**: Queue **Required (Yes/No):** Yes **Description**: Queue
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 4–125 Notify Repossession Specialist on Task Completion C1-NOTRSTSK

Description	Notify Repossession Specialist on Task Completion
Detailed Description	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.
Algorithm Entity	To Do Type - To Do Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.NotifyOnTaskCompletion
Parameters	Name: Display Date Required (Yes/No): Yes Description: Display Date
Detailed Design	Create Notification.

Table 4–126 Automatic sending of Redemption letters C1-REDEMPLTR

Description	Automatic sending of Redemption letters
Detailed Description	For each of the accounts associated to the repossession case send the Redemption letter (create customer contact of given template code) If Only Primary Account = Yes then send letter only on the primary account. Parameter Description as follows : 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	**Name**: Contact Class **Required (Yes/No)**: Yes **Description**: Contact Class **Name**: Contact Type **Required (Yes/No)**: Yes

	Description: Contact Type **Name**: Primary Account Sw **Required (Yes/No)**: No **Description**: Primary Account Sw **Name**: Validation Date **Required (Yes/No)**: Yes **Description**: Validation Date
Detailed Design	For each of the accounts associated to the repossession case send the Redemption letter (create customer contact of given template code) If Only Primary Account = Yes then send letter only on the primary account.

Table 4–127 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 4–128 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	Result Type - Post Processing Algorithm

Entity	
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 **Required (Yes/No)**: No **Description**: Dynamic Panel Two Fields **Name**: Dynamic Panel Three **Required (Yes/No)**: No **Description**: Dynamic Panel Three **Name**: Dynamic Panel Three Fields **Required (Yes/No)**: No **Description**: Dynamic Panel Three Fields **Name**: Dynamic Panel Four **Required (Yes/No)**: No **Description**: Dynamic Panel Four **Name**: Dynamic Panel Four Fields **Required (Yes/No)**: No **Description**: Dynamic Panel Four Fields
Parameters(Contd.)	**Name**: Dynamic Panel Five **Required (Yes/No)**: No **Description**: Dynamic Panel Five **Name**: Dynamic Panel Five Fields **Required (Yes/No)**: No **Description**: Dynamic Panel Five Fields **Name**: caseCharacteristcs **Required (Yes/No)**: No

	Description: Case Characteristics **Name:** Previous Status Transition **Required (Yes/No):** Yes **Description:** Previous Status Transition
Detailed Design	Validate if the Dynamic Panel Data Elements and Case Characteristics mentioned in the parameters have some values for the case.

Table 4–129 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 4–130 Send Repossession Alert to Vendor C1-REPOASAL

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

	Required (Yes/No): No **Description**: Contact Type **Name**: Service Type **Required (Yes/No)**: No **Description**: Service Type **Name**: Validation Date **Required (Yes/No)**: Yes **Description**: Validation Date
Detailed Design	Generate and send the email to the email id of the contact person associated to the service type mentioned in the parameter. Email of specified template code will be sent. The algorithm will generate the contact as well as initiate contact processing

Table 4–131 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 4–132 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 4–133 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	**Name**: Case Status **Required (Yes/No)**: No **Description**: Case Status **Name**: Approval Task Type **Required (Yes/No)**: No **Description**: Approval Task Type **Name**: Validation Date

	Required (Yes/No): Yes **Description**: Validation Date
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 4–134 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. This is mandatory parameter. 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 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. Stored Date Format is a mandatory parameter whereas Date Format2 is not. Date Format2 is given for future requirement, if any.
Algorithm Entity	Characteristic Type - Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDate
Parameters	**Name**: Validation Date **Required (Yes/No)**: Yes **Description**: Validation Date **Name**: Stored Date Format **Required (Yes/No)**: Yes **Description**: Stored Date Format **Name**: Date Format2 **Required (Yes/No)**: No **Description**: Date Format2

Detailed Design	This algorithm is used to validate format enter by user for result characteristics during taking follow up. 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. 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 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. Stored Date Format is a mandatory parameter whereas Date Format2 is not. Date Format2 is given for future requirement, if any.

Table 4–135 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.

4.46 Miscellaneous

Table 4–136 Update Review Date for associated accounts C1-UPDRVWDT

Description	Update Review Date for associated accounts
Detailed Description	For all accounts associated with the case this process will update the review date. Below parameters should be available for the process Update Type ■ Set Review Date - This will set the Review Date for the account ■ Remove Review Date - This will remove the Review date from the account Days Offset - Applicable only of Update Type = Set. System will set the review date as Current business days + Offset days. Override Flag ■ 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	**Name**: Override Flag Value **Required (Yes/No)**: Yes **Description**: Override Flag Value **Name**: Days Offset **Required (Yes/No)**: Yes **Description**: Days Offset **Name**: Update Type **Required (Yes/No)**: Yes **Description**: Update Type

Table 4–137 Case Monitoring CS-MONITOR

Description	Case Monitoring
Detailed Description	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. 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 par configuration:

	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	**Name**: Next Status **Required (Yes/No)**: No **Description**: Next Status **Name**: Work List **Required (Yes/No)**: No **Description**: Work List **Name**: Reallocate Switch **Required (Yes/No)**: No **Description**: Reallocate Switch **Name**: To Do Type **Required (Yes/No)**: No **Description**: To Do Type **Name**: No Of Days **Required (Yes/No)**: No **Description**: No Of Days

Table 4–138 Update warning indicator for the customer C1-UPDWARN

Description	Update warning indicator for the customer
Detailed Description	This process will update the warning indicator for the customer ■ 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 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)
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateWarningIndicator
Parameters	**Name**: Association Type **Required (Yes/No)**: Yes **Description**: Override Association Type **Name**: Warning Indicator Type **Required (Yes/No)**: Yes **Description**: Warning Indicator Type **Name**: Update Type **Required (Yes/No)**: Yes **Description**: Update Type

Table 4–139 Transition to Default Next Status C1-TRAN-STAT

Description	Transition to Default Next Status
Detailed Description	This is a common algorithm that will automatically transition the case to the next status. Following are the parameters : 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	**Name**: Next Status **Required (Yes/No)**: No **Description**: Next Status **Name**: Next Transition Condition **Required (Yes/No)**: No **Description**: Next Transition Condition

Table 4–140 Set Account Warning Indicator C1-ACTCSWGID

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	**Name**: Account Warning Indicator **Required (Yes/No)**: Yes **Description**: Account Warning Indicator Code
Detailed Design	Set Account Warning Indicator for host accounts

4.47 Derived Field

Table 4–141 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

4.48 Task

Table 4–142 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 4–143 Validate Task Completion C1-VALTASKEX

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	ccom.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 4–144 Automatic Task Creation when case enters a particular status C1-CREATTASK

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

	Description: Queue **Name**: Task Type3 **Required (Yes/No)**: No **Description**: Task Type **Name**: Queue3 **Required (Yes/No)**: No **Description**: Queue **Name**: Task Type4 **Required (Yes/No)**: No **Description**: Task Type **Name**: Queue4 **Required (Yes/No)**: No **Description**: Queue
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

4.49 Event Manager

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

Table 4–145 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

4.50 Legal Vendor Allocation C1-LGLVNDRAL

Table 4–146 Legal vendor Allocation C1-LGLVNDRAL

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

Table 4–147 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	Hardship Expiry Date Characteristic **Required (Yes/No)**: Yes **Description**: Hardship Expiry Date Characteristic Type Code **Name**: Extended Expiry Date Characteristic **Required (Yes/No)**: Yes **Description**: Extended Expiry Date Characteristic Type Code
Detailed Description	Validate Extended Expiry Date

4.51 Extend Expiry Date C1-EXT-EXPDT

Table 4–148 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.ExtendExpiryDate
Parameters	**Name**: Extended Expiry Date Char Type **Required (Yes/No)**: Yes **Description**: Extended Expiry Date Char Type Code **Name**: Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition

4.52 Capture Case Status Update Date/Time C1-CASE-STAT

Table 4–149 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	**Name**: Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition Code **Name**: Xpath to Date Element **Required (Yes/No)**: Yes **Description**: Xpath to Date Element Code

4.53 Create To Do C1-TO-DO

Table 4–150 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	**Name**: To Do Type **Required (Yes/No)**: Yes **Description**: To Do Type Code **Name**: Message Category **Required (Yes/No)**: No **Description**: Message Category Code **Name**: Message Number **Required (Yes/No)**: No **Description**: Message Number Code **Name**: Characteristic Type For Log Entry **Required (Yes/No)**: Yes **Description**: Characteristic Type For Log Entry Code **Name**: Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition Code

4.54 Check customer eligibility C1-CHKCUST

Table 4–151 Check customer eligibility C1-CHKCUST

Description	Check customer eligibility
Detailed Description	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: Primary entity for the case is account. Based on ownership type parameter for the process, system should consider the customers for eligibility check. ■ If ownership type parameter is set to "financial owner" • 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" ● System should call the rule engine to check for primary customers eligibility. Customers' facts should be used for rule engine decision. For each call ■ 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. ● Check Validation failure option ○ 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. Parameters: ■ Ownership Type - Ownership type can be FINANCIAL_OWNER(Financial Owner) or PRIMARY(Primary Owner). ■ 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.CheckCustomerEligiblity
Parameters	**Name**: Cancel Reason Char Value **Required (Yes/No)**: No **Description**: Cancel Reason Char Value **Name**: Cancel Reason Char Type

	Required (Yes/No): No **Description**: Cancel Reason Char Type **Name**: Validation Failure Transition Status **Required (Yes/No)**: No **Description**: Validation Failure Transition Status **Name**: Validation Failure Option **Required (Yes/No)**: Yes **Description**: Validation Failure Option **Name**: Rule Id **Required (Yes/No)**: Yes **Description**: Rule Id **Name**: Ownership Type **Required (Yes/No)**: Yes **Description**: Ownership Type

4.55 Capture Hardship Approval Date C1-HARAP-DT

Table 4–152 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	**Name**: Xpathto Date Element **Required (Yes/No)**: Yes **Description**: Xpathto Date Element **Name**: Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition

4.56 Algorithm that will interface with Rule Engine. C1-RULEADAPT

Table 4–153 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

4.57 Perform Validation for Collateral C1-VRFYCOL

Table 4–154 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

4.58 Check target account eligibility C1-CHKTRGT

Table 4–155 Check target account eligibility C1-CHKTRGT

Description	Check target account eligibility
Detailed Description	■ System should call the rule engine for eligibility check. Output of rule engine will be "Validation Status" • If validation status is "Success" ○ 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" ○ 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". • 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.CheckTargetAccountEligiblity
Parameters	**Name**: Cancel Reason Char Value **Required (Yes/No)**: No **Description**: Cancel Reason Char Value **Name**: Cancel Reason Char Type **Required (Yes/No)**: No **Description**: Cancel Reason Char Type Code **Name**: Validation Failure Transition Status **Required (Yes/No)**: Yes **Description**: Validation Failure Transition Status Code **Name**: Minimum Residual Amount **Required (Yes/No)**: Yes **Description**: Minimum Residual Amount **Name**: Rule Id **Required (Yes/No)**: Yes **Description**: Rule Id

4.59 Approval check for set-off transaction C1-ROSOAPPR

Table 4–156 Approval check for set-off transaction C1-ROSOAPPR

Description	Approval check for set-off transaction
Detailed Description	This process will check if approval is required for a set-off transaction. Approval will be required if ■ 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 Based on whether approval is required or not, transition the case to a status as set in the parameters of the process.
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 **Required (Yes/No)**: Yes **Description**: Total Debit Amount Threshold **Name**: No Approval Status **Required (Yes/No)**: Yes **Description**: No Approval Status **Name**: Approval Required Status **Required (Yes/No)**: Yes **Description**: Approval Required Status **Name**: Accrual Status Flag **Required (Yes/No)**: Yes **Description**: Accrual Status Flag **Name**: Asset Classification Code **Required (Yes/No)**: Yes **Description**: Asset Classification Code

4.60 Get target accounts C1-GETTRGT

Table 4–157 Get target accounts C1-GETTRGT

Descriptio	Get target accounts

n	
Detailed Description	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: - 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). Parameters: - 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) identifiers. - 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	**Name**: Validation Failure Transition Status **Required (Yes/No)**: Yes **Description**: Validation Failure Transition Status **Name**: Cancel Reason Char Type **Required (Yes/No)**: No

	Description: Cancel Reason Char Type Code **Name**: Cancel Reason Char Value **Required (Yes/No)**: No **Description**: Cancel Reason Char Value **Name**: Casa Account Type Identifier List **Required (Yes/No)**: Yes **Description**: Casa Account Type Identifier List **Name**: Casa Account Exclude Status List **Required (Yes/No)**: Yes **Description**: Casa Account Exclude Status List **Name**:Td Account Type Identifier List **Required (Yes/No)**: Yes **Description**: To Do Account Type Identifier List **Name**: Td Account Exclude Status List **Required (Yes/No)**: Yes **Description**: Td Account Exclude Status List **Name**: Exclude Blocked Td Account **Required (Yes/No)**: Yes **Description**: Exclude Blocked Td Account **Name**: Exclude Blocked Deposit **Required (Yes/No)**: Yes **Description**: Exclude Blocked Deposit

4.61 Update status of relief to Expired in Hardship C1-UPDHDSTAT

Table 4–158 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
Paramet ers	**Name**: Hardship Expire Status **Required (Yes/No)**: Yes **Description**: Hardship Expire Status Code

4.62 To Do Completion for case C1-TO-DO-END

Table 4–159 To Do Completion for caseC1-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	**Name**: Do Not Complete To Do Type Characteristic Type **Required (Yes/No)**: No **Description**: Do Not Complete To Do Type Characteristic Type Code **Name**: Do Not Complete To Do Type Characteristic Value **Required (Yes/No)**: No **Description**: Do Not Complete To Do Type Characteristic Value

4.63 Update Marketing Consent flag C1-MKT-FLG

Table 4–160 Update Marketing Consent flag C1-MKT-FLG

Descript ion	Update Marketing Consent flag
Detailed Descript ion	This is a generic algorithm that will make a service call to host to update the Marketing Consent flag.
Algorith m Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateMarketingConsentFlag
Paramet ers	**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

4.64 Check Default Notice for Voluntary possession C1-CHKDFLT

Table 4–161 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 Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CheckDefaultNoticeForVoluntaryPossession
Parameters	**Name**: Check Expiry Status **Required (Yes/No)**: Yes **Description**: Check Expiry Status

4.65 Check Submission Date C1-CHKSUBDT2

Table 4–162 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

4.66 Update Financial Hardship flag C1-FNHRD-FLG

Table 4–163 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

4.67 Result Type Case Transition Algorithm C1-RTCT

Table 4–164 Update Financial Hardship flag C1-FNHRD-FLG

Description	Result Type Case Transition Algorithm
Detailed Description	If specified on the Result Type, this algorithm will be invoked when the corresponding result is recorded for a Case (Action/Result UI) This can be used to transition the case from the current status to the next possible status as follows: ■ 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	**Name**: Output Status

	Required (Yes/No): Yes **Description**: Output Status **Name**: Input Status **Required (Yes/No)**: No **Description**: Input Status

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

Table 4–165 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.CheckActiveArsCase
Parameters	**Name**: Case Category **Required (Yes/No)**: Yes **Description**: Case Category

4.69 Check Deceased status for the customer C1-CHKDCD

Table 4–166 Check Deceased status for the customer .C1-CHKDCD

Description	Check Deceased status for the customer
Detailed Description	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.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.CheckDeceasedStatusForCustomer
Paramet	**Name**: Case Category

ers	**Required (Yes/No)**: Yes **Description**: Case Category

4.70 Associated accounts with deceased customer case C1-DCDACCTS

Table 4–167 Associated accounts with deceased customer case.C1-DCDACCTS

Description	Associated accounts with deceased customer case
Detailed Description	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.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.AssociatedAccWithDeceasedCustomer
Parameters	NA

4.71 Execute Fund Transfer C1-FUNDTRFR

Table 4–168 Execute Fund Transfer C1-FUNDTRFR

Description	Execute Fund Transfer
Detailed Description	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" ■ 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 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" Once all target accounts have been processed, check if at least one payment transfer has status as "Success". ■ 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. Parameters:

	Execution Success Status - Case trasition status if fund tranfer is successful. Execution Failure Status - Case trasition status if fund tranfer fails. Cancel Reason Char Type - Characteristic type to set as case characteristic if fund transfer fails. Cancel Reason Char Value - Characteristic value for the defined characteristic type. Successful Fund Transfer Transaction Status - Trasaction status code to identify successful fund transfer.This values is returned from host service.
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ExecuteFundTransfer
Parameters	**Name**: Execution Success Status **Required (Yes/No)**: Yes **Description**: Execution Success Status Code **Name**: Execution Failure Status **Required (Yes/No)**: Yes **Description**: Execution Failure Status **Name**: Cancel Reason Char Type **Required (Yes/No)**: No **Description**: Cancel Reason Char Type **Name**: Cancel Reason Char Value **Required (Yes/No)**: No **Description**: Cancel Reason Char Value **Name**: Successful Fund Transfer Transaction Status **Required (Yes/No)**: Yes **Description**: Successful Fund Transfer Transaction Status

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

Table 4–169 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.SavePrevioustStatus
Parameters	NA

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

Table 4–170 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

4.74 Update Collection Warning Indicator C1-UPD-WRIND

Table 4–171 Update Collection Warning Indicator. C1-UPD-WRIND

Description	Update Collection Warning Indicator
Detailed Description	This is a generic algorithm that will make a service call to the Host to update Party level warning indicators for the Main Customer. It has following parameters: 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 **Required (Yes/No**): Yes **Description**: Collection Column To Be Updated **Name**: Set In Collections On Related Accounts **Required (Yes/No)**: Yes **Description**: Set In Collections On Related Accounts **Name**: Exception Transition Condition **Required (Yes/No)**: No **Description**: Exception Transition Condition

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

Table 4–172 Hardship Entity Association to nominated accounts and financial owners of account C1-HARDASSOD

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

4.76 Assign Applicable Relief Type C1-RELIF-TYP

Table 4–173 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 Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignApplicableReliefTypes
Parameters	**Name**: Rule Type Code **Required (Yes/No)**: Yes **Description**: Rule Type Code

4.77 Create Customer Contact for Resultype Algo C1-CREATCC

Table 4–174 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

4.78 Calculate an expiry date when entering case status C1-CSEXPDT

Table 4–175 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 **Description**: Number of Days **Name**: Expiry Date Characteristic Type **Required (Yes/No)**: Yes **Description**: Expiry Date Characteristic Type

4.79 Create Customer Contact C1-CUST-CONT

Table 4–176 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	**Name**: Customer Class **Required (Yes/No)**: Yes **Description**: Customer Class **Name**: Customer Contact Type

	Required (Yes/No): Yes **Description**: Customer Contact Type **Name**: Char Type Cust Cont Log Entry **Required (Yes/No)**: Yes **Description**: Char Type Cust Cont Log Entry **Name**: X Path Completion Flag **Required (Yes/No)**: No **Description**: X Path Completion Flag **Name**: Transition Condition **Required (Yes/No)**: Yes **Description**: Transition Condition **Name**: Contact Method **Required (Yes/No)**: Yes **Description**: Contact Method

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

Table 4–177 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	**Name**: Days Before Expiry **Required (Yes/No)**: Yes **Description**: Days Before Expiry **Name**: Xpath to Expiry Date **Required (Yes/No)**: Yes **Description**: Xpath to Expiry Date **Name**: Next Status **Required (Yes/No)**: No **Description**: Next Status

	Name: Next Transition Condition **Required (Yes/No)**: No **Description**: Next Transition Condition

4.81 Validate Hardship Expiry Date. C1-VAL-FHEXP

Table 4–178 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	**Name**: Xpath To Hardship Expiry Date **Required (Yes/No)**: No **Description**: Xpath To Hardship Expiry Date **Name**: Expiry Date Char Type **Required (Yes/No)**: No **Description**: Expiry Date Char Type

4.82 Update Account in collections flag C1-ACTINCOL

Table 4–179 Update Account in collections flag. C1-ACTINCOL

Description	Update Account in collections flag
Detailed Description	Get all accounts for the customer from the host. Relationship type to be considered will be primary or financial ownership based on parameter set for the process. For the accounts retrieved, check if the account is setup in collections i.e. an active contract is present for the account ▪ 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	**Name**: relationshipType **Required (Yes/No)**: Yes **Description**: Relationship_Type **Name**: sourceHostId **Required (Yes/No)**: Yes **Description**: Source Host Id

4.83 Associate related entities with the case C1-ARSENTITY

Table 4–180 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	**Name**: Customer Association **Required (Yes/No)**: Yes **Description**: Customer Association **Name**: Account Association **Required (Yes/No)**: Yes **Description**: Account Association

4.84 Revalidate target account C1-REVALTRGT

Table 4–181 Revalidate target account. C1-REVALTRGT

Description	Revalidate target account
Detailed Description	This algoritm validates target account (savings and term deposit) balance and computes maximum amount to be debited. Processing logic should be as below: ■ 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
  - 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".
  - If for any of the account validation status = "Failure",
    - Set set-off status and exclude reason as "Not eligible".

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

- 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
  - 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
- Sum of Debit amounts of target account <= Overdue amount for delinquent account. In this case there is no exception and set-off process should proceed.

Cancel Set-off

- Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters)

Example of proportionate adjustment:

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

Now during revalidation it is found that overdue has dropped to $ 60. So now below computations should be done

- X = (Overdue amount) / (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

So in this case

- 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 Parameters: - 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 Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RevalidateTargetAccount
Parameters	**Name**: Cancel Reason Char Value **Required (Yes/No)**: No **Description**: Cancel Reason Char Value **Name**: Cancel Reason Char Type **Required (Yes/No)**: No **Description**: Cancel Reason Char Type **Name**: Validation Failure Transition Status **Required (Yes/No)**: Yes **Description**: Validation Failure Transition Status **Name**: Excess Debit Option **Required (Yes/No)**: Yes **Description**: Excess Debit Option **Name**: Minimum Residual Amount

	Required (Yes/No): Yes **Description**: Minimum Residual Amount **Name**: Rule Id **Required (Yes/No)**: Yes **Description**: Rule Id **Name**: Casa Account Type Identifier List **Required (Yes/No)**: Yes **Description**: Casa Account Type Identifier List **Name**: Td Account Type Identifier List **Required (Yes/No)**: Yes **Description**: Td Account Type Identifier List

4.85 Initiate LMI C1-INITLMIS

Table 4–182 Initiate LMI. C1-INITLMIS

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

	Name: Balance Threshold **Required (Yes/No)**: Yes **Description**: Balance Threshold

4.86 NGP Collection case creation algorithm C1-COLLCASE

Table 4–183 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	**Name**: New Case Creation Rule Id **Required (Yes/No)**: Yes **Description**: New Case Creation Rule Id **Name**: Exist Case Creation Rule Id **Required (Yes/No)**: Yes **Description**: Exist Case Creation Rule Id

4.87 Stop Contract Algorithm C1-CONTSTOP

Table 4–184 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

4.88 Check for existing Hardship C1-CHKHRDSHP

Table 4–185 Check for existing Hardship. C1-CHKHRDSHP

Description	Check for existing Hardship
Detailed Description	Before creating case in Pending state. This Algorithm checks, if there is any active case of Hardship case type (By Retrieveing case type code from feature configuration). If yes, It Display message ' Party is already in Hardship' If no, It will proceed with case creation. This checks for an existing Hardship application for the party.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckExistingHardship
Parameters	**Name**: Hardship Case Type Feature Config **Required (Yes/No)**: Yes **Description**: Hardship Case Type Feature Config

4.89 Algorithm for contact processing C1-CNTCT

Table 4–186 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

4.90 Check application expiry date C1-CHKEXP

Table 4–187 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

4.91 New Customer Contact Creation Algorithm C1-CCCREATE

Table 4–188 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 **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

4.92 Removes a case characteristic on case status exit C1-REMCSCH

Table 4–189 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	**Name**: Characteristic Type **Required (Yes/No)**: Yes **Description**: Characteristic Type

4.93 Transition case on a date on a case characteristic C1-TRANSDT

Table 4–190 Transition case on a date on a case characteristic. C1-TRANSDT

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	**Name**: Characteristic Type **Required (Yes/No)**: Yes **Description**: Characteristic Type **Name**: Next Status **Required (Yes/No)**: No **Description**: Next Status **Name**: Next Transition Condition **Required (Yes/No)**: No **Description**: Next Transition Condition

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

Table 4–191 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

4.95 Mark accounts for strategy review C1-REVIW-ACT

Table 4–192 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

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

Table 4–193 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	**Name**: Expiration Days **Required (Yes/No)**: No **Description**: Expiration Days **Name**: Xpath To Expiration Days **Required (Yes/No)**: No **Description**: Xpath To Expiration Days **Name**: Time Out To Do Type **Required (Yes/No)**: Yes

	Description: Time Out To Do Type **Name**: Log Entry Char Type Fk To To Do **Required (Yes/No)**: Yes **Description**: Log Entry Char Type Fk To To Do **Name**: Work Calendar **Required (Yes/No)**: Yes **Description**: Work Calendar

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

Table 4–194 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

4.98 Check for Operational Relief Types C1-OP-RT

Table 4–195 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	**Name**: Operational Relief Type 1 **Required (Yes/No)**: No

	Description: Operational Relief Type 1 **Name**: Operational Relief Type 2 **Required (Yes/No)**: No **Description**: Operational Relief Type 2 **Name**: Operational Relief Type 3 **Required (Yes/No)**: No **Description**: Operational Relief Type 3 **Name**: Operational Relief Type 4 **Required (Yes/No)**: No **Description**: Operational Relief Type 4 **Name**: Operational Relief Type 5 **Required (Yes/No)**: No **Description**: Operational Relief Type 5 **Name**: Operational Relief Type 6 **Required (Yes/No)**: No **Description**: Operational Relief Type 6 **Name**: Operational Relief Type 7 **Required (Yes/No)**: No **Description**: Operational Relief Type 7 **Name**: Operational Relief Type 8 **Required (Yes/No)**: No **Description**: Operational Relief Type 8 **Name**: Operational Relief Type 9 **Required (Yes/No)**: No **Description**: Operational Relief Type 9 **Name**: Operational Relief Type 10 **Required (Yes/No)**: No **Description**: Operational Relief Type 10

4.99 Auto-Approval Check C1-FH-AUTOAP

Table 4–196 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	**Name**: Rule Type Code **Required (Yes/No)**: Yes **Description**: Rule Type Code

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

Table 4–197 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	**Name**: Xpath To Completion Flag **Required (Yes/No)**: Yes **Description**: Xpath To Completion Flag **Name**: Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition

4.101 Update Party Warning Indicator C1-UPD-PRTWI

Table 4–198 Update Party Warning Indicator. C1-UPD-PRTWI

Description	Update Party Warning Indicator
Detailed Descript	This is a generic algorithm that will make a service call to host to update Party level warning indicators for Main Customer.

ion	If a Rule Type Code is populated, it will first invoke the rule to determine if the Warning Indicator should be updated.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdatePartyWarningIndicator
Parameters	**Name**: Warning Indicator Type **Required (Yes/No)**: Yes **Description**: 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**: Exception Transition Condition **Required (Yes/No)**: No **Description**: Exception Transition Condition

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

Table 4–199 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	**Name**: Next Status

	Required (Yes/No): No **Description**: Next Status Code **Name**: Next Transition Condition **Required (Yes/No)**: No **Description**: Next Transition Condition

4.103 Collection - Entity Activity Population. C1-ENTACTPOP

Table 4–200 Collection - Entity Activity Population. C1-ENTACTPOP

Description	Collection - Entity Activity Population
Detailed Description	This sample algorithm is called from various entities classes for population of Account Activity. The algorithm takes following input parameters: 1. EntityType : Person/Account for which activity is getting created (e.g. Case can be created on Person as well as Account) 2. EntityId : Person/Account Id 3. ModeOfOperation: Add/Update/Delete/Cancel 4. HostEntitytId: Activity Entity Id (e.g PTP/CC/Follow-up/Case Id) 5. HostEntitytName: PTP/CC/FOLLOWUP/CASE
Algorithm Entity	Installation – Entity Activity Populate
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseCreation.PopulateAccountActivityAlgo
Parameters	NA

4.104 Cancel Hardship Application C1-CXLFH

Table 4–201 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

4.105 Perform Payment Transfer for ROSO C1-ROSOPMTXR

Table 4–202 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	**Name** : Exception Transition Condition **Required (Yes/No)**: Yes **Description**: Exception Transition Condition **Name**: Host Code **Required (Yes/No)**: Yes **Description**: Host Code **Name** : Failed Leg Notification To Do Type **Required (Yes/No)**: Yes **Description**: Failed Leg Notification To Do Type

4.106 Validate ROSO Target Account inputs C1-RS-VALIN

Table 4–203 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. ValidateRosoInputs
Parameters	**Name** : Exception Transition Condition **Required (Yes/No)**: No **Description**: Exception Transition Condition **Name**: Exception Status **Required (Yes/No)**: No **Description**: Exception Status

4.107 Create RMB Entities from Host Data C1-VCREATE

Table 4–204 Create RMB Entities from Host Data. C1-VCREATE

Description	Create RMB Entities from Host Data
Detailed Description	Create RMB Entities such as Person, Account ,Account Person, PartyCollect etc from Host Data. Input parameters: 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 (accrl_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

4.108 Populate Activity Table For Notes Creation C1-NTACTVITY

Table 4–205 Populate Activity Table For Notes Creation. C1-NTACTVITY

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

4.109 Suspend Activity for Account Pre Processing C1-SPATACPRE

Table 4–206 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

4.110 Sample TAM Algorithm Type C1-TAMALG

Table 4–207 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	**Name**: acctReviewDays **Required (Yes/No)**: No **Description**: Account Review Days **Name**: tamReviewDays

	Required (Yes/No): No **Description**: TAM Review Days

4.111 Cancel Approval Request C1-CANAPPR

Table 4–208 Cancel Approval Request. C1-CANAPPR

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

4.112 Update Customer in collections flag C1-CUSINCOL

Table 4–209 Update Customer in collections flag. C1-CUSINCOL

Description	Update Customer in collections flag
Detailed Description	This algorithm will set or reset in-collections flag for the customer in core banking While setting the Flag ■ Skip the process if flag is already set ■ If not, set the in-collection flag for the customer to "Y" in the host. While resetting the Flag, ■ 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.UpdateCustomerInCollFlg
Parameters	**Name**: Update Type **Required (Yes/No)**: Yes **Description**: Update Type **Name**: Case Category1 **Required (Yes/No)**: No **Description**: Case Category1 Code **Name**: Case Category2 **Required (Yes/No)**: No **Description**: Case Category2 Code **Name**: Case Category3 **Required (Yes/No)**: No **Description**: Case Category3 Code **Name**: Case Category4 **Required (Yes/No)**: No **Description**: Case Category4 Code **Name**: Case Category5 **Required (Yes/No)**: No **Description**: Case Category5 Code

4.113 Set Display Date C1-SETDSPDT

Table 4–210 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	**Name**: Offset Dayes **Required (Yes/No)**: Yes **Description**: Offset Dayes

4.114 Transition to Default next status after N Days C1-TRNDFLT

Table 4–211 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 Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.DefaultNextStatusAutoTransition
Parameters	**Name**: Wait Days **Required (Yes/No)**: Yes **Description**: Wait Days

4.115 Check current cases on account for exclusion C1-EXCLCASE

Table 4–212 Check current cases on account for exclusion. C1-EXCLCASE

Description	Check current cases on account for exclusion
Detailed Description	System should maintain a lookup with list of case categories for set-off exclusion. Processing logic should be as below- ■ 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

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

4.116 Update Collateral Status in the host C1-UPCOLLST

Table 4–213 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	**Name**: To Do Type **Required (Yes/No)**: Yes **Description**: To Do Type **Name**: Collateral Status **Required (Yes/No)**: Yes **Description**: Collateral Status

4.117 Initiate collateral valuation C1-COLLVAL

Table 4–214 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	**Name**: To Do Type **Required (Yes/No)**: Yes **Description**: To Do Type **Name**: Days Since Closure Of Last To Do **Required (Yes/No)**: Yes **Description**: Days Since Closure Of Last To Do

	Name: Assessment Expiry Days **Required (Yes/No)**: Yes **Description**: Assessment Expiry Days

4.118 Mandatory characteristics check for Asset Repo C1-CHARVAL

Table 4–215 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	**Name**: Reference Characteristic Value **Required (Yes/No)**: Yes **Description**: Reference Characteristic Value **Name**: Reference Characteristic **Required (Yes/No)**: Yes **Description**: Reference Characteristic **Name**: Case Characteristic5 **Required (Yes/No)**: Yes **Description**: Case Characteristic5 **Name**: Case Characteristic4 **Required (Yes/No)**: Yes **Description**: Case Characteristic4 **Name**: Case Characteristic3 **Required (Yes/No)**: Yes **Description**: Case Characteristic3 **Name**: Case Characteristic2 **Required (Yes/No)**: Yes **Description**: Case Characteristic2

	Name: Case Characteristic1 **Required (Yes/No)**: Yes **Description**: Case Characteristic1

4.119 Update Collateral Status in the Host C1-UPCOLLSTS

Table 4–216 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	**Name**: To Do Type **Required (Yes/No)**: Yes **Description**: To Do Type **Name**: Collateral Status **Required (Yes/No)**: Yes **Description**: Collateral Status

4.120 Set exclusion date for delinquent account. C1-EXCLROSO

Table 4–217 Set exclusion date for delinquent account. C1-EXCLROSO

Description	Set exclusion date for delinquent account
Detailed Description	This process will set set-off exclusion date for the delinquent account. Processing will be driven by parameters set for the process. Set-off Exclusion date should be updated only if current exclusion date is <= business date. Else, skip all below processing If Cancel Reason char type parameters is not bank ■ 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. If Cancel Reason char type parameters is blank but Reason code is provided - 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. Parameters: - 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	**Name**: Default Offset **Required (Yes/No)**: Yes **Description**: Default Offset **Name**: Reason Code **Required (Yes/No)**: No **Description**: Reason Code **Name**: Cancel Reason **Required (Yes/No)**: No **Description**: Cancel Reason

4.121 Cancel Set-off C1-CANROSO

Table 4–218 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

4.122 Complete Set-off C1-COMPROSO

Table 4–219 Complete Set-off. C1-COMPROSO

Description	Complete Set-off
Detailed Description	This algorithm transitions the case to complete. Processing Logic will be as below ■ 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

4.123 Reverse Set-off C1-REVROSO

Table 4–220 Reverse Set-off. C1-REVROSO

Description	Reverse Set-off
Detailed Description	This algorithm transitions the case to Reversed status. Processing Logic will be as below ■ Validate below for each target account • 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

4.124 Algorithm type for update case id for Insurance C1-UPCASFINS

Table 4–221 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

4.125 Case Creation on enter processing C1-CCOENTER

Table 4–222 Case Creation on enter processing. C1-CCOENTER

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

4.126 Collection - Case Creation On Exit of Status C1-CCOE

Table 4–223 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 algorihtm 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

4.127 Action category Validation algorithm C1-ACTCAT

Table 4–224 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

4.128 Action Type Algorithm Type C1-ACTTYP

Table 4–225 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

4.129 Case Type Status Mapping Algorithm Type C1-CASETYMP

Table 4–226 Case Type Status Mapping Algorithm Type C1-CASETYMP

Descriptio	Case Type Status Mapping Algorithm Type

n	
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

4.130 Collection - Close Processing Algorithm C1-CCALG

Table 4–227 Collection - Close Processing Algorithm. C1-CCALG

Description	Collection - Close Processing Algorithm
Detailed Description	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). ■ It will update the financial balance of the Contract to zero through an adjustment. ■ 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 The following parameters are available and are required: ■ Adjustment Type used for the adjustment created by this algorithm. ■ Cancelation 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	**Name**: Adjustment Type **Required (Yes/No)**: Yes **Description**: Adjustment Type **Name**: Cancelation Reason Code for PTP **Required (Yes/No)**: Yes **Description**: Cancelation Reason Code for PTP **Name**: Is Closing Required Flag (Y/N) **Required (Yes/No)**: Yes **Description**: Is Closing Required Flag **Name**: Final Default Case Status **Required (Yes/No)**: Yes **Description**: Final Default Case Status

4.131 Algorithm type for case list update C1-CASELIST

Table 4–228 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

4.132 Copy Characteristics Algorithm C1-COPYCHAR

Table 4–229 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	Case Type – Enter Status

Entity	
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CopyCharacteristicsOnCaseCreate
Parameters	**Name**: Case Category **Required (Yes/No)**: Yes **Description**: Case Category **Name**: Characteristics List **Required (Yes/No)**: No **Description**: Characteristics List

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

Table 4–230 Call Advice - Red/Green logic calculation. C1-CALADVICE

Description	Call Advice - Red/Green logic calculation
Detailed Description	Call Advice - Red/Green logic calculation ■ Call Advice will be 'Green' if • '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 Else it is 'Red'.
Algorithm Entity	Installation – Contact Information
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.collectionLandingPage.ContactInformationCallAdviceAlgo
Parameters	NA

4.134 Task Case Mapping Validation Algorithm C1-TCVAL

Table 4–231 Task Case Mapping Validation Algorithm. C1-TCVAL

Description	Task Case Mapping Validation Algorithm

Detailed Description	Task Case Mapping Validation Algorithm 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. This algorithm will validate the Repossession Date field only if value is already present. Validation Date can be SYSTEM DATE or POSTING DATE.
Algorithm Entity	Task Type Case Type Validation Algorithm Spot
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepoDateValidation
Parameters	**Name**: Validation Date **Required (Yes/No)**: Yes **Description**: Validation Date **Name**: Case Type Code **Required (Yes/No)**: Yes **Description**: Case Type Code **Name**: Task Type Code **Required (Yes/No)**: Yes **Description**: Task Type Code

4.135 Monitoring Algorithm For Dispute Resolved to exit from Dispute Status C1-DISMON

Table 4–232 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	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: ■ 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	**Name**: Validation Date **Required (Yes/No)**: Yes **Description**: Validation Date **Name**: No Of Grace Days **Required (Yes/No)**: Yes **Description**: No Of Grace Days **Name**: Contact Status **Required (Yes/No)**: No **Description**: Contact Status **Name**: Contact RM Status **Required (Yes/No)**: No **Description**: Contact RM Status **Name**: Contact Alternate Status **Required (Yes/No)**: No **Description**: Contact Alternate Status **Name**: Characteristic Code

	Required (Yes/No): No **Description**: Characteristic Code **Name**: Characteristic Value **Required (Yes/No)**: No **Description**: Characteristic Value **Name**: Risk Indicator Code **Required (Yes/No)**: No **Description**: Risk Indicator Code

4.136 SLA Parameters validation algorithm C1-SLAPARAM

Table 4–233 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 CI-SLABO.
Algorithm Entity	Business Object – Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.vendor.SLAParametersPostProcessAlgo
Parameters	NA

4.137 Case Group add validation algorithm C1-CGVAL

Table 4–234 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

4.138 Collection - Get Strategy Algorithm C1-COLGS

Table 4–235 Collection - Get Strategy Algorithm. C1-COLGS

Description	Collection - Get Strategy Algorithm
Detailed Description	This algorithm calls the Rules Engine to determine a collection strategy. It is invoked by the Collection Class Overdue Rule - Overdue Monitor Rule The following parameters are passed to the Rules Engine :

	- 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	**Name**: Rule Type **Required (Yes/No)**: Yes **Description**: Rule Type

4.139 Create/Move Collection Strategy Cases for Account C1-COLOMR

Table 4–236 Create/Move Collection Strategy Cases for Account. C1-COLOMR

Description	Create/Move Collection Strategy Cases for Account
Detailed Description	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. It will first check for the Collection events (contracts) that are under the account. 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. 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. Notes on the algorithm parameters - 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	**Name**: Final Default Case Status **Required (Yes/No)**: Yes **Description**: Final Default Case Status **Name**: Is Closing Required Flag (Y/N) **Required (Yes/No)**: Yes **Description**: Is Closing Required Flag (Y/N) **Name**: Collection Closing Algorithm Code **Required (Yes/No)**: Yes **Description**: Collection Closing Algorithm Code **Name**: Get Strategy Algorithm Code **Required (Yes/No)**: Yes **Description**: Get Strategy Algorithm Code

4.140 Collection - Case SA Update for Manual Creation C1-CSAUPD

Table 4–237 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

4.141 Promise to Pay - Additional Grace Days Sample Algo C1-PPADDLGRD

Table 4–238 Promise to Pay - Additional Grace Days Sample Algo. C1-PPADDLGRD

Description	Promise to Pay - Additional Grace Days Sample Algo
Detailed Description	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.

	The algorithm takes the input parameter value and passes it back to the Promise to Pay Monitor as additional grace days.
Algorithm Entity	Installation – Additional Grace Days
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.payPlan.AdditionalGraceDaysCalculationAlgorithm
Parameters	**Name**: Additional Grace Days **Required (Yes/No)**: No **Description**: Additional Grace Days

4.142 Promise to Pay Threshold Percentage C1-PPTHRESH

Table 4–239 Promise to Pay Threshold Percentage. C1-PPTHRESH

Description	Promise to Pay Threshold Percentage
Detailed Description	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. It receives the following inputs from the pay plan monitor ■ Promise to Pay ID ■ Total Amount Paid towards the promise to pay ■ Date (Business Date - Grace Days) ■ Array of Promise to Pay Scheduled Payments balance The algorithm will check if the Total Amount Paid is within the threshold percentage (input parameter) of the Total Scheduled Payments expected. 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 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.
Algorithm Entity	Installation – Payment Threshold Percentage
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.payPlan.PaymentThresholdPercentageCalculationAlgorithm
Parameters	**Name**: Threshold Percentage **Required (Yes/No)**: No **Description**: Threshold Percentage

4.143 Result type Post Processing Case Transition Algo C1-RTPCC

Table 4–240 Result type Post Processing Case Transition Algo. C1-RTPCC

Description	Result type Post Processing Case Transition Algo.
Detailed Description	If specified on the Result Type, this algorithm will be invoked when the corresponding result is recorded for a Case (Action/Result UI). This can be used to transiton the case from the current status to the next possible status as follow, ■ 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 algorihtm 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	**Name**: Output Status **Required (Yes/No)**: Yes **Description**: Output Status **Name**: Input Status **Required (Yes/No)**: No **Description**: Input Status

4.144 Inbound Customer algorithm C1-IN-CUST

Table 4–241 Inbound Customer algorithm. C1-IN-CUST

Description	Inbound Customer algorithm
Detailed Description	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 ■ 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 and error while processing it will transition the lifecycle to the "Rejected" status. It has two parameters , both optional. ■ 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	**Name**: Account Id Type **Required (Yes/No)**: No **Description**: Account Id Type **Name**: Person Id Type **Required (Yes/No)**: No **Description**: Person Id Type

4.145 Result Type Pre-processing Algorithm Type C1-RSTPRE

Table 4–242 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

4.146 Result Type Post-processing Algorithm Type C1-RSTPOST

Table 4–243 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	**Name**: contactType **Required (Yes/No)**: Yes **Description**: Contact Type **Name**: contactClass **Required (Yes/No)**: Yes **Description**: Contact Class **Name**: contactMethod **Required (Yes/No)**: Yes **Description**: Contact Method

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

Table 4–244 Characteristic Type :Validate Date Field (Custom). C1-ADHDATE

Description	Characteristic Type :Validate Date Field (Custom)
Detailed Description	Custom Date validation 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. 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. 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.

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

4.148 Algorithm Type for Dialer Results Upload C1-DLRRSUPLD

Table 4–245 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

4.149 Algorithm for Hardship case creation activity C1-CRTHDSP

Table 4–246 Algorithm for Hardship case creation activity. C1-CRTHDSP

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

4.150 This Algorithm is used to abort Approval work item C1-ABORTAPP

Table 4–247 This Algorithm is used to abort Approval work item. C1-ABORTAPP

Description	This Algorithm is used to abort Approval work item
Detailed Description	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. ■ caseStatusExclusionList: Comma seperated 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. • 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 parmeters 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	**Name**: Composite Name **Required (Yes/No)**: Yes **Description**: Composite Name **Name**: Instance Title **Required (Yes/No)**: Yes **Description**: Instance Title **Name**: Case Status Exclusion List **Required (Yes/No)**: No **Description**: Case Status Exclusion List

4.151 Cancel Process Approval Request:Financial Hardship C1-CANFHAPPR

Table 4–248 Cancel Process Approval Request:Financial Hardship. C1-CANFHAPPR

Description	Cancel Process Approval Request:Financial Hardship

Detailed Description	This algorithm will cancel all pending approval requests for the case. Example for parametervalues for hardship Process: ■ Composite Name: com.ofss.fc.workflow.process.FinancialHardshipProcessForApproval ■ Instance Title: FINANCIAL_HARDSHIP_CASE_ Value of the above parmeters depends upon the SOA approval work flow.
Algorithm Entity	Case Type – Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelFinancialHardshipApprovalReqAlgo
Parameters	**Name**: Composite Name **Required (Yes/No)**: Yes **Description**: Composite Name **Name**: Instance Title **Required (Yes/No)**: Yes **Description**: Instance Title

4.152 Hardship Characteristic Association C1-FHCHARASC

Table 4–249 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

4.153 Pre-Populated system facts to be used for Rule C1-PPSF

Table 4–250 Pre-Populated system facts to be used for Rule. C1-PPSF

Description	Pre-Populated system facts to be used for Rule
Detailed Description	This algoritm is used to populate input system fact for Rule.It used as an input to RuleFactPopulation algorithm. System Facts populated through this algoritm are SystemDate and PostingDate. This is sample implementation to provide populated facts to RuleFactPopulation algoritm.User can create his own algoritm type based on his requirement (Algoritm 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

4.154 Assign Batch level TODOs(task) to a queue C1-ASGNTASK

Table 4–251 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

4.155 Validate Date Field :Custom C1-ADHV-DTD

Table 4–252 Validate Date Field :Custom. C1-ADHV-DTD

Description	Validate Date Field :Custom
Detailed Description	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. 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.

	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.
Algorithm Entity	Characteristic Type – Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.ProductAdhocDateValidationAlgComp
Parameters	**Name**: From Date **Required (Yes/No)**: No **Description**: From Date **Name**: To Date **Required (Yes/No)**: No **Description**: To Date **Name**: Date Format1 (Stored Format) **Required (Yes/No)**: Yes **Description**: Date Format1 (Stored Format) **Name**: Date Format2 **Required (Yes/No)**: No **Description**: Date Format2 **Name**: Date Format3 **Required (Yes/No)**: No **Description**: Date Format3 **Name**: Date Format4 **Required (Yes/No)**: No **Description**: Date Format4 **Name**: Date Format5 **Required (Yes/No)**: No **Description**: Date Format5 **Name**: Date Format6 **Required (Yes/No)**: No **Description**: Date Format6

4.156 Characteristic Date field Validation C1-CHARDTVAL

Table 4–253 Characteristic Date field Validation. C1-CHARDTVAL

Description	Characteristic Date field Validation
Detailed Description	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. 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. 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. 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.
Algorithm Entity	Characteristic Type – Adhoc Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.CharAdhocDateValidation
Parameters	**Name**: From Date **Required (Yes/No)**: No **Description**: From Date **Name**: To Date **Required (Yes/No)**: No **Description**: To Date **Name**: Date Format1 (Stored Format) **Required (Yes/No)**: Yes **Description**: Date Format1 (Stored Format) **Name**: Date Format2 **Required (Yes/No)**: No **Description**: Date Format2 **Name**: Date Format3 **Required (Yes/No)**: No

	Description: Date Format3 **Name**: Date Format4 **Required (Yes/No)**: No **Description**: Date Format4 **Name**: Date Format5 **Required (Yes/No)**: No **Description**: Date Format5 **Name**: Date Format6 **Required (Yes/No)**: No **Description**: Date Format6 **Name**: Business Date Validation Required **Required (Yes/No)**: No **Description**: Business Date Validation Required

4.157 Retry Case in Error C1-RCASEE

Table 4–254 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

4.158 Allocate Queue for Customer Level Case C1-ALLOCQUE

Table 4–255 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

4.159 Person Address – Collection C1-PERADDRC

Table 4–256 Person Address – Collection. C1-PERADDRC

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

4.160 Person Contact Point Update - Post Processing C1-PERCONTPP

Table 4–257 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 Prefrence – Post Process
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot
Parameters	NA

4.161 Update Self Serve Flag Algorithm C1-SELFSERVE

Table 4–258 Update Self Serve Flag Algorithm. C1-SELFSERVE

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

4.162 Create Task for Self Serve Request for Assistance transaction C1-FLWRTSK

Table 4–259 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 Localized Algorithms

5.1 Localized Algorithms

Table 5–1 Case Transition for Active Service Member C1-ACTMEMCHK

Description	Case Transition for Active Service Member
Detailed Description	This algorithm will transit the case to Suspend Status if the customer is in Active Service or dependent of a person in Active Service. 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.
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.scra.algorithm.ActiveServiceAlgorithm
Parameters	Name: Suspend Status Required (Yes/No): No Description: Suspend Status Name: All Financial Owner Validation Required (Yes/No): Yes Description: All Financial Owner Validation Name: Validation Date Required (Yes/No): Yes Description: Validation Date Name: Dependent Validation Required (Yes/No): Yes Description: Dependent Validation Name: Suspend Reason Characterics Required (Yes/No): No Description: Suspend Reason Characterics
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 5–2 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	Name: Validation Date Required (Yes/No): Yes Description: Validation Date Name: Repossession Block Period Required (Yes/No): No Description: Repossession Block Period
Detailed Design	Verify if repossession needs to be blocked as per SCRA regulations

Table 5–3 Metro 2 Reporting - Account Status Code post Liquidation C1- ASCLIQU

Description	Metro 2 Reporting - Account Status Code post Liquidation
Detailed Description	If Repossession Reason = Voluntary Surrender If Account Status Condition = Consumer not responsible for Remaining Balance/ No Deficiency Balance Set Account Status Code = 95; If Account Status Condition = Consumer responsible for Remaining Balance' Set Account Status Code = 95; If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full Set Account Status Code = 61; Else If Account Status Condition = Consumer not responsible for Remaining Balance/ No Deficiency Balance

	Set Account Status Code = 96; If Account Status Condition = Consumer responsible for Remaining Balance' Set Account Status Code = 96; If Account Status Condition = Consumer responsible for Remaining Balance - Amount Paid in Full Set Account Status Code = 63; 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) Account status Code Char value should be C1-ASCOD. It should be product shipped. Char Values are: CNRBND,CRBAP,CRRB
Algorithm Entity	Result Type – Post Processing Algorithm
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms. Metro2AcctStatusCodePostLiquidationPostProcessing
Parameters	Name: Account status Code Char Required (Yes/No): Yes Description: Account status Code Char Name: Voluntary Surrender Code Required (Yes/No): Yes Description: Voluntary Surrender Code
Detailed Design	Metro 2 Reporting - Account Status Code post Liquidation

Table 5–4 Metro 2 Reporting - Account Status Code C1- ASCREPO

Description	Metro 2 Reporting - Account Status Code
Detailed Description	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

	associated accounts)
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatuscodeEnterProcessingAlgo
Parameters	Name: Special Comment Code for Normal Repossession Required (Yes/No): No Description: Special Comment Code for Normal Repossession Name: Special Comment Code for Voluntary Surrender Required (Yes/No): No Description: Special Comment Code for Voluntary Surrender Name:Voluntary Surrender Code Required (Yes/No): Yes Description: Voluntary Surrender Code Name: Account Status Code for Normal Repossession Required (Yes/No): Yes Description: Account Status Code for Normal Repossession Name: Account Status Code for Voluntary Surrender Required (Yes/No): Yes Description: Account Status Code for Voluntary Surrender
Detailed Design	Metro 2 Reporting - Account Status Code

Table 5–5 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 5–6 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 5–7 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

	Description: Chapter7 CII Code Name: Chapter11 CII Code Required (Yes/No): Yes Description: Chapter11 CII Code Name: Chapter12 CII Code Required (Yes/No): Yes Description: Chapter12 CII Code Name: Chapter13 CII Code Required (Yes/No): Yes Description: Chapter13 CII Code Name: Other CII Code Required (Yes/No): Yes Description: Other CII Code
Detailed Design	Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.

Table 5–8 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	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
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInfoIndiChap13PostDis
Parameters	Name:No Confirmed Plan CII Code Required (Yes/No): Yes Description: No Confirmed Plan CII Code Name: Chapter12 CII Code Required (Yes/No): Yes Description: Chapter12 CII Code

	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 5–9 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 5–10 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	Name: Party Id Char Required (Yes/No): Yes Description: Party Id Char Name: Cii Char Required (Yes/No): Yes Description: Cii Char
Detailed Design	Consumer Now Located (Removes previously reported T Indicator)

Table 5–11 Metro 2 Reporting - Set DPD and Outstanding amount to all associated accounts C1-SETDPD

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

6 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 6–1 Feeder Services

Service Name	Method Name	Description	Mandatory Fields
AccountFeederApplicationService	AccountFeederResponse update (SessionContext sessionContext,AccountFeederWrapperDTO accountFeederWrapperDTO) throws FatalException	This service adds or updates account related fields in the feeder table. It handles add, update and delete operations.	hostAcctNumber, srcHostId
AccountHardshipDtlsFeederApplicationService	AccountHardshipDtlsFeederResponse update(SessionContext sessionContext,AccountFeederHardshipDtlsWrapperDTO accountFeederHardshipDtlsWrapperDTO) throws FatalException;	This service adds or updates accounts hardship related fields in the feeder table. It handles add, update and delete operations.	hostAcctNumber, srcHostId, reliefEffDt, reliefExpDt, reliefType, hrshipAppId
AccountArrearFeederApplication Service	AccountArrearFeederResponse update(SessionContext sessionContext,AccountArrearFeeder	This	hostAcctNumber, srcHostId, referenceVal

Service Name	Method Name	Description	Mandatory Fields
	WrapperDTO accountArrearFeederWrapperDTO) throws FatalException;	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.	
AccountWarningIndFeederApplicationService	AccountWarningIndFeederResponse update(SessionContext sessionContext,AccountWarningIndFeederWrapperDTO accountWarningIndFeederWrapperDTO) throws FatalException;	This service adds or updates account warning indicator related fields in the feeder table. It handles add, update and delete operations.	hostAcctNumber, srcHostId

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 address related fields in the feeder table. It handles add, update and	srcHostId, hostCustomerNbr, fdrAddrSeqId, addrTypeCd

Service Name	Method Name	Description	Mandatory Fields
		delete operations.	
FeederPerEmpProfileApplicationService	FeederPerEmpProfileResponse update(SessionContext sessionContext,FeederPerEmpProfileWrapperDTO feederPerEmpProfileWrapperDTO) throws FatalException	This service adds or updates party employment details fields in the feeder table. It handles add, update and delete operations.	srcHostId, hostCustomerNbr, determinantValue, fdrEmpSeqId
FeederContactPrefApplicationService	FeederContactPrefResponse update (SessionContext p_SessionContext, FeederContactPrefWrapperDTO p_FeederContactPrefWrapperDTO) throws FatalException	This service adds or updates party contact preferences fields in the feeder table. It handles add, update and delete operations.	srcHostId, hostCustomerNbr, contactPrefType, contactPointType
FeedePerIdApplicationService	FeedePerIdResponse update (SessionContext p_SessionContext, FeedePerIdWrapperDTO p_FeedePerIdWrapperDTO) throws FatalException	This service adds or updates party ID type related fields, such as driving	srcHostId, hostCustomerNbr, 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	hostCustomerNbr, 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	

7 Dialer Webservice Integration

Dialer web service can be consumed by consultants to notify collector about the outbound call to customer by vendor.

7.1 Generic Data Type

This section provides details of the generic data type.

Table 7–1 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)	■ 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	■ 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	A valid value must be reported. For a single character mandatory field, blank (space) is not a valid value. A mandatory: ■ 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	A valid value must be reported where specified conditions are met. In this case the field becomes mandatory. See rules above. If the data is not available, then: ■ alpha field, must be space filled ■ alphanumeric field, must be space filled

Data Type	Format
	■ numeric, must be zero filled ■ date field, must be zero filled

7.2 Summary

This section provides the detail summary for dialer webservice.

Table 7–2 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

7.3 Interface

This section provides the details on the interface.

Table 7–3 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

7.4 Service Management

This section provides the details on service management.

Table 7–4 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

7.5 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.

7.6 Header Record

Not Applicable

7.7 Detail Record

This section provides the information on detail record.

Table 7–5 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

7.8 Translation Rules

Not Applicable

7.9 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.

7.10 Customer Information

This section provides the details on customer information.

Table 7–6 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

7.11 Constraints

Not Applicable