Oracle® Banking Platform Collections

Interface Specification Guide

Release 2.6.2.0.0

E95189-01

May 2018

ORACLE®

Oracle Banking Platform Collections Interface Specification Guide, Release 2.6.2.0.0

E95189-01

Contents

List of Tables

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

This document is intended for the following audience:

- IT Deployment Team
- Consulting Staff
- Administrators

Documentation Accessibility

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

Access to Oracle Support

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

Organization of the Guide

This document contains:

Chapter 1 About this Guide

This chapter provides details about applicability of this guide.

Chapter 2 Introduction

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

Chapter 3 System Overview

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

Chapter 4 Staging Area

This chapter provides details of the feeder tables.

Chapter 5 Algorithms

This chapter outlines the pre-shipped algorithm details.

Chapter 7 Feeder Services

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

Chapter 8 Dialer Webservice Integration

This chapter provides details of the dialer web service integration.

Related Documents

For more information, see the following documentation:

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

This guide is applicable for the following products:

- Oracle Banking Platform
- Oracle Banking Enterprise Collections

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

2 Introduction

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

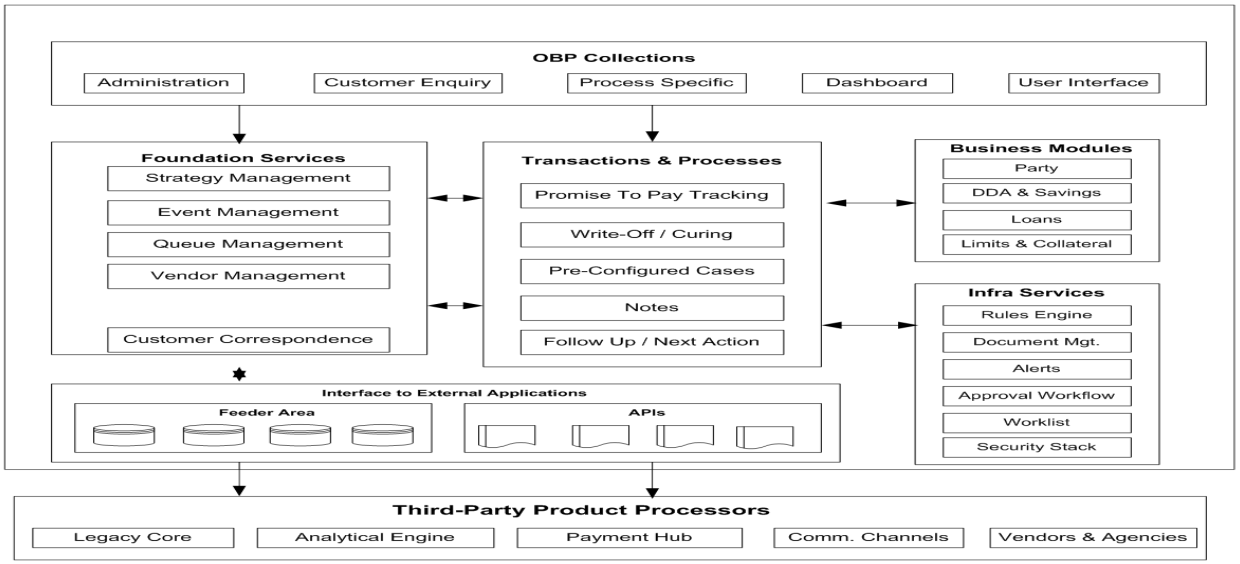
The feeder services exposed by Collections are invoked when changes in data take place in OBP. These services bring modified data into staging tables before batch processes these and update collections tables.

3 System Overview

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

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

Figure 3–1 System Overview



4 Staging Area

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

4.1 Feeder Tables

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

4.1.1 Account Data

This section provides information on the tables related to accounts.

4.1.1.1 Account Details

Table Name: Account Details (CI_FDR_ACCT)

Description: This table holds account related data from host.

Table 4–1 Account Details

Field Name	Description	Values	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Business Unit	Business Unit of the Account. This field is used only if multi-branding features are to be used.		VARCHAR2	40	N	BUSINESS_UNIT
Market Entity	Market Entity to which account belongs. This field is used only if multi-branding features are to be used.		VARCHAR2	40	N	MARKET_ENTITY
Facility ID	Facility ID under which account is created. This field is used based on the structure of accounts in the		VARCHAR2	40	N	FACILITY_ID

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
BICO Loan Switch			CHAR	1	N	BICOE_LOAN_ SW
First Default Date			DATE		N	FIRST_ DEFAULT_DATE
Last DPD Update Date	Last DPD update Date		DATE		N	LAST_DPD_ UPDATE_DT
Relation Officer Code	Relation Officer Code		VARCHAR2	40	N	RELATION_ OFFICER_CODE
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

Field Name	Description	Values	Data Type	Length	Required	Column Name
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
Account Reopen Switch	Account Reopen Switch		NUMBER	1	N	ACCT_REOPEN_SW
Charge Off Primary Reason	Charge Off Primary Reason		VARCHAR2	60		CHARGE_OFF_PRIMARY_RSN
Charge Off Secondary Reason	Charge Off Secondary Reason		VARCHAR2	60		CHARGE_OFF_SECONDARY_RSN
Recovery Score	Recovery Score		VARCHAR2	10		RECOVERY_SCORE
Fee Charge	Fee Charge		NUMBER	36.18		FEE_CHARGES
Insurance Amount	Insurance Amount		NUMBER	36.18		INSURANCE
Interest Amount	Interest Amount		NUMBER	36.18		INTEREST
Principal Amount	Principal Amount		NUMBER	36.18		PRINCIPAL_AMT
Interest Rate	Interest Rate		NUMBER	36.18		INTEREST_RATE
Account Term	Account Term		NUMBER	4		ACCT_TERM
Account Non Due Amount	Account Non Due Amount		NUMBER	36.18		ACT_NON_DUE_AMT

4.1.1.2 Account Arrears Details

Table Name: Account Arrear Details (CI_FDR_ACCT_ARS)

Description: This table holds account arrears data from host.

Table 4–2 Account Arrears Details

Field Name	Description	Values	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_NBR
Host ID	Source Host ID for		VARCHAR2	10	Y	SRC_HOST_ID

Field Name	Description	Values	Data Type	Length	Required	Column Name
	host					
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
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	To check whether		VARCHAR2	1	Y	RECORD_ EXISTS_SW

Field Name	Description	Values	Data Type	Length	Required	Column Name
Exist Switch	record is already available or not					
RES due date	RES due date		DATE	10	N	ARS_DUE_DT
Sub Arrear Type	Sub Arrear Type		VARCHAR2	40	Y	SUB_ARREAR_TYPE
Account Non Due Amount	Account Non Due Amount		NUMBER	22	N	ACT_NON_DUE_AMT
Account Non Due Flag	Account Non Due Flag		VARCHAR2	1	N	ACT_NON_DUE_FLG

4.1.1.3 Account Hardship Details

Table Name: Account Hardship Details (CI_FDR_ACCT_HARDSHIP_DTLS)

Description: This table holds account hardship data from host.

Table 4–3 Account Hardship Details

Field Name	Description	Values	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Application ID	Hardship Application ID		VARCHAR2	40	Y	HARSHIP_APPLICATION_ID
Relief Effective Date	Will be unique per Application ID		DATE	10	Y	RELIEF_EFFECTIVE_DT
Relief Expiry Date	Will be unique per Application ID		DATE	10	Y	RELIEF_EXPIRY_DT
Relief Type(s)	Can be more than 1 per application ID		VARCHAR2	40	Y	RELIEF_TYPE
Number of Payments Waived	Number of Payments Waived		NUMBER	4,0	N	NO_PAYMNT_WAIVED
User's Discretionary Margin (UDM)	These field details will be received only in case of Change Interest Rate relief type.		VARCHAR2	60	N	USR_DISCRTN_MRGN
UDM Start Date	User's discretionary		DATE	10	N	USR_DISCRTN_

Field Name	Description	Values	Data Type	Length	Required	Column Name
	Margin start date for the relief					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 Hardship Relief if applicable		CHAR	60	N	STATUS
Original Relief Type	Original Relief Type		VARCHAR2	40	N	ORIG_RELIEF_ TYPE
Record Creation Date	Date on which the data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether record is already available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW

4.1.1.4 Account Repayment Schedule

Table Name: Account Repayment Schedule (CI_FDR_REPAYMENT_SCH)

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

Table 4–4 Account Repayment Schedule

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_ NBR

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
Interest	Interest component		NUMBER	36,18	N	INTEREST_AMT
Fee	Fee component, if any		NUMBER	36,18	N	FEE_AMT
Balance	Outstanding balance after the installment is paid		NUMBER	36,18	N	PRINCIPAL_BALANCE
Installment Number	Installment number as per the loan structure		NUMBER	5,0	N	INSTALLMENT_NUM
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_UPDATE_DT
Record Exist Switch	To check whether record is already available or not		VARCHAR2	1	Y	RECORD_EXISTS_SW

4.1.1.5 Account Warning Indicator

Table Name: Account Warning Indicator (CI_FDR_ACCT_WARNING_IND)

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

Table 4–5 Account Warning Indicator

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ ACCT_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ ID
Warning Indicator Code	Warning Indicator code as stored in host		VARCHAR2	50	Y	WARN_IND_ CD
Warning Indicator Value	Warning Indicator Value		VARCHAR2	1	N	WARN_IND_ VAL
Start Date	Start Date for warning indicator		DATE	10	N	START_DT
End Date	End Date for the warning indicator code		DATE	10	N	END_DT
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether record is already available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW

4.1.1.6 Minimum Amount Due on Bill (MAD)

Table Name: Minimum Amount Due on Bill (CI_FDR_MIN_AMT_DUE_BILL)

Description: This table holds Billing data from host.

Table 4–6 Minimum Amount Due Bill

Field Name	Description	Value	Data Type	Length	Required	Column Name
Source Host ID	Source Host ID		VARCHAR2	10	Y	SOURCE_ HOST_ID
Host Account Number	Host Account Number		VARCHAR2	40	Y	HOST_ ACCNT_NBR
Bill Due Date	Bill Due Date		DATE		Y	DUE_DATE
Bill Date	Bill Date		DATE		N	BILL_CYCLE
Minimum Due Amount	Minimum Due Amount		NUMBER	36,18	N	MIN_AMT_ DUE
Bill Status on Due Date	Bill Status on Due Date		VARCHAR2	10	N	STATUS_ ON_DUE_ DATE
Total Minimum Amount Due	Total Minimum Amount Due		NUMBER	36,18	N	TOT_MIN_ AMT_DUE
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR

4.1.2 Party Data

This section provides information on the tables related to party.

4.1.2.1 Party Account Relationship

Table Name: Party Account Relationship (CI_FDR_ACCT_PER)

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

Table 4–7 Account Party Relationship

Field Name	Description	Value	Data Type	Length	Required	Column Name
Source Host ID	Source Host ID		VARCHAR2	10	Y	SOURCE_ HOST_ID
Host Account	Host Account		VARCHAR2	40	Y	HOST_

Field Name	Description	Value	Data Type	Length	Required	Column Name
Number	Number					ACCNT_NBR
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
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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 in case any additional attributes are required		VARCHAR2	60	N	UDF5
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
User Defined Field 10	User Defined Field in case any additional attributes are required		VARCHAR2	60	N	UDF10
Account Nickname	Account Nickname		VARCHAR2	120	N	ACCT_ NICKNAME
Corresponde Nomination Switch	Corresponde Nomination Switch		CHAR	1	N	CORRES_ NOMINATION_ SW

4.1.2.2 Party Details

Table Name: Party Details (CI_FDR_PER)

Description: This table holds party data from host.

Table 4–8 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		VARCHAR2	20	N	MARITAL_

Field Name	Description	Value	Data Type	Length	Required	Column Name
	Party in case of Individual Customer					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 party interface		VARCHAR2	60	N	PROBABILITY_ OF_DEFLT_VAL
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	User Defined		VARCHAR2	60	N	UDF1

Field Name	Description	Value	Data Type	Length	Required	Column Name
Value 1	Fields					
User Defined Value 2	User Defined Fields		VARCHAR2	60	N	UDF2
User Defined Value 3	User Defined Fields		VARCHAR2	60	N	UDF3
User Defined Value 4	User Defined Fields		VARCHAR2	60	N	UDF4
User Defined Value 5	User Defined Fields		VARCHAR2	60	N	UDF5
User Defined Value 6	User Defined Fields		VARCHAR2	60	N	UDF6
User Defined Value 7	User Defined Fields		VARCHAR2	60	N	UDF7
User Defined Value 8	User Defined Fields		VARCHAR2	60	N	UDF8
User Defined Value 9	User Defined Fields		VARCHAR2	60	N	UDF9
User Defined Value 10	User Defined Fields		VARCHAR2	60	N	UDF10
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		DATE	10	N	CRET_DTTM

Field Name	Description	Value	Data Type	Length	Required	Column Name
	Collections					
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
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	Service Member Missing on Duty		NUMBER	1	Y	SCRA_MEMBER_MISSING_FLG

Field Name	Description	Value	Data Type	Length	Required	Column Name
DutySCRA						
Service Member Active Dependent	Service Member Active Dependent		NUMBER	1	Y	SCRA_DEP_ ACTIVE_SW
Service Member Updated Switch	Service Member Updated Switch		CHAR	1	N	SCRA_ UPDATE_SW

4.1.2.3 Party Address Details

Table Name: Party Address Details (CI_FDR_PER_ADDR)

Description: This table holds party address data from host.

Table 4–9 Party Address Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Party ID	Party ID as stored in Host		VARCHAR2	40	Y	HOST_CUST_ NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Address Type	Address Type Code maintained in Host	Home, 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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 update for existing data	I - Insert U - Update	VARCHAR2	10	N	RCD_TYPE
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

4.1.2.4 Party Employment Details

Table Name: Party Employment Details (CI_FDR_PER_EMPLOYMENT_PROF)

Description: This table holds party employment details from host.

Table 4–10 Party Employment Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Party ID	Party ID as stored in Host		VARCHAR2	40	Y	HOST_CUST_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Determinant Value	Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features.		VARCHAR2	60	Y	DETERMINANT_VALUE
Sequence ID	Sequence ID of Employment details		VARCHAR2	40	Y	FDR_EMP_SEQ_ID
Employment Status	Employment Status Code	**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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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 Annual Salary	Gross Annual Salary		NUMBER	36,18	N	GRS_ANNUAL_ INCOME
Start Date	Start Date		DATE	10	N	START_DT
End Date	End Date		DATE	10	N	END_DT
Status	Status		VARCHAR2	60	N	STATUS
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether the record is available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW

4.1.2.5 Party Identification Details

Table Name: Party Identification Details (CI_FDR_PER_ID)

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

Table 4–11 Party Identification Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Party ID	Party ID as stored in Host		VARCHAR2	40	Y	HOST_CUST_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Identification Type	Value of Identification Type Code	Passport No, 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
ID_TYPE_ VAL_ STATUS	ID Type Status		VARCHAR2	10	N	ID_TYPE_VAL_ STATUS

4.1.2.6 Party Name Details

Table Name: Party Name Details (CI_FDR_PER_NAME)

Description: This table holds party name details from host.

Table 4–12 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		VARCHAR2	250	N	FDR_FULL_

Field Name	Description	Value	Data Type	Length	Required	Column Name
	customer					NAME
Short Name	Short Name of the customer		VARCHAR2	60	N	FDR_SHORT_ NAME
Determinant Value	Determinant Value for identification of Party. This will depend on setups in host and is used in case of multi-branding features.		VARCHAR2	60	Y	FDR_ DETERMINANT_ VALUE
Primary Name Flag	Signifies if a particular name needs to be used as a primary name for the customer	Y/N	CHAR	1	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 description

4.1.2.7 Party Contact Preference Details

Table Name: Party Contact Preference Details (CI_FDR_CONTACT_PREF)

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

Table 4–13 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	NUMBER	10,0	N	WKDAY_FROM_TM

Field Name	Description	Value	Data Type	Length	Required	Column Name
		format (for example, 1800 for 6:00 PM)				
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
Do Not Disturb Flag	Do Not Disturb Flag		NUMBER	1	Y	DND_FLG
DND Start Date	DND Start Date		DATE		N	DND_START
DND End Date	DND End Date		DATE		N	DND_END
Proffered Flag	Proffered Flag		NUMBER	1	Y	IS_ PREFERRED_ FLAG
Proffered Alert SMS	Proffered Alert SMS		NUMBER	1	Y	IS_ PREFERRED_ FOR_ALERT_ SMS
Permission to Call or Not	Permission to Call or Not		NUMBER	1	Y	IS_ PERMISSION_ CALL
Permission to Record Calls	Permission to Record Calls		NUMBER	1	Y	IS_ PERMISSION_ RECORD_ CALLS
Email Communication Consent Flag	Email Communication Consent Flag		NUMBER	1	Y	ELEC_COMM_ CONSENT
Host Update Flag	Host Update Flag		NUMBER	1	Y	HOST_ UPDATED_FLG
Time Zone	Time Zone		VARCHAR2	50	N	TIME_ZONE
Country Code	Country Code		VARCHAR2	10	N	COUNTRY_CD
Phone Communication Consent	Phone Communication Consent Flag		NUMBER	1	Y	ELEC_COMM_ CONSENT_ PHONE

4.1.2.8 Party Warning Indicators

Table Name: Party Warning Indicators (CI_FDR_PARTY_WARNING_IND)

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

Table 4–14 Party Warning Indicators

Field Name	Description	Value	Data Type	Length	Required	Column Name
Party ID	Party ID as stored in Host		VARCHAR2	40	Y	HOST_ CUST_NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ ID
Warning Indicator Code	Warning Indicator Code		VARCHAR2	50	Y	WARN_IND_ CD
Warning Indicator Value	Value of Warning Indicator Code	Y/N	VARCHAR2	1	N	WARN_IND_ VAL
Start Date	Start Date of Warning Indicator		DATE	10	N	START_DT
End Date	End Date of warning Indicator		DATE	10	N	END_DT
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether the record is available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW

4.1.2.9 Service Member History Details

Table Name: Service Member History Details (CI_FDR_SCRA_HIST_DTLS)

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

Table 4–15 Service Member History Details

Field Name	Description	Value	Data Type	Length	Required	Column Name
Source Host ID	Source Host ID		VARCHAR2	10	Y	SOURCE_ HOST_ID
Party ID	Party ID		VARCHAR2	40	Y	HOST_CUST_ NBR
Determinant Value	Determinant Value		VARCHAR2	50	Y	DETERMINANT_ VALUE
Service Member Order Number	Service Member Order Number		VARCHAR2	50	Y	SCRA_ORDER_ NUM
Service Member Notification Date	Service Member Notification Date		DATE		Y	SCRA_ NOTIFICATION_ DT
Active Duty Start Date for Service Member	Active Duty Start Date for Service Member		DATE		Y	SCRA_START_ DT_OF_ ACTIVE_SVC
Unit name of Service Member	Unit name of Service Member		VARCHAR2	50	Y	SCRA_UNIT_ NAME
End Date of Active Duty of Service Member	End Date of Active Duty of Service Member		DATE		N	SCRA_END_ DT_OF_ ACTIVE_SVC
Service Member Order Status	Service Member Order Status		VARCHAR2	50	N	SCRA_ORDER_ STATUS
Court Order Start Date	Court Order Start Date		DATE		N	COURT_ ORDER_ START_DATE
Court Order End Date	Court Order End Date		DATE		N	COURT_ ORDER_END_ DATE
Court Order Applicable Switch	Court Order Applicable Switch		NUMBER	22		IS_CURT_ ORDER_APP_ SW
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Court Order Applicable Switch	Court Order Applicable Switch		NUMBER	22		IS_CURT_ ORDER_APP_ SW

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

4.1.3 Collateral Data

This section provides information on the tables related to collaterals.

4.1.3.1 Collateral Details

Table Name: Collateral Details (CI_FDR_COLLATERAL)

Description: This table holds collateral data from host.

Table 4–16 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

4.1.3.2 Collateral Charge Details

Table Name: Collateral Charge Details (CI_FDR_COLLATERAL_CHRG)

Description: This table holds collateral charges details from host.

Table 4–17 Collateral Charges Details

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

Field Name	Description	Value	Data Type	Length	Required	Column Name
	is update for existing data					
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether the record is available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW
Registration Number	Registration Number		VARCHAR2	20	N	CHARGE_ REG_NUM
Charge Status	Charge Status		VARCHAR2	60	N	CHARGE_ STATUS

4.1.3.3 Collateral Entity Mapping

Table Name: Collateral Entity Mapping (CI_FDR_COLLATERAL_ENTITY)

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

Table 4–18 Collateral Entity Mapping

Field Name	Description	Value	Data Type	Length	Required	Column Name
Collateral Code	Collateral Code as stored in host		VARCHAR2	40	Y	COLLATERAL_ CD
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Entity Type	Entity to which collateral is mapped	ACCOUNT, FACILITY	VARCHAR2	10	Y	ENTITY_TYPE
Entity ID	Entity ID of entity to which collateral is mapped		VARCHAR2	40	Y	COL_ENTITY_ID
Contribution Switch	Identify if the collateral is contributing towards an entity	Y/N	VARCHAR2	1	N	FDR_LIMIT_ CONTRIBUTION_ SW

Field Name	Description	Value	Data Type	Length	Required	Column Name
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_CAT_ NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether the record is available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW
Charge Code	Charge Codes maintained in the host		VARCHAR2	20	N	CHRG_CD

4.1.3.4 Collateral Guarantor Mapping

Table Name: Collateral Guarantor Mapping (CI_FDR_COLLATERAL_GRNTR)

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

Table 4–19 Collateral Guarantor Mapping

Field Name	Description	Value	Data Type	Length	Required	Column Name
Collateral Code	Collateral Code as stored in host		VARCHAR2	40	Y	COLLATERAL_ CD
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ ID
Party ID	Party ID of the guarantor		VARCHAR2	40	Y	HOST_CUST_ NBR
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM

Field Name	Description	Value	Data Type	Length	Required	Column Name
Record Type	Signifies if the data is created initially or is update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_ NBR
Record Update Date	Date on which the record is updated		DATE	10	N	RECORD_ UPDATE_DT
Record Exist Switch	To check whether the record is available or not		VARCHAR2	1	Y	RECORD_ EXISTS_SW

4.1.3.5 Collateral Owner Mapping

Table Name: Collateral Owner Mapping (CI_FDR_COLLATERAL_PARTY)

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

Table 4–20 Collateral Owner Mapping

Field Name	Description	Value	Data Type	Length	Required	Column Name
Collateral Code	Collateral Code as stored in host		VARCHAR2	40	Y	COLLATERAL_ CD
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ ID
Party ID	Party ID of Customer mapped to collateral		VARCHAR2	40	Y	HOST_CUST_ NBR
Percentage of Ownership	Ownership Percentage of each of the Party		VARCHAR2	10	N	OWNERSHIP_ PERCENT
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Record Type	Signifies if the data is created initially or is an update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process Status	To check the current status of the process. Default is		VARCHAR2	1	Y	PROCESS_ STATUS

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

4.1.3.6 Collateral Vehicle Mapping

Table Name: Collateral Vehicle Mapping (CI_FDR_COLLATERAL_AUTOMOBILE)

Description: This table holds Vehicle information for the collateral.

Table 4–21 Collateral Vehicle Mapping

Field Name	Description	Value	Data Type	Length	Required	Column Name
Vehicle Identification Number	Vehicle Identification Number		VARCHAR2	30	Y	VHCL_ IDENTIFY_NO
Vehicle Make	Vehicle Make		VARCHAR2	20	N	VHCL_MAKE
Vehicle model	Vehicle model		VARCHAR2	20	N	VHCL_MODEL
Vehicle Trim	Vehicle Trim		VARCHAR2	20	N	VHCL_TRIM
Manufacturing Year	Manufacturing Year		VARCHAR2	5	N	MANUFACTUR_ YEAR
Licence Plate Zip Code	Licence Plate Zip Code		VARCHAR2	10	N	LICNC_PLT_ ZIPCODE
Licence Plate Number	Licence Plate Number		VARCHAR2	20	N	LICNC_PLT_ NUMBER
Licence Plate State	Licence Plate State		VARCHAR2	20	N	LICNC_PLT_ STATE
Source Host Id	Source Host Id		VARCHAR2	10	Y	SRC_HOST_ID
Record Creation Date	Record Creation Date		DATE		N	CRET_DTTM
Record Type	Signifies if the data is created initially or is an update for existing data	I - Insert U - Update	VARCHAR2	10	Y	RCD_TYPE
Process	To check the		VARCHAR2	1	Y	PROCESS_

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

4.1.4 Insurance Data

This section provides information on the tables related to insurance.

4.1.4.1 Insurance Details

Table Name: Insurance Details (CI_FDR_INSR_DTLS)

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

Table 4–22 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

4.1.5 Payment Data

This section provides information on the tables related to payments.

4.1.5.1 Online Payment Records

Table Name: Online Payment (CI_FDR_PAYMENTS)

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

Table 4–23 Online Payment

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account No	Account Number as stored in Host		VARCHAR2	40	Y	HOST_ACCT_ NBR
Host ID	Source Host ID for host		VARCHAR2	10	Y	SRC_HOST_ID
Transaction Reference Number	Transaction Reference Number for payment transaction in host		VARCHAR2	30	Y	XREF_NO
Transaction Date	Date of Transaction		DATE	10	N	FDR_ TRANSACTION_ DT
Transaction Time	Time for Transaction		DATE	10	N	FDR_ TRANSACTION_ TM
Value Date	Value Date on which the transaction was posted in the host		DATE	10	N	FDR_VALUE_DT
Transaction Currency	Currency code of the transaction		VARCHAR2	3	N	FDR_ TRANSACTION_ CURR_CD

Field Name	Description	Value	Data Type	Length	Required	Column Name
Transaction Amount	Payment Amount		NUMBER	36,18	N	FDR_ TRANSACTION_ AMT
Account Currency	Account Currency Code		VARCHAR2	3	N	FDR_ACCT_ CURR_CD
Account Balance	Account Balance after Payment		NUMBER	36,18	N	FDR_ACCT_ AMT
Transaction Code	Transaction Code as captured in the host		VARCHAR2	30	N	FDR_ TRANSACTION_ CD
Narration Text	Narration text for the transaction		VARCHAR2	120	N	FDR_ NARRATION_ TXT
Transaction Type Flag	Identify if the transaction is Credit or Debit that is, actual payment transaction or reversal	C/D	CHAR	1	Y	FDR_ TRANSACTION_ TYPE_FLG
Record Creation Date	Date on which data is fed to Collections		DATE	10	N	CRET_DTTM
Original Transaction ref number	Used for cancellation of payments		VARCHAR2	30	N	ORIG_XREF_NO
Transaction sequence number	Transaction sequence number		VARCHAR2	30	Y	FDR_XREF_ SUB_SEQ_NO
Original Transaction sequence number	Used for cancellation of payments		VARCHAR2	30	N	FDR_ORIG_ XREF_SUB_ SEQ_NO
Process Status	To check the current status of the process. Default is P-Pending.		VARCHAR2	1	Y	PROCESS_ STATUS
Message Category Number	Defined error message category		NUMBER	5,0	Y	MESSAGE_ CAT_NBR
Message Number	Error message number		NUMBER	5,0	Y	MESSAGE_NBR

4.2 Interfacing Tables

This section provides details about the Interfacing tables.

4.2.1 Agency or Vendor Upload

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

4.2.1.1 Upload Followup Table Details

Table Name: Vendor/Agency Upload Follow up Table (CI_VNDR_UPLD_FOLLOWUP)

Description: This table holds Follow up Upload data.

Table 4–24 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
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

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

4.2.1.2 Upload Result Table Details

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

Description: This table holds Result Upload data.

Table 4–25 Upload Result Table

Field Name	Description	Value	Data Type	Length	Required	Column Name
Vendor Upload Result ID	Vendor Upload Result ID		CHAR	10	Y	VNDR_ UPLD_ RESULT_ID
Vendor Upload Followup ID	Vendor Upload Followup ID		CHAR	10	Y	VNDR_ UPLD_ FLWUP_ID
Result Type Code	Result Type Code		CHAR	12	Y	RESULT_ TYPE_CD
Collector Comments	Collector Comments		VARCHAR2	2000	N	RESULT_ COL_ COMMENTS
Result Date	Result Date		DATE		Y	RESULT_ DTTM
Primary Result Switch	Primary Result Switch		CHAR	1	Y	PRIM_ RESULT_SW

4.2.1.3 Upload PTP Table Details

Table Name: Vendor/Agency Upload Promise To Pay(PTP) Table (CI_VNDR_UPLD_PTP)

Description: This table holds PTP Upload data.

Table 4–26 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
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	Message Number		NUMBER	5	N	MESSAGE_

Field Name	Description	Value	Data Type	Length	Required	Column Name
Number						NBR
Batch Run Date	Batch Run Date		DATE		N	BATCH_ RUN_DTTM

4.2.1.4 Upload PTP Schedule Table Details

Table Name: Vendor/Agency Upload Promise To Pay(PTP) Schedule Table (CI_VNDR_UPLD_PTP_ SCHED)

Description: This table holds PTP Schedule Upload data.

Table 4–27 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 Schedule Date	PTP Schedule Date		DATE		Y	PP_SCHED_ DT
PTP Schedule Amount	PTP Schedule Amount		NUMBER	36,18	Y	PP_SCHED_ AMT
Currency Code	Currency Code		CHAR	3	Y	CURRENCY_ CD
Pay Clear ID	Pay Clear ID		CHAR	12	N	APAY_CLR_ ID

4.2.2 Dialer Results Upload

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

4.2.2.1 Upload Dialer Result Table Details

Table Name: Dialer Result Upload Batch (CI_DIALER_RESULTS_UPLOAD)

Description: This table holds Dialer Result Upload data.

Table 4–28 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
Channel Type	Channel Type		CHAR2	4	Y	CHANNEL_ TYPE
Account Number	Account Number		VARCHAR2	40	N	HOST_ACCNT_ NBR
Customer Number	Customer Number		VARCHAR2	40	Y	HOST_CUST_ NBR
Source Host ID	Source Host ID		VARCHAR2	10	Y	SOURCE_ HOST_ID
Termination code	Termination code		CHAR	12	Y	TERMINATION_ CD
Status Code	Status Code		CHAR	12	Y	STATUS_CD
Enterprise Customer Number	Enterprise Customer Number		VARCHAR2	60	N	ENTERPRISE_ CUST_NBR
Attempts	Attempts		NUMBER	5	Y	ATTEMPTS
Call Date	Call Date		DATE		N	CALL_DTTM
Campaign ID	Campaign ID		CHAR	10	Y	CAMPAIGN_ID
Dialer Contact ID	Dialer Contact ID		CHAR	10	Y	DIALER_ CONTACT_ID
Contact Number	Contact Number		NUMBER	20	N	CONTACT_ NUMBER
Record Creation Date	Record Creation Date		DATE		N	CRE_DTTM
UDF1	User Defined Fields		VARCHAR2	60	Y	UDF1
UDF2	User Defined Fields		VARCHAR2	60	Y	UDF2
UDF3	User Defined Fields		VARCHAR2	60	Y	UDF3
UDF4	User Defined Fields		VARCHAR2	60	Y	UDF4
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
UDF9	User Defined Fields		VARCHAR2	60	Y	UDF9
UDF10	User Defined Fields		VARCHAR2	60	Y	UDF10
UDF11	User Defined Fields		VARCHAR2	60	Y	UDF11
UDF12	User Defined Fields		VARCHAR2	60	Y	UDF12
UDF13	User Defined Fields		VARCHAR2	60	Y	UDF13
UDF14	User Defined Fields		VARCHAR2	60	N	UDF14
UDF15	User Defined Fields		VARCHAR2	60	Y	UDF15
UDF16	User Defined Fields		VARCHAR2	60	Y	UDF16
UDF17	User Defined Fields		VARCHAR2	60	Y	UDF17
UDF18	User Defined Fields		VARCHAR2	60	Y	UDF18
UDF19	User Defined Fields		VARCHAR2	60	Y	UDF19
UDF20	User Defined Fields		VARCHAR2	60	Y	UDF20
UDF21	User Defined Fields		VARCHAR2	60	Y	UDF21
UDF22	User Defined Fields		VARCHAR2	60	Y	UDF22
UDF23	User Defined Fields		VARCHAR2	60	Y	UDF23
UDF24	User Defined Fields		VARCHAR2	60	Y	UDF24
UDF25	User Defined 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		VARCHAR2	60	Y	UDF29

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

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

4.2.3 Account Dialer Extract

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

4.2.3.1 Dialer Extract Account Table Details

Table Name: Account Dialer Extract Table (CI_ACCT_DILR_EXTRCT)

Description: This table holds Account Dialer Extract data.

Table 4–29 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
Account Relation Type Code	Account Relation Type Code		VARCHAR2	10	Y	ACCT_REL_ TYPE_CD
Dialer Extract Channel Type	Dialer Extract Channel Type		VARCHAR2	40	Y	DILREXTCT_ CHANNEL_ TYPE
Dialer Extract filter ID	Dialer Extract filter ID		VARCHAR2	40	Y	DILREXTCT_ FILTER_ID
Campaign Description	Campaign Description		VARCHAR2	40	N	CAMPAIGN_ DESCR
Campaign Priority	Campaign Priority		VARCHAR2	40	N	CAMPAIGN_ PRIORITY
Dialer Extract Status	Dialer Extract Status		VARCHAR2	20	N	DILREXTCT_ STATUS
Dialer Extract Termination Code	Dialer Extract Termination Code		VARCHAR2	20	N	DILREXTCT_ TERMINATION_ CD
Exclude Reason Code	Exclude Reason Code		VARCHAR2	20	N	EXCLUDE_ REASON_CODE
Extract Date	Extract Date		DATE		N	EXTRACT_ DTTM
Next Display Date	Next Display Date		DATE		N	NEXT_ DISPLAY_DATE
Dialer Extract File Name	Dialer Extract File Name		VARCHAR2	400	N	DILREXTCT_ FILE_NAME
Queue Code	Queue Code		CHAR	10	N	QUEUE_CD
Host Product Group Code	Host Product Group Code		VARCHAR2	30	N	HOST_PROD_ GRP_CD
Host Product Code	Host Product Code		VARCHAR2	30	N	HOST_PRD_CD
Overdue Amount	Overdue Amount		NUMBER	36,18	Y	OVERDUE_AMT
Outstanding 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	Suspended Switch		CHAR	1	Y	SUSPEND_SW

Field Name	Description	Value	Data Type	Length	Required	Column Name
Switch						
Full Name	Full Name		VARCHAR2	400	N	FULL_NAME
Customer Prefix	Customer Prefix		VARCHAR2	40	N	CUST_PREFIX
First Name	First Name		VARCHAR2	200	N	FIRST_NAME
Last Name	Last Name		VARCHAR2	200	N	LAST_NAME
Customer Suffix	Customer Suffix		VARCHAR2	40	N	CUST_SUFFIX
Address Type Code	Address Type Code		VARCHAR2	20	N	ADDR_TYPE_ CD
Address Line1	Address Line1		VARCHAR2	400	N	ADDRESS_LN1
Address Line2	Address Line2		VARCHAR2	400	N	ADDRESS_LN2
Address Line3	Address Line3		VARCHAR2	400	N	ADDRESS_LN3
Address Line4	Address Line4		VARCHAR2	400	N	ADDRESS_LN4
City	City		VARCHAR2	40	N	CITY
Country	Country		VARCHAR2	40	N	COUNTRY
State	State		VARCHAR2	40	N	STATE
Postal Code	Postal Code		VARCHAR2	40	N	POSTAL
Birthdate	Birthdate		DATE		N	BIRTH_DT
Next Action Time	Next Action Time		VARCHAR2	8	N	NEXT_ACTION_ TIME

4.2.3.2 Dialer Extract Contact Table Details

Table Name: Dialer Extract Contact Table (CI_DIALER_EXTRACTS_CONTACT)

Description: This table holds Dialer Extract Contact data.

Table 4–30 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

Field Name	Description	Value	Data Type	Length	Required	Column Name
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
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

Field Name	Description	Value	Data Type	Length	Required	Column Name
Electronic Communication Consent Switch	Electronic Communication Consent Switch		CHAR	1	N	ELEC_COMM_ CONSENT_SW

4.3 OBP Views

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

4.3.1 Main Account Views

The main account views are as follows:

- FLX_COL_ACCT_DATA_XF
- FLX_LN_COL_FD_ACCT_VW
- FLX_DD_COL_DATA_TOD_XF_VW
- FLX_DD_COL_DATA_XF_VW
- FLX_AC_COL_FD_ACCT_ARS_VW
- FLX_LN_COL_FD_SCH_VW
- FLX_COL_ACCT_WARN_IND_DATA_XF
- FLX_DD_COL_BILL_DATA_XF_VW

4.3.2 Account Updateable Views

The account updateable views are as follows:

- FLX_LN_COL_ACCT_UPDATE_VW
- FLX_DD_COL_DATA_XF_UPD_ACCT_VW
- FLX_DD_COL_DATA_XF_UPD_EXTN_VW

4.3.3 Hardship Views

The hardship views are as follows:

- FLX_COL_ACCT_HRDSHIP_VW
- FLX_LN_COL_ACCT_HRDSHIP_VW
- FLX_DD_COL_ACCT_HRDSHIP_VW

4.3.4 Party Views

The party views are as follows:

- FLX_PI_COL_FD_ACCT_PER_VW
- FLX_PI_COL_FD_PER_VW
- FLX_PI_COL_FD_PARTY_IDENT_VW

- FLX_PI_COL_FD_PER_NAME_VW
- FLX_PI_COL_FD_PER_WARN_IND_VW
- FLX_PI_COL_FD_EMP_PROF_VW
- FLX_PI_COL_FD_PER_ADDR_VW
- FLX_PI_COL_FD_CONTACT_PREF_VW

4.3.5 LCM / Collateral Views

The LCM / Collateral views are as follows:

- FLX_LM_COL_FD_COL_ENTITY_VW
- FLX_LM_COL_FD_COLLATERAL_VW
- FLX_LM_COL_FD_COL_PARTY_VW
- FLX_LM_COL_FD_COL_CHRG_VW
- FLX_LM_COL_FD_COL_GRNTR_VW
- FLX_LM_COL_FD_INSR_DTLS_VW
- FLX_LM_COL_FD_COL_ADDR_VW
- FLX_LM_COL_FD_COL_AUTO_VW

5 Algorithms

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

5.1 Stop Contract: C1-CURENTITY

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

Table 5–1 Stop Contract: C1-CURENTITY

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

5.2 Cure Account: C1-FINCOLL

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

Table 5–2 Cure Account: C1-FINCOLL

Description	This algorithm is used to invoke the OBP Services when contract is stopped during the finalize collection process.
Detailed Description	This algorithm type is used to invoke the OBP Services to update the delinquent flag=N when the contract is stopped during the finalize collection process.
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	This algorithm invokes the OBP Services to update the delinquent flag =N and In collection

Design	flag = N in host (updateInCollectionIndicator()) when the contract is stopped during the final collection process. It also deletes the account review date from CI_ADM_RVW_SCH table, and updates the number of times an account is self-cured.

Table 5–3 Cure Account: Sample Algorithm

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

5.3 Queue Allocation: C1-ALLOCQUEU

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

Table 5–4 Queue Allocation: C1-ALLOCQUEU

Description	Allocation algorithm for allocation cases to queue in round-robin method.
Detailed Description	This is an allocation algorithm for the allocation group to allocate cases to queues in round-robin method. This algorithm is invoked by the Allocation monitor batch (C1-ALOCM).
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
Detailed Design	This algorithm receives input as Allocation Group code from the batch. The view used to filter cases is accepted as an algorithm soft parameter. Product will ship CI_ALLOCATION_MONITOR_VW view. For the given allocation group code, it allocates cases to linked queues of the allocation group in round-robin method. For detailed process, see batch process (C1-ALOCM).

Table 5–5 Queue Allocation: Sample Algorithm

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

5.4 Update Customer Switch: C1-CUSTSW

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

Table 5–6 Update Customer Switch: C1-CUSTSW

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

Table 5–7 Update Customer Switch: Sample Algorithm

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

5.5 Update Legal/Repo Switch: C1-LEREPOCT

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

Table 5–8 Update Legal/Repo Switch: C1-LEREPOCT

Description	This algorithm is used to update Legal and Repo case status on enter processing.
Detailed	Legal Repo Switch Name: Specify the Legal or Repo case switch column name of

Description	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 5–9 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

5.6 User Allocation - Round Robin: C1-USRALCRR

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

Table 5–10 User Allocation - Round Robin: C1-USRALCRR

Description	This algorithm is used to allocate cases to users or teams in round-robin method.
Detailed	This algorithm is used to allocate cases to user or teams in round-robin method. This

Description	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.

5.7 User Allocation - % Based: C1-USRALCPR

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

Table 5–11 User Allocation - % Based: C1-USRALCPR

Description	This algorithm is used for allocating cases to users or teams in percentage-based method.

<table>
<tr><td>Detailed Description</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).</td></tr>
<tr><td>Algorithm Entity</td><td>User Allocation</td></tr>
<tr><td>Program Type</td><td>java</td></tr>
<tr><td>Program Name</td><td>com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPerBasedAllocRoundOff</td></tr>
<tr><td>Parameters</td><td>NA</td></tr>
<tr><td>Detailed Design</td><td>

Open unallocated cases will be allocated to valid queue users based on corresponding percentage distribution value mentioned in the Queue Details screen.
Algorithm will round off the decimal percent allocated case count to the nearest whole integer value.

For example:
User allocation percentage of Queue Q1 is as follows:

Table 5–12 User allocation percentage of Queue Q1
<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:

Table 5–13 Calculations for allocating cases
<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>
<tr><td>U2</td><td>33%</td><td>3.3</td><td>3</td></tr>
<tr><td>U3</td><td>34%</td><td>3.4</td><td>3</td></tr>
</table>
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:
</td></tr>
</table>

Table 5–14 Final Distribution

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

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

For example:

User allocation percentage of Queue Q2 is as follows:

Table 5–15 User allocation percentage of Queue Q2

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

Total unallocated cases = 10

Then, cases will be allocated as per following calculations:

Table 5–16 Calculations for allocating cases

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

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 5–17 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.

5.8 Vendor Allocation - Round Robin: C1-VENALCRR

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

Table 5–18 Vendor Allocation - Round Robin: C1-VENALCRR

Description	This algorithm is used for allocating cases to vendors in round-robin method.
Detailed Description	This algorithm allocates cases to vendors in round-robin method. This algorithm is invoked from the User Allocation batch (C1-USALC).
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.

5.9 Vendor Allocation - % Based: C1-VENALCPR

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

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

Description	This algorithm is used for allocating cases to vendors in percentage-based method.
Detailed Description	This algorithm allocates cases to vendors in percentage-based method. This algorithm is invoked from the User Allocation batch (C1-USALC).

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.

5.10 Bulk Contact Creation: C1-BLKCNTCRE

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

Table 5–20 Bulk Contact Creation: C1-BLKCNTCRE

Description	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).
Algorithm Entity	Bulk contact creation
Program Type	java
Program Name	com.splwg.ccb.domain.collection.batch.algorithm.BulkContactCreationAlgoComp
Parameters	NA
Detailed Design	This algorithm will be invoked from bulk contact creation batch from where the hard parameter values are set. The algorithm will call business service 'C1-GenerateCorrespondence'. addMultiple() method of 'C1-GenerateCorrespondence' will be called which in turn adds customer contact to CI_CC via add () method of the same service.

5.11 Cross Strategy Action Matrix: C1-CSAM

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

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

Description	This algorithm is used for Cross Strategy Action Matrix
Detailed Description	
Algorithm Entity	Case Type- Enter status
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 5–22 Cross Strategy Action Matrix: Sample Algorithm

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

5.12 Last Payment for Account: C1-PAYDTAMTU

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

Table 5–23 Last Payment for Account: C1-PAYDTAMTU

Description	This algorithm is used to update last payment date and amount in account extension table.
Detailed Description	This algorithm will be invoked on FT freeze algorithm spot and will update Last Payment date and amount in account extension table.
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.

5.13 Association Review Check: C1-ASORVCHK

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

Table 5–24 Association Review Check: C1-ASORVCHK

Description	This algorithm is used to check if association review is required.
Detailed Description	This is to decide if the user should review the system association of entities or not. If Association Review is Required - Stay in current status for user review. Set display date to current business date. If association Review is not required then transition to specified next status.
Algorithm Entity	Case Enter Validation
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal
Parameters	**Name:** NextStatus **Required (Yes/No):** N **Description:** NA **Name:** AssociationReviewRequired **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in the pending status of Legal Process. It decides whether the user should review the system association of entities or not. 'Y' in the algorithm parameter specifies that Association review is required.

Table 5–25 Association Review Check: Sample Algorithm

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

5.14 Validate Expired Default Notice: C1-DEFNOEXP

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

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

Description	This algorithm is used to validate expired default notices.
Detailed Description	This algorithm returns an error if there is no default notice on a given account or a default notice has not yet expired.
Algorithm	Case Type - Enter Validation

Entity	
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.DefaultNoticeExpiryCheck
Parameters	**Name:** associationType **Required (Yes/No):** Y **Description:** NA **Name:** validationfailureOption **Required (Yes/No):** Y **Description:** NA **Name:** toDoType **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked in the pending status of the Legal Process case. It checks if the default notice has expired for a particular account.

Table 5–27 Validate Expired Default Notice: Sample Algorithm

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

5.15 Associate Related Entity: C1-ASSOENTY

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

Table 5–28 Associate Related Entity: C1-ASSOENTY

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

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

Table 5–29 Associate Related Entity: Sample Algorithm

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

5.16 Validate Legal Case Exists: C1-CHKLGL

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

Table 5–30 Validate Legal Case Exists: C1-CHKLGL

Description	This algorithm is used to validate if an active legal case exists at the same time.
Detailed Description	This algorithm checks if a legal case is already running on the primary account any account in the collection with the same owner.
Algorithm Entity	Case Enter Validation
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase
Parameters	**Name:** Case Category **Required (Yes/No):** Y **Description:** NA

	Name: toDoType **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in the pending state of the Legal Process case. It checks if there is any legal case running on the primary account or its related entities.

Table 5–31 Validate Legal Case Exists: Sample Algorithm

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

5.17 Assign New LSP: C1-ASGNLSP

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

Table 5–32 Assign New LSP: C1-ASGNLSP

Description	This algorithm is used to assign LSP to the case.
Detailed Description	This algorithm assigns the LSP to the case either automatically or let the user assign manually depending on the value entered in the algorithm parameters.
Algorithm Entity	Case 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:** NA **Name:** Change LSP Allocation Option **Required (Yes/No):** N **Description:** NA **Name:** Reset Document Submission Date **Required (Yes/No):** N **Description:** NA **Name:** Previous Allocation Check **Required (Yes/No):** N

	Description: NA **Name:** Next Status **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked in the Assign New LSP status of the Legal Process case. Depending on the different algorithm parameter values, the LSP is assigned automatically or manually (both in cases of First time assignment or change assignment).

Table 5–33 Assign New LSP: Sample Algorithm

Algorithm Name	C1-ASGNLSP
Parameters	**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

5.18 Check Approval Requirement: C1-APPRCHK

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

Table 5–34 Check Approval Requirement: C1-APPRCHK

Description	This algorithm is used to check the need of approval.
Detailed Description	This algorithm checks if LSP assignments should be approved.
Algorithm Entity	Case Type - Enter Processing
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CreateApprovalRequest
Parameter	**Name:** Exposure Threshold

s	**Required (Yes/No):** N **Description:** NA **Name:** Approval Request Status **Required (Yes/No):** N **Description:** NA **Name:** Approved Status **Required (Yes/No):** N **Description:** NA **Name:** Reject Request Status **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked in the Prepare Legal Documents status of the Legal Process Case. It checks if the approval is required for the LSP assignment depending on the algorithm parameter values. It also decides where to transit the case.

Table 5–35 Check Approval Requirement: Sample Algorithm

Algorithm Name	C1-ASGNLSP
Parameters	**Name:** Exposure Threshold **Value:** 10 **Name:** Approval Request Status **Value:** PENDINGAPP **Name:** Approved Status **Value:** WTFRLSPACK **Name:** Reject Request Status **Value:** ASSNEWLSP

5.19 Save the Status Before Change LSP: C1-SAVESTATUS

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

Table 5–36 Save the Status Before Change LSP: C1-SAVESTATUS

Description	This algorithm is used to save the status before the status changes in LSP.
Detailed Description	This algorithm saves the status from where it came to Change LSP status. This will be stored in CI_LSP_DTLS table.
Algorithm Entity	Case Type-Enter Processing
Program Type	java

Program Name	com.splwg.ccb.domain.collection.process.legal.ResumeStatusLSP
Parameters	NA
Detailed Design	It is invoked in the Change or Retire LSP status of the Legal Process Case. It stores the previous state of the case so that it returns to that state after the LSP for the case is changed.

5.20 Resume Status from Previous LSP: C1-RESSTATUS

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

Table 5–37 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	Customer class - FT Freeze
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.ResumeStatusLSP
Parameters	NA
Detailed Design	It is invoked in the Legal in Progress status of the Legal Process Case. It resumes the status where the case was previously in before changing the LSP for the case.

5.21 Check Submission Date: CI_CHKSUBDT1

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

Table 5–38 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 5–39 Check Submission Date: Sample Algorithm

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

5.22 Update LSP (CLOS): C1-LSPSTATUS

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

Table 5–40 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.
Algorithm Entity	Case Type-Enter Status
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssignment
Parameters	**Name:** Lsp Assignment Status **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in the Complete, Withdraw status of the Legal Process case. This algorithm updates the end date and assignment status of the CI_LSP_DTLS table after the Legal case is either completed or withdrawn.

Table 5–41 Update LSP (CLOS): Sample Algorithm

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

5.23 Update LSP (CANCEL): C1-LSPSTACAN

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

Table 5–42 Update LSP (CANCEL): C1-LSPSTACAN

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

Table 5–43 Update LSP (CANCEL): Sample Algorithm

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

5.24 Validate Expired Default Notice: C1-DEFNOEXP

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

Table 5–44 Validate Expired Default Notice: C1-DEFNOEXP

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

Entity	
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.DefaultNoticeExpiryCheck
Parameters	**Name:** AssociationType **Required (Yes/No):** Y **Description:** NA **Name:** Validationfailure Option **Required (Yes/No):** Y **Description:** NA **Name:** To Do Type **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked in the Assign New LSP status of the Legal Process case. It checks if the default notice has expired for a particular account.

Table 5–45 Validate Expired Default Notice: Sample Algorithm

Algorithm Name	C1-DEFNOTEXP
Parameters	**Name:** Association Type **Value:** P **Name:** Validation failure Option **Value:** N **Name:** To Do Type **Value:** C1-TD-DN

5.25 Collateral Verification: C1-VRFYCOLS

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

Table 5–46 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:** NA
Detailed Design	It is invoked in the Pending status of the Asset Repossession Process case. It Verifies the collateral associated with account.

Table 5–47 Collateral Verification: Sample Algorithm

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

5.26 Account Association for Asset Repossession Case: C1-ARSACCTS

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

Table 5–48 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.

5.27 Customer Association for Asset Repossession Case: C1-ARSCUSTS

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

Table 5–49 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.

5.28 Update Collateral Property: C1-UPCOLPROP

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

Table 5–50 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.

5.29 Close To do's Algorithm: C1-CLSTODO

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

Table 5–51 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:** NA **Name:** To Do Type2 **Required (Yes/No):** N **Description:** NA

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

Table 5–52 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:**

5.30 Validations for Mandatory Characteristics: C1-CHARVALS

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

Table 5–53 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:** 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 5–54 Validations for Mandatory Characteristics: Sample Algorithm

Algorithm Name	C1-CHARVALS
Parameters	**Name:** ReferenceCharacteristicsValue **Value:** Type of Possession **Name:** ReferenceCharacteristic

	Value: Voluntary Possession **Name:** CaseCharacteristics1 **Value:** Vacancy Date **Name:** CaseCharacteristics2 **Value:** Vacancy Possession Indemnity Policy Reference **Name:** CaseCharacteristics3 **Value:** Property Surrender Letter Reference **Name:** CaseCharacteristics4 **Value:** Property Surrender Letter Reference **Name:** CaseCharacteristics5 **Value:**

5.31 Validations for Mandatory Characteristics: C1-CHARVALA

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

Table 5–55 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 5–56 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:**

5.32 Update Collateral Status in the Host: C1-UPCOLLSTX

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

Table 5–57 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:** NA **Name:** Collateral Status **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in Effected Possession status of the Asset Repossession Process case. This process will update the collateral status in the host.

Table 5–58 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

5.33 Initiate Collateral Valuation: C1-COLLVALX

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

Table 5–59 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-Enter Status
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation
Parameters	**Name:** To Do Type **Required (Yes/No):** Y **Description:** NA **Name:** Days Since Closure Of Last To Do **Required (Yes/No):** Y **Description:** NA **Name:** Assessment Expiry Days **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked while exiting from Pending status of the Asset Repossession Process case. This process will close all To-Do's of "Asset repossession No activity" To-do types associated with the case.

Table 5–60 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

5.34 Close To do's Algorithm: C1-CLSTODO

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

Table 5–61 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:** NA **Name:** To Do Type2 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type3 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type4 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type5 **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked while exiting from Sale In-Progress status of the Asset Repossession Process case. This process will close all To-Do's of "No activity" To-do types associated with the case.

Table 5–62 Close To do's Algorithm: Sample Algorithm

Algorithm Name	C1-CLSTODO
Parameters	**Name:** To Do Type1 **Value:** C1-LNA1 **Name:** To Do Type2 **Value:** C1-LNA1

	Name: To Do Type3 **Value:** C1-TD-CV **Name:** To Do Type4 **Value:** **Name:** To Do Type5 **Value:**

5.35 Validations for Mandatory Characteristics: C1-CHARVALS

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

Table 5–63 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:** 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 Settlement status of the Asset Repossession Process case. This algorithm will carry out subjective validation based on the type of input.

Table 5–64 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:** Contactor Details **Name:** CaseCharacteristics2 **Value:** Conveyance Details **Name:** CaseCharacteristics3 **Value:** **Name:** CaseCharacteristics4 **Value:** **Name:** CaseCharacteristics5 **Value:**

5.36 Update Collateral Status in the Host: C1-UPCOLLSTX

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

Table 5–65 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:** NA **Name:** Collateral Status **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in Settlement status of the Asset Repossession Process case. This process will update the collateral status in the host.

Table 5–66 Update Collateral Status in the Host: Sample Algorithm

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

5.37 Validation Settlement: C1-VALSET

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

Table 5–67 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:** NA
Detailed Design	It is invoked in Settlement status of the Asset Repossession Process case. This process will update the collateral status in the host.

Table 5–68 Validation Settlement: Sample Algorithm

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

5.38 Initiate LMI Process: C1-INITLMI

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

Table 5–69 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 5–70 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

5.39 Close To do's Algorithm: C1-CLSTODO

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

Table 5–71 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:** NA **Name:** To Do Type2 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type3 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type4 **Required (Yes/No):** N **Description:** NA **Name:** To Do Type5 **Required (Yes/No):** N **Description:** NA
Detailed Design	It is invoked while exiting from Settlement status of the Asset Repossession Process case. This process will close all To-Do's associated with the case.

Table 5–72 Close To do's Algorithm: Sample Algorithm

Algorithm Name	C1-CLSTODO
Parameters	**Name:** To Do Type1 **Value:** C1-TD-CL **Name:** To Do Type2 **Value:** C1-TD-AC **Name:** To Do Type3 **Value:** C1-TD-DN **Name:** To Do Type4

	Value: C1-DNA1 **Name:** To Do Type5 **Value:**

5.40 Update Collateral Property: C1-UPCOLPROP

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

Table 5–73 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 Cancelled status of the Asset Repossession Process case. It will update the collateral Properties like IS_LEGAL_SW, IS_REPO_SW depending upon user inputs.

Table 5–74 Update Collateral Property: Sample Algorithm

Algorithm Name	C1-UPCOLPROP
Parameters	**Name:** UpdateCollateralProperty **Value:** RESET

5.41 Update Collateral Status in the Host: C1-UPCOLLSTX

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

Table 5–75 Update Collateral Status in the Host:C1-UPCOLLSTX

>	Update Collateral Status in the host

Description	
Detailed Description	This process will update the collateral status in the host. The value of status to be set will be passed as parameter to the process. 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:** NA **Name:** Collateral Status **Required (Yes/No):** Y **Description:** NA
Detailed Design	It is invoked in Withdrawn status of the Asset Repossession Process case. This process will update the collateral status in the host.

Table 5–76 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:** With the Customer

5.42 PTP Active Algorithm: C1-PTPACTIVE

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

Table 5–77 PTP Active Algorithm: C1-PTPACTIVE

Description	Algorithm to generate letter or SMS on Active Status
Detailed Description	This algorithm is used to generate letter or SMS when PTP moves to Active state.
Algorithm Entity	PTP Active Algorithm

Program Type	java
Program Name	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm
Parameters	**Name:** contactTypeForLetter **Required (Yes/No):** **Description:** Contact Type for Letter generation **Name:** contactClassForLetter **Required (Yes/No):** **Description:** Contact Class for letter generation **Name:** contactMethodForLetter **Required (Yes/No):** **Description:** Contact Method for Letter generation **Name:** contactTypeForSMS **Required (Yes/No):** **Description:** Contact Type for SMS **Name:** contactClassForSMS **Required (Yes/No):** **Description:** Contact Class for SMS **Name:** contactMethodForSMS **Required (Yes/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 5–78 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

5.43 PTP Kept Algorithm: C1-PTPKEPT

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

Table 5–79 PTP Kept Algorithm: C1-PTPKEPT

Description	Algorithm to generate letter or SMS on **PTP** status.
Detailed Description	This algorithm is used to generate letter or SMSon PTP
Algorithm Entity	PTP Kept Algorithm
Program Type	java
Program Name	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPKeptForNgpAlgorithm
Parameters	**Name:** contactTypeForLetter **Required (Yes/No):** **Description:** Contact Type for Letter generation **Name:** contactClassForLetter **Required (Yes/No):** **Description:** Contact Class for letter generation **Name:** contactMethodForLetter **Required (Yes/No):** **Description:** Contact Method for Letter generation **Name:** contactTypeForSMS **Required (Yes/No):** **Description:** Contact Type for SMS **Name:** contactClassForSMS **Required (Yes/No):** **Description:** Contact Class for SMS **Name:** contactMethodForSMS **Required (Yes/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 5–80 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

5.44 PTP Letter SMS Genaration Algorithm: C1_PTPLTRSMS

This section provides details of the PTP Broken Algorithm: C1_PTPLTRSMS algorithm.

Table 5–81 PTP Letter SMS Generation Algorithm:C1_PTPLTRSMS

Description	Algorithm to generate letter or SMS on PTP
Detailed Description	This algorithm is used to generate letter or SMS on PTP
Algorithm Entity	PTP Letter or SMS Generation Algorithm
Program Type	java
Program Name	com.splwg.ccb.domain.customerinfo.paymentPlan.PtpLetterSmsGeneration
Parameters	**Name:** contactType **Required (Yes/No):** Yes **Description:** Contact Type for Letter generation **Name:** contactClass **Required (Yes/No):** Yes **Description:** Contact Class for letter generation

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

Table 5–82 PTP Active Algorithm: Sample Algorithm

Algorithm Name	C1_PTPLTRSMS
Parameters	**Name:** contactType **Value:** OVERDUE **Name:** contactClass **Value:** CCC **Name:** contactMethod **Value:** OTBL

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

- contactTypeForLetter
- contactClassForLetter
- contactMethodForLetter

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

- contactTypeForSMS
- contactClassForSMS
- contactMethodForSMS

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

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

5.45 PTP Broken Algorithm: C1-BRKPTPNGP

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

Table 5–83 PTP Broken Algorithm: C1-BRKPTPNGP

Description	Algorithm to generate letter or SMS on Broken Status
Detailed Description	This algorithm is used to generate letter or SMS when PTP moves to broken state.
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):** Yes **Description:** Contact Class for letter generation **Name:** contactMethodForLetter **Required (Yes/No):** Yes **Description:** Contact Method for letter generation **Name:** contactTypeForSMS **Required (Yes/No):** No **Description:** Contact Class for SMS generation **Name:** contactMethodForSMS **Required (Yes/No):** Yes **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.

5.46 Rule facts populating algorithm: C1-BRLSR

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

Table 5–84 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 5–85 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:**

5.47 Borrower Centric Case Lifecycle

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

Table 5–86 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 Enter Status
Program Type	java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssociateDelinquentAccount
Parameters	
Detailed Design	It is invoked in Pending status of borrower centric case. Transition to Borrower Centricity happens only if a customer has multiple delinquent accounts where he is the main customer only.

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

Table 5–87 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 5–88 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 5–89 Borrower Level : C1-BRWRSW_N

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

5.48 Update Collection Address on Borrower Panel

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

Table 5–90 Person Address Update -Pre-Processing: C1-PADDPRE

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

Detailed Design	This algorithm is hooked in PreprocessBusinessObjectRequestAlgorithmSpot. Business object Name: C1-PERADDRCO. Currently there is no logic inside this algorithm. Implementation team can write their own algorithm in this spot and they can attach this in C1-PERADDRCO

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

Table 5–91 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	
Detailed Design	This is a reference implementation of Post processing algorithm. Customization team can utilize this hook. This is a sample algorithm without having any logic.

5.49 Update Collection Contact Point

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

Table 5–92 Person Contact Point Update - Pre Processing: C1-PCONTPRE

Description	Person Contact Point Update - Pre Processing
Detailed 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.

This table provides details of Collection Contact Point Update - Post Processing: C1-COLLCONTPOST algorithm.

Table 5–93 Collection Contact Point Update - Post Processing: C1-COLLCONTPOST

Description	Person Contact Point 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 Contact Preference - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot
Parameters	
Detailed Design	This is a reference implementation of Post processing algorithm. Customization team can utilize this hook. This is a sample algorithm without having any logic.

5.50 Bankruptcy Process

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

Table 5–94 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Check if any active case is present of a given case category or case type on the customer.
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 Category **Name:** Consider Enterprise Id **Required (Yes/No):** Yes **Description:** Case Category
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 5–95 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Pull all the non delinquent accounts of the customer into collections.
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 5–96 Associate all accounts to the case where customer is a primary borrower- Enter Processing: C1-ASSCTEACC

Description	Pull all the non delinquent accounts of the customer into collections- 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 Pull all Not in Collections accounts into OB Collections (from OBP) whose primary owner is the primary associated customer of the case.
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 5–97 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.
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 5–98 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before.
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 5–99 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Monitor if any of the associated account need to be charged off and monitor delinquency.
Algorithm Entity	Case Type -Auto Transitions
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 Monitring algorithm. Customization team can utilize this hook.

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

Table 5–100 Notify the Bankruptcy Specialist on Hearing Dates- Monitoring: C1-MTR341HRG

Description	Notify the Bankruptcy Specialist on Hearing Dates- Monitoring
Detailed Description	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Notify the Bankruptcy Specialist on Hearing Dates.
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:** Monitor Delinquency
Detailed Design	This is a reference implementation of Monitring algorithm. Customization team can utilize this hook.

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

Table 5–101 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Monitor if the payment plan on any of the associated accounts is Broken for more than x days.
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 Monitring algorithm. Customization team can utilize this hook.

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

Table 5–102 Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD

Description	Monitor if the payment plan on any of the associated accounts is Broken for more than x days- Monitoring
Detailed Description	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Monitor if the payment plan on any of the associated accounts is Broken for more than x days.
Algorithm Entity	Case Type -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorAssetLiquidation
Parameters	**Name:**Validation Date **Required (Yes/No):** Yes **Description:** Validation Date
Detailed Design	This is a reference implementation of Enter Processing algorithm. Customization team can utilize this hook.

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

Table 5–103 Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring: C1-MTRHRNGDT

Description	Notify the Bankruptcy Specialist on RFS Hearing Date- Monitoring
Detailed Descripti	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Notify the Bankruptcy Specialist on RFS

on	Hearing Date.
Algorithm Entity	Case Type -Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorHearingDate
Parameters	**Name:**Validation Date **Required (Yes/No):** Yes **Description:** Validation Date
Detailed Design	This is a reference implementation of 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 5–104 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.
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 5–105 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which Validate if appropriate Case Details have been entered by the user.
Algorithm Entity	Result Type -Post Processing
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
Pameters (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:**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 5–106 Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK

Description	Validate if appropriate Case Details have been entered by the user- 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 Notify Bankruptcy Specialist when a Payment Plan status becomes Kept.
Algorithm Entity	Business Object -Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyPaymentPlanKept
Paramet	NA

ers	
Detailed Design	This is a reference implementation of Business Object Enter Processing algorithm. Customization team can utilize this hook.

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

Table 5–107 Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP

Description	Notify Bankruptcy Specialist of Task Completion - 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 Notify Bankruptcy Specialist of Task Completion.
Algorithm Entity	TO DO Type-Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyTaskCompletion
Parameters	NA
Detailed Design	This is a reference implementation TO DO Type-Post Processing algorithm. Customization team can utilize this hook.

Table 5–108 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)

5.51 Task - Automatic Allocation of tasks to Vendors

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

Table 5–109 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	This is a reference implementation of Pre processing algorithm. Customization team can utilize this hook. This is a sample algorithm which have logic for Automatic Allocation of tasks to Vendors.
Algorithm Entity	TO DO Type - Post Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.vendor.VendorManagementAutomaticTaskAllocation
Parameters	NA
Detailed Design	This is a reference implementation Result Type - Post Processing algorithm. Customization team can utilize this hook.

5.52 Hardship - Associate Accounts of Main Customer

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

Table 5–110 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 have logic for Hardship Entity Association.
Algorithm Entity	Case Type -Enter Processing
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipAssociation
Parameters	NA
Detailed Design	This is a reference implementation Enter Processing algorithm. Customization team can utilize this hook.

5.53 Early Collection

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

Table 5–111 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.
Algorithm Entity	Case Type-Auto Transitions
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 5–112 Park Small Balance Accounts - Monitoring C1-ECPSBA

Description	Park Small Balance Accounts - Monitoring
Detailed	This is a reference implementation of Pre processing algorithm. Customization team can

Description	utilize this hook. This is a sample algorithm which have logic for Park Small Balance Accounts.
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 5–113 Initiate Skip Tracking - No Telephone Number- Enter Processing C1-ECISTNTN

Description	Initiate Skip Tracking - No Telephone Number- 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 have logic for Transition to skip tracing status if no telephone number exists for any of the account holder.
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 5–114 Initiate Skip Tracking - No Telephone Number- Monitoring C1-ECTTSS

Description	Transition to suspended 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 have logic for Transition to suspended status.
Algorithm Entity	Case Type-Auto Transitions
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 5–115 Validate Contact Cap- Monitoring C1-ECVCC

Description	Validate Contact Cap- 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 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 5–116 Schedule Contact - Monitoring C1-ECSC

Description	Schedule Contact - 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 Schedule Contact.
Algorithm Entity	Case Type-Auto Transitions
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 5–117 Initiate Skip Tracing - Wrong Telephone Number- Monitoring C1-ECISTITN

Description	Initiate Skip Tracing - Wrong Telephone Number- 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 Initiate Skip Tracing - Wrong Telephone Number.
Algorithm Entity	Case Type-Auto Transitions
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 5–118 Transition to Under Resolution Status- Monitoring C1-ECTTURS

Description	Initiate Skip Tracing - Wrong Telephone Number- 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 Initiate Skip Tracing - Wrong Telephone Number.
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToUnderResolution Status
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 5–119 Resume Contact From Under Resolution- Monitoring C1-ECRCFUR

Description	Resume Contact From Under Resolution- 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 Resume Contact From Under Resolution.
Algorithm Entity	Case Type-Auto Transitions
Program	Java

Type	
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 5–120 Resume Contact from Small Balance- Monitoring C1-ECRCSB

Description	Resume Contact from Small Balance- 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 Resume Contact from Small Balance.
Algorithm Entity	Case Type-Auto Transitions
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactfromSmallBalance
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 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 5–121 Determine Contact Intensity - Monitoring C1-ECDCI

Description	Determine Contact Intensity - 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 Transitions
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 5–122 Generic Result Post Processing Algorithm for Case Transition and Task Creation- Result Type - Post Processing C1-CTRANTCRET

Description	Generic Result Post Processing Algorithm for Case Transition and Task Creation- 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 Generic Result Post Processing Algorithm for Case Transition and Task Creation.
Algorith	Result Type - Post Processing

m Entity	
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 5–123 Refer to Supervisor- Result Type - Post Processing C1-ECRTS

Description	Refer to Supervisor - 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 Refer to Supervisor.
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 5–124 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
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 5–125 Create case on Follow up- Result Type - Post Processing C1-CRETCSFL

Description	Create case on Follow up - 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 Create case on Follow up.
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 5–126 Hold Case- Result Type - Post Processing C1-HOLDCASE

Description	Hold Case - 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 Hold the Case.
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 5–127 Set Case Data- Result Type - Enter Processing C1-ECUPCASE

Description	Set Case Data- - 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 have logic for set the Case data.
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 5–128 This algorithm will transition the case status to the Suspension status if Cease and Desist = Y C1-CSETRANS

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

Table 5–129 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.
Detailed Design	This is a reference implementation of Pre Processing algorithm. Customization team can utilize this hook.

5.54 Asset Repossession

Table 5–130 Validate Collateral - Enter Validation C1-VALDCOLL

Description	Validate Collateral - Enter Status Validation
Detailed Description	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.
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 5–131 Validate Demand Letter and Acceleration Letter - Enter Validation C1-VALIDDLAL

Description	Validate Collateral - Enter Status Validation

Detailed Description	Validate if Demand Letter and Acceleration letter have been sent
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 CodeRequired Required (Yes/No): No Description: Demand Letter Template 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 5–132 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 Status - Enter Status
Program	Java

Type	
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 5–133 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 Status - 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 5–134 Bankruptcy Check on Associate Customers - Enter Status C1-CHKBKPTCY

Description	Check Bankruptcy- Enter Status
Detailed Description	Verify if any of the customer associated with the case has claimed Bankruptcy
Algorithm Entity	Case Status - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ChkBkpcyOnAssociateCust
Parameters	Name: Repossession Reason for Bankruptcy Required (Yes/No): Yes Description: Repossession Reason for Bankruptcy
Detailed Design	Verify if any of the customer associated with the case has claimed Bankruptcy.

Table 5–135 Monitor if Demand letter and Acceleration letter have been sent on the account.

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 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 5–136 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 Validation
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 5–137 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 5–138 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 5–139 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	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.NotifyOnTa

Name	skCompletion
Parameters	Name: Display Date Required (Yes/No): Yes Description: Display Date
Detailed Design	Create Notification.

Table 5–140 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 Validation
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 5–141 Monitor for Redemption Proceeds C1- REDEPROC

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

Table 5–142 Validate if appropriate Case Details have been entered by the user and transition C1-VALDATAPR

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

	Description: Dynamic Panel One Name: Dynamic Panel One Fields Required (Yes/No): No Description: Dynamic Panel One Fields Name: Dynamic Panel Two Fields 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: caseCharacteristcs
Detailed Design	Validate if the Dynamic Panel Data Elements and Case Characteristics mentioned in the parameters have some values for the case.

Table 5–143 Monitor for Liquidation Setup Complete C1- LIQSETCMP

Description	Monitor for Liquidation Setup Complete

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

Table 5–144 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 Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete
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 5–145 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
Detailed Design	Extract all the Collateral, Account and Customer Information and send it to Alert Module. The contact person details of the Vendor will also be sent to the Alert Module to generate the alert.

Table 5–146 Monitor Redemption Clear Date C1- REDCLRDT

Description	Monitor Redemption Clear Date
Detailed Description	When the redemption clear date is reached transition the case to the Liquidation Setup Status.
Algorithm Entity	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorRedemptionClearDate
Parameters	Name: Next Status

	Required (Yes/No): Yes Description: Next Status Name: Validate Date Required (Yes/No): No Description: Validate Date
Detailed Design	When the redemption clear date is reached transition the case to the Liquidation Setup Status.

Table 5–147 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
Description	Result Post Processing Algorithm for Approvals
Detailed	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.

Description	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 5–148 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 5–149 Result Post Processing Algorithm for Redemption Clear Date C1-RDEEMDATE

Description	Redemption Clear Date Value Date field Calculation
Detailed Description	This algorithm is used to calculate the Redemption Clear Date. By Default Redemption Clear Date will be calculated if REDEM_CLEAR_DT in CI_REPO_CLOSURE table is null and will be calculated as repossession Date + Redemption Clearing Days. Otherwise, Redemption Clear Date will be shown as per the date mentioned in REDEM_CLEAR_DT in CI_REPO_CLOSURE table.

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

5.55 Miscellaneous

Table 5–150 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 5–151 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: Update No Of Days Required (Yes/No): No Description: No Of Days

Table 5–152 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 5–153 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 ConditionRequired Required (Yes/No): No Description: Next Transition Condition

Table 5–154 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

5.56 Derived Field

Table 5–155 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	
Detailed Design	This algorithm will update timezone of a person if it is blank

5.57 Task

Table 5–156 Validate Task Completion (Case Closure) C1- VALTASKCM

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

Table 5–157 Validate Task Completion C1- VALTASKEX

Description	Monitor for Liquidation Setup Complete
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 5–158 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 Validation
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.tasks.algo.AutomaticTaskCreatiomn
Parameters	Name: Task Type Required (Yes/No): Yes Description: Task Type Name: Queue1 Required (Yes/No): Yes Description: Queue 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

5.58 Event Manager

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

Table 5–159 Set Account Warning Indicator C1-ACWRNGIND

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

6 Localized Algorithms

6.1 Localized Algorithms

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

Description	Case Transition for Active Service Member
Detailed Description	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 6–2 Case Transition for Active Service Member C1-ACTMEMCHK

Description	Case Transition for Active Service Member

Table 6–3 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): No Description: Validation Date Name: Repossession Block Period Required (Yes/No): Yes Description: Repossession Block Period
Detailed Design	Verify if repossession needs to be blocked as per SCRA regulations

Table 6–4 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	Case Type - Auto Transition
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete
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 6–5 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): No Description: Account Status Code for Voluntary Surrender
Detailed Design	Metro 2 Reporting - Account Status Code

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

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

Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.Metro2ComplianceCodePostProcessingAlgo
Parameters	Name: Compliance Condition Code Char Required (Yes/No): Yes Description: Compliance Condition Code Char
Detailed Design	Set the Compliance Condition Code sent to Credit Bureau with the value selected in the characteristic given in the parameter.

Table 6–7 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 a Account is close than it should be marked as Close
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2CheckForOpenStatusEnterProcessing
Parameters	NA
Detailed Design	The logic is incorporated for Metro Algorithm only if an Account is close than it should be marked as Close

Table 6–8 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 6–9 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 6–10 Metro 2 Reporting - Credit Grantor Cannot Locate Consumer C1-CGCLC

Description	Metro 2 Reporting - Credit Grantor Cannot Locate Consumer
Detailed Description	Automatically Set in Skip Tracing Status - Enter Processing. Set for all borrowers on the account. Parameter CII CODE - Mandatory
Algorithm Entity	Case Type - Enter Status
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.caseType.earlyCollections.CreditGrantorCannotLocateConsumer
Parameters	Name: Cii Code Required (Yes/No): Yes Description: Cii Code
Detailed Design	Credit Grantor Cannot Locate Consumer

Table 6–11 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	Case Type - Enter Status
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 6–12 Metro 2 Reporting - Set DPD and Outstanding amount to all associated accounts C1-SETDPD

Description	Metro 2 Reporting - Consumer Information Indicator Chapter 13 Post Discharge
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

Table 6–13 DMDC Check is required or not C1-DMDCREQ

Description	DMDC Check is required or not
Detailed Description	This algorithm is used to check whether SCRA verification request should call to DMDC or not based on number of days passed.
Algorithm Entity	Generic Algorithm Spot
Program Type	Java
Program Name	com.splwg.ccb.domain.collection.dmdc.VerifyDMDCDetailsAlgorithm
Parameters	Name: Frequency of DMDC verification (Days)

	Required (Yes/No): Yes Description: Frequency of DMDC verification (Days) Name: Validation Date Required (Yes/No): Yes Description: Validation Date
Detailed Design	This algorithm is used to check whether SCRA verification request should call to DMDC or not based on number of days passed.

7 Feeder Services

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

Table 7–1 Feeder Services

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

Service Name	Method Name	Description	Mandatory Fields
AccountArrearFeederApplicationService	AccountArrearFeederResponse update(SessionContext sessionContext,AccountArrearFeederWrapperDTO accountArrearFeederWrapperDTO) throws FatalException;	This service adds or updates account arrears related fields in the feeder table. It handles add, update and delete operations. In case of delete, the service also deletes the record from main table.	hostAcctNumber, srcHostId, referenceVal
AccountWarningIndFeederApplicationService	AccountWarningIndFeederResponse update(SessionContext sessionContext,AccountWarningIndFeederWrapperDTO accountWarningIndFeederWrapperDTO) throws FatalException;	This service adds or updates account warning indicator related fields in the feeder table. It handles add, update and delete	hostAcctNumber, srcHostId

Service Name	Method Name	Description	Mandatory Fields
		operations.	
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	srcHostId, hostCustomerNbr, fdrAddrSeqId, addrTypeCd

Service Name	Method Name	Description	Mandatory Fields
		feeder table. It handles add, update and 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

Service Name	Method Name	Description	Mandatory Fields
FeedePerIdApplicationService	FeedePerIdResponse update (SessionContext p_SessionContext, FeedePerIdWrapperDTO p_ FeedePerIdWrapperDTO) throws FatalException	This service adds or updates party ID type related fields, such as driving license and so on in the feeder table. It handles add, update and delete operations.	srcHostId, hostCustomerNbr, idType
GroupFeederApplicationService	GroupFeederResponse update (SessionContext sessionContext,GroupFeederWrapperDTO groupFeederWrapperDTO) throws FatalException	This service adds or updates group related fields in the feeder table. It handles add and update operations.	Group_id, determinantValue,srcHostId
GroupMemberFeederApplicationService	GroupMemberFeederResponse update(SessionContext sessionContext, GroupMemberWrapperDTO groupMemberWrapperDTO) throws FatalException	This service adds or updates group member related fields in the feeder	Group_id, srcHostId,determinantValue,Party_id(Host_cust_nbr),party_Name

Service Name	Method Name	Description	Mandatory Fields
		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 updates custo	hostCustomerNbr, determinantValue, svcOrdNum, srcHostId

Service Name	Method Name	Description	Mandatory Fields
		mer 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

8 Dialer Webservice Integration

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

8.1 Generic Data Type

This section provides details of the generic data type.

Table 8–1 Generic Data Type

Data Type	Format
A (CHARACTERS A-Z)	UPPERCASE Characters, LEFT JUSTIFIED, PADDING with BLANK spaces.
AN (ALPHANUMERIC A-Z,0-9)	Alphanumeric fields those are mainly used as RECORD IDENTIFIER must be in UPPERCASE and LEFT JUSTIFIED and PADDING characters are BLANK SPACES.
DT (DATE)	■ 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

8.2 Summary

This section provides the detail summary for dialer webservice.

Table 8–2 Summary

Field Name	Description
Description	Dialer Webservice
Service Status	New
Business Process	Notify collector on outbound call to customer
Owner	OB Collections
Source System(s)	OB Collections
Target System(s)	OB Collections consultant
Service Layer	Data Service
Service Scope	OB Collections
Service Domain	OB Collections

8.3 Interface

This section provides the details on the interface.

Table 8–3 Interface

Field Name	Description
Direction	Outbound
Interaction Pattern	Online
Protocol	< SOAP/HTTP>
Webservice Name	CollectionDialerWebService
Method	notifyUser
Message Exchange Pattern	Synchronous Request Response
Filename Format	NA
Security Pattern	SAML assertion

Field Name	Description
Transaction Pattern	Atomic
Error Pattern	Handled in Synchronous response
Recovery Pattern	None
Business Data Element (s)	OB Collections Case Data
Request Data Object(s)	Customer Information
Response Data Object (s)	Successful
Pre-Conditions	Customer Information exists in OB Collections
Post-Conditions	Customer information is passed to requesting system

8.4 Service Management

This section provides the details on service management.

Table 8–4 Service Management

Field Name	Description
Monitoring	
Alerting	
Availability	24*7
Failover / Failback	
Backup / Restore	
Performance (Latency)	2 Seconds per request
Performance (Peak Volumes)	
Performance (Data Volumes)	
Performance (Concurrency)	Expectation of maximum 20 concurrent requests.
Error Logging	Required
Auditing / Logging	Required

8.5 Request Message Details

As a part of request, consultant will pass user ID of logged in user, account number,case ID, party ID and Transaction Branch, Target Unit, Accessible Target Units, Host String. These fields will also be sent as these

are required by OB Collections to perform Authentication and Authorization checks.

8.6 Header Record

Not Applicable

8.7 Detail Record

This section provides the information on detail record.

Table 8–5 Detail Record

Sr. No	OBP Field Name	Data Type	Length	Mandatory / Optional	Description	DTO Mapping
1	User ID	AN	255	Mandatory	User ID of logged in user	Username
2	Account Number	N	40	Mandatory	Unique identifier of account	SessionContext.transactionBranch
3	Case id	N	10	Optional	Unique identifier of case	SessionContext.targetUnit
4	Customer Number	N	40	Optional	Unique identifier of customer	SessionContext.accessibleTargetUnits
5	Host String	AN	120	Mandatory	Source Host String field provides the information about the host where the concerned account is stored	AccountCustomerProfileWrapperDTO.CollectionDTO.sourceHostString

8.8 Translation Rules

Not Applicable

8.9 Response Message Details

As a response to the request, the proper success or failure success or failure response message will be sent.Service notify collector working on specified account/customer in the input about the outbound call made by vendor.

8.10 Customer Information

This section provides the details on customer information.

Table 8–6 Customer information

Sr. No	OBP Field Name	Data Type	Length	Mandatory / Optional	Description
1	Message	AN	Mandatory	Success or failure message	Success or failure message

8.11 Constraints

Not Applicable