Oracle® Banking Platform

Host Extensibility Guide – Collections Algorithm Spots

Release 2.7.1.0.0

F16199-01

March 2019

ORACLE®

Oracle Banking Platform Host Extensibility Guide – Collections Algorithm Spots, Release 2.7.1.0.0

F16199-01

Contents

Preface

This document provides the detailed list of algorithm spots which can be used for extending and customizing the product.

Audience

This guide is intended for the users of Oracle Banking Platform.

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.

Related Documents

For more information, see the following documentation:

- For installation and configuration information, see the Oracle Banking Platform Localization Installation Guide - Silent Installation guide.
- For a comprehensive overview of security for Oracle Banking, see the Oracle Banking Security Guide.
- For the complete list of Oracle Banking licensed products and the Third Party licenses included with the license, see the Oracle Banking Licensing Guide.
- For information related to setting up a bank or a branch, and other operational and administrative functions, see the Oracle Banking Administrator Guide.
- For information related to customization and extension of Oracle Banking, see the Oracle Banking Extensibility Guides for HOST, SOA, and UI.

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.

Convention	Meaning
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 List of Algorithm Spots

The detailed list of algorithm spots which can be used for extending and customizing the product are listed in the following table.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
SaTypeSaStopAlgorithmSpot	This algorithm spot is used to stop the contarct.	void setServiceAgreement(ServiceAgreement serviceAgreement);	com.splwg.ccb.domain.collection.batch.algorithm.FinalizeCollectionContractStopAlgoComp	com.splwg.ccb.domain.collection.batch.algorithm.FinalizeCollectionContractStopAlgoComp_Impl	Stop Contract: C1-CURENTITY	This algorithm will stop the contract for the account which is to be cured.
CureEntityAlgorithmSpot	This algorithm spot is used to cure the account.	void setAccountId(Account_Id acctId);	com.splwg.ccb.domain.collection.batch.algorithm.CureEntityAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.CureEntityAlgorithm_Impl	Cure Account: C1-FINCOLL	This algorithm performs following activities: - Invoke OBP service to set the incollection flag in host as "N". - Mark incollection flag as "N" in collections. - Set end date in CI_PARTY_COLLECT as posting date. - Update number of times account is self cured(used for statistics). - Remove strategy review date. Parameter:

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						contactMethods : This soft parameter accept the comma separated value of customer contact method which is used to calculate the no. of times account is self cured
AllocationGroupQueueAlgorithmSpot	This algorithm spot is used to allocate the entities.	void setAllocationGroup(String allocationGroup); void setBusinessDate(Date businessDate); void seToQueueBean(AllocationGroupCasesToQueueBean caseAllocToQueue); AllocationGroupCasesToQueueBean getCaseToQueueBean(); AllocationGroupCasesToQueueBean getCaseAllocationMap();	Com.splwg.ccb.domain.collection.batch.algorithm.AllocationGroupQueueAlgoComp	com.splwg.ccb.domain.collection.batch.algorithm.AllocationGroupQueueAlgoComp_Impl	Queue Allocation: C1-ALLOCQUEU	This Algorithm type is used to allocate the entities such as cases to queues. ci_allocation_monitor_vw view is shipped from product to filter cases.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.batch.algorithm.CustomerLevelSwitchUpdateAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.CustomerLevelSwitchUpdateAlgorithm_Impl	Update Customer Switch: C1-CUSTSW	Update customer level case status on case enter processing. Customer Level Switch Name: Please enter the customer level case status switch which needs to update. eg. BANKRUPT_SW, HARDSHIP_SW, IMPRISONED_SW, DECEASED_SW, ABSCONDING_SW etc Switch Value: Please enter the switch value as Y or N

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.batch.algorithm.RepoAndLegalCaseUpdateAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.RepoAndLegalCaseUpdateAlgorithm_Impl	Update Legal/Repo Switch: C1-LEREPOCT	Algorithm Type to update Legal and Repo case status on enter process Legal Repo Switch Name: Please enter the Legal or Repo case switch column name of account extension eg. LEGAL_CASE_EXISTS_SW or REPO_CASE_EXISTS_SW etc Switch Value: Please enter the switch value as Y or N
UserAllocationAlgorithmSpot	This spot being used for allocation of user using various algorithms.	void setUserToQueueBean(UserAllocationToQueueBean userAllocToQueue); UserAllocationToQueueBean getUserToQueueBean(); UserAllocationToQueueBean	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationRoundRobinAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationRoundRobinAlgorithm_Impl	User Allocation - Round Robin: C1-USRALCRR	User Allocation Round Robin algorithm type allocates cases to users on the basis of capacity set during configuration on queue admin. OverFlow cases will get assigned to Exception User.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getUserAllocation Map();				
UserAllocationAlgorithmSpot	This spot being used for allocation of user using various algorithms.	void setUserToQueueBean(UserAllocationToQueueBean userAllocToQueue); UserAllocationToQueueBean getUserToQueueBean(); UserAllocationToQueueBean getUserAllocationMap();	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPercentageBaseAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.UserAllocationPercentageBaseAlgorithm_Impl	User Allocation - % Based: C1-USRALCPR	User Allocation Percentage based algorithm type allocates cases to users on the basis of percentage allocations set during configuration on queue admin. OverFlow cases will get assigned to Exception User.
VendorAllocationAlgorithmSpot	This spot being used for allocation of vendor using various algorithms.	void setVendorToQueueBean(VendorAllocationToQueueBean vendorAllocToQueue); VendorAllocationToQueueBean getVendorToQueueBean(); VendorAllocationToQueueBean getVendorAllocationMap();	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationRoundRobinAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationRoundRobinAlgorithm_Impl	Vendor Allocation - Round Robin: C1-VENALCRR	This algorithm will allocate cases to vendors in round robin fashion. This algorithm is invoked from the User Allocation batch (C1-USALC). OverFlow cases will get assigned to Exception User of the queue.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
VendorAllocationAlgorithmSpot	This spot being used for allocation of vendor using various algorithms.	void setVendorToQueueBean(VendorAllocationToQueueBean vendorAllocToQueue); VendorAllocationToQueueBean getVendorToQueueBean(); VendorAllocationToQueueBean getVendorAllocationMap();	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationPercentageBaseAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.VendorAllocationPercentageBaseAlgorithm_Impl	Vendor Allocation - % Based: C1-VENALCPR	This algorithm will allocate cases to vendors in percentage base. This algorithm is invoked from the User Allocation batch (C1-USALC). OverFlow cases will get assigned to Exception User of the queue.

BulkContactCreationAlgorithmSpot	This algorithm spot is used for creation of contact for accounts in bulk.	void setAccountId(String accountId);
		void setContactClass(String contactClass);
		void setContactTypeCode(String contactTypeCode);
		void setMode(String mode);
		void setCharacteristicType(String characteristicType);
			com.splwg.ccb.domain.collection.batch.algorithm.BulkContactCreationAlgoComp	com.splwg.ccb.domain.collection.batch.algorithm.BulkContactCreationAlgoComp_Impl	Bulk Contact Creation: C1-BLKCNTCRE	This algorithm type is called from Bulk Contact Creation Batch. It invokes business service 'C1-GenMultipleCorrespondence' which creates a customer contact for the accounts filtered by the condition builder attached to the process codes in bulk contact admin.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		void setCharacteristicValue(String characteristicValue);
		void setJointNominationForAcc(String jointNominationForAcc);
		void setContactDateTime(String contactDateTime);
		void setGeneratedBy(String generatedBy);				

CrossStrategyActionMatrixAlgorithmSpot	This algorithm spot is used to execute the business logic when a case enters into a particular state and all the changes related to the state change are committed to DB.	void setCase(ToDoCase toDoCase);
		void setCaseOriginalStatus(CaseStatus caseStatus);
		String getNextCaseStatus();
			com.splwg.ccb.domain.collection.batch.algorithm.CrossStrategyActionMatrixAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.CrossStrategyActionMatrixAlgorithm_Impl	Cross Strategy Action Matrix: C1-CSAM	Cross Strategy Action Matrix Algorithm Type is used by Strategy Monitor and case association process in order to take actions on existing strategies and recommended strategies based on CSAM Matrix.
						
						Parameters :
						Check Status- It checks the status with which the matrix has to be

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						dealt with. Possible values are "Y" or "N"

CustomerClassFtFreezeAlgorithmSpot	The purpose of the algorithm spot is to call an algorithm defined as the FT Freeze algorithm for FT Freeze System Event on a Customer Class.	void setFinancialTransaction(FinancialTransaction financialTransaction);
		void setFinancialTransactionType(FinancialTransactionTypeLookup financialTransactionType);
		void setRegularFinancialTransaction(FinancialTransaction regularFinancialTransaction);
		Bool getFinancialTransactionProcessAdded();
			com.splwg.ccb.domain.collection.batch.algorithm.LastPaymentDtAmtUpdateAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.LastPaymentDtAmtUpdateAlgorithm_Impl	Last Payment for Account: C1-PAYDTAMTU	This algorithm is used to stamp the last payment date and last payment amount for written off accounts.

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckAssociationReview	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckAssociationReview_Impl	Association Review Check: C1-ASORVCHK	This is to decide if the system association of entities should be reviewed by the user or not. Soft Parameters: Next Status: Values Possible

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	entry and links the Case to it as FK Characteristic	String getNextCaseStatus() String getNextTransCondition()				for Next Status{ASSNEWLSP}. This is applicable if Association Review Required is set N. Association Review Required= Possible Values{Y,N} If Association Review is Required {Y}- Stay in current status for user review. Set display date to current business date. If association Review Not Required {N} - Transition to specified next status.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.DefaultNoticeExpiryCheck	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.DefaultNoticeExpiryCheck_Impl	Validate Expired Default Notice: C1-DEFNOEXP	System should check that for associated accounts default notice has expired, This check can be for primary account or for all associated delinquent account based on

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	Case to it as FK Characteristic	getNextCaseStatus() String getNextTransCondition()				parameter. 1. Association Type= {P,A}.P=Primary Type Association,A= Primary as well as Secondary type association 2. To Do Type= To Do will be created if validation failure option is N. 3. To Do Role= To Do Role for the specified To Do Type. 4. Validationfailure Option= {Y,N}. If it is Y then case transition will be failed elsen a To Do will be created.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase_Impl	Associate Related Entity: C1-ASSOENTY	The algorithm checks the associated accounts of the primary account.The association of the primary account is done on the basis of the persons attached to the account and their financially

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	Characteristic	String getNextTransCondition()				responsible status.if the account has the same set of financially reponsible persons attached as in the case for the primary account, the accound is associated. The algorithm parameter are as follows: 1)To Do Role:Specifies the role for the To Do Type created in case of any exception arrising in association of accounts. 2)To Do Type:Specifies the To Do Type created in case of any exception arrising in association of accounts. 3)Host Id:Specifies the host Id.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckLegalCase_Impl	Validate Legal Case Exists: C1-CHKLGL	The Algorithm checks if there is already open legal case for the primary account/Associated accounts linked to the case.The algorithm takes the parameters as follows: 1)To Do Role:Specifies the Role for the To Do Type. 2)To Do Type:Specifies the todo type created when the legal case has been created from batch mode and there is open legal case for the Primary Account/Associated Accounts. 3)Case Category:Specifies the case category for the case(LEGL is for Legal Case)

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.AssignNewLSP	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.AssignNewLSP_Impl	Assign New LSP: C1-ASGNLSP	This algorithm will assign a new LSP to the current case. LSP is a external vendor which is mapped LEGAL service Type. If manual review is not required then case will automatically transition to next status metntioned in softparameter.Below are the softparameter eaxample 1. Next Status: value can be possible next status example{PREPLGLDOC etc.} 2. Prv Allocation Check: Possible values {Y, N}. If this switch is Y system will check if a legal case was created for any of the accounts associated with the current legal case in past.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						3 .Reset Doc Sub Date Sw= Possible values {Y, N}. Value N means document submission date from previous assignment will be copied to new assignment. 4. Change Allocation Option= Possible values {AUTO_WITH_REVIEW,AUTO_WITHOUT_REVIEW, MANUAL}. AUTO_WITH_REVIEW= System allocation with review option. AUTO_WITHOUT_REVIEW=System allocation without review option. MANUAL=Manual allocation. System will not allocate LSP. 5. New Allocation And Review Option= Possible values {AUTO_WITH_RE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						VIEW,AUTO_WITH_REVIEW_PRVALLOC,AUTO_WITHOUT_REVIEW, MANUAL} AUTO_WITH_REVIEW= System allocation with review option. AUTO_WITH_REVIEW_PRVALLOC=System allocation and review will be required if previous allocation was retained. AUTO_WITHOUT_REVIEW=System allocation without review option. MANUAL=Manual allocation. System will not allocate LSP.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CreateApprovalRequest	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CreateApprovalRequest_Impl	Check Approval Requirement: C1-APPRCHK	This algorithm creates approval request if required based on certain conditions.This process will check if LSP assignment needs to be approved if LSP assignment status = "Pending Approval" Approval would be required if either of below is true • System allocation override by user i.e. user has changed the LSP assigned by the system. Set Approval Reason as "Allocation override" • Exposure i.e. sum of balances for all accounts associated with the case is more than a specified threshold. However if no threshold has been specified this parameter should be ignored. Set

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Approval Reason as "High Exposure" • In case approval is required for both the reason, concatenate the approval reasons before sending for approval. If approval is required • Transition the case to a specified status defined as the parameter
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.ResumeStatusLSP	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.ResumeStatusLSP_Impl	Resume Status from Previous LSP: C1-RESSTATUS	Algorithm to resume previous status

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDate_Impl	Check Submission date - CI_CHKSUBDT1	The algorithm checks if the legal documents have been submittedand if that is the case,it changes the status for the caseId. The algorithm takes the following parameters: 1)Next Status: Specifies the next status for case transition if change status is{Y} 2)Change Status{Y,N}: Y defines to change the status,N defines not to change the status.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatu	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssignment	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.UpdateLSPAssignment_Impl	Update LSP (CLOS): C1-LSPSTATUS	Set LSP assignment status to value provided in the parameter. This should be done only for Latest LSP assignment and if it was done by current legal case. If Status = Closed or Cancelled set

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	Characteristic	s() String getNextTransCondition()				Assignment End date = Business Date Status possible values {CLOS,REJ,CAN, PNAP} CLOS=Closed REJ=Rejected PNAP=Pending for Approval.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification_Impl	Collateral Verification: C1-VRFYCOLS	This will perform following validations for the collateral with the case • if soft parameter Collateral type to this algorithm type is "PROPERTY" then, Only one collateral is associated with the case also 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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						is set to not Sold • it will also validate that if There is not active Asset repossession case running for the collateral. If any of the above validations fail case creation process should be terminated
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AccountAssociationForAssetRepossessionCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AccountAssociationForAssetRepossessionCase_Impl	Account Association for Asset Repossession Case: C1-ARSACCTS	This algorithm will perform following actions: • It will get all facilities to which this collateral is associated also it will get all accounts for these facilities. • It will Associate 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.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						This algorithm doesn't have any soft parameter.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CustomerAssociationForAssetRepossessionCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CustomerAssociationForAssetRepossessionCase_Impl	Customer Association for Asset Repossession Case: C1-ARSCUSTS	This algorithm will perform following actions: • It will get all customers who are the owners for the selected collateral • It will Associate 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 doesn't have any soft parameter.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralProperty	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralProperty_Impl	Update Collateral Property: C1-UPCOLPROP	This algorithm will perform foolowing operations: 1)if the value of updateCollateralProperty soft parameter is "SET" and type of

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	sample algorithm creates To Do entry and links the Case to it as FK Characteristic	getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()				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. 2)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

CaseTypeExitStatusAlgorithmSpot		void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CloseTodo_Impl	Close To do's Algorithm: C1-CLSTODO	This process will close all To-Do's of specific To-do types associated with the case. Up to 5 To-DO types can be given to this algorithm to close.

CaseTypeEnterSt	The purpose of the algorithm spot	void setCase(ToDoCas	com.splwg.ccb.domain.collection.ca	com.splwg.ccb.domain.collection.ca	Validations for Mandatory	Subjective Validations for

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
atusAlgorithmSpot	is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	e toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	seType.specialisedCollections.AssetRepo.MandatoryCharacteristics	seType.specialisedCollections.AssetRepo.MandatoryCharacteristics_Impl	Characteristics: C1-CHARVALS	Mandatory Characteristics: This process will validate specified characteristics to be present on the case with reference to value selected by the user for one of the characteristics. This algorithm will have reference characteristic type and up to 5 validation characteristic type as parameters,So based on the reference characteristic type and value of this characteristic, system should validate that mandatory characteristic types have some value captured. If the parameter specifying mandatory characteristic type is blank, it should be ignored
CaseTypeEnterSt	The purpose of	void	com.splwg.ccb.do	com.splwg.ccb.do	Update Collateral	Subjective

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
atusAlgorithmSpot	the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	main.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost	main.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost_Impl	Status in the Host: C1-CHARVALZ	Validations for Mandatory Characteristics: This process will validate specified characteristics to be present on the case with reference to value selected by the user for one of the characteristics. This algorithm will have reference characteristic type and up to 5 validation characteristic type as parameters,So based on the reference characteristic type and value of this characteristic, system should validate that mandatory characteristic types have some value captured. If the parameter specifying mandatory characteristic type is blank, it should be ignored

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost_Impl	Update Collateral Status in the host - C1-UPCOLLSTX	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. Collateral status can be one of the following: 1) REALIZATION_IN_PROGRESS 2)RELIZATION_COMPLETE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost_Impl	Initiate Collateral Valuation: C1-COLLVALX	this alogrithm will work 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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						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.

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateralSettlementStatus	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateralSettlementStatus_Impl	Validation Settlement: C1-VALSET	This algorithm will perform following actions:
						Before completing the asset repossession case below validations should be done for the case
						1. Collateral should have a settlement date
						2. Realization status for the collateral should be "REALIZATION_COMPLETE"
						Possible values of Realization status

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						are : REALIZATION_IN_PROGRESS and REALIZATION_COMPLETE. Transition to completed status should fail if above validations fail.

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateLMIP	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateLMIP_Impl	Initiate LMI Process: C1-INITLMI	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"

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						use NA
PtpActiveForNgpAlgorithmSpot	This algorithm spot is used for performing additional processing like generation of customer contact when PTP is acitve.	void setPromiseToPay(PromiseToPay promiseToPay); PaymentPlanStatusLookup getPaymentPlanStatus();	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPActiveForNgpAlgorithm_Impl	PTP Active Algorithm: C1-PTPACTIVE	This algorithm is used to perform additional processing when the status of a PTP becomes Active. Customer Contacts can be generated via this algorithm.Contact Class, method and type have to be specified. Following parameters used to perform processing--- 1) contactTypeForLetter -- Contact Type for Letter. 2)contactClassForLetter -- Contact Class for Letter. 3)contactMethodForLetter -- Contact Method for Letter. (Value should be --OTBL(Outbound Letter)) 4)contactTypeForSMS--- Contact Type for SMS. 5)contactClassForSMS--- Contact

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Class for SMS. 6)contactMethodForSMS--- Contact Method for SMS. (Value should be--OTBS (Outbound Short Message Service))
PtpKeptForNgpAlgorithmSpot	This algorithm spot is used for performing additional processing when PTP is kept	void setPromiseToPay(PromiseToPay promiseToPay); PaymentPlanStatusLookup getPaymentPlanStatus();	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPKeptForNgpAlgorithm	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPKeptForNgpAlgorithm_Impl	PTP Active Algorithm: C1-PTPKEPT	This algorithm is used to perform additional processing when the status of a PTP becomes Kept. Customer Contacts can be generated via this algorithm. Contact Class, method and type have to be specified. Following parameters used to perform processing--- 1) contactTypeForLetter -- Contact Type for Letter. 2)contactClassForLetter -- Contact Class for Letter. 3)contactMethodForLetter -- Contact

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Method for Letter. (Value should be --OTBL(Outbound Letter)) 4)contactTypeForSMS--- Contact Type for SMS. 5)contactClassForSMS--- Contact Class for SMS. 6)contactMethodForSMS--- Contact Method for SMS. (Value should be--OTBS (Outbound Short Message Service))
PtpBrokenForNgpAlgorithmSpot	This Algorithm spot is used for performing additional processing when PTP is broken.	void setPromiseToPay(PromiseToPay promiseToPay); PaymentPlanStatusLookup getPaymentPlanStatus();	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPBrokenForNgpAlgorithm	com.splwg.ccb.domain.customerinfo.paymentPlan.CollectionPTPBrokenForNgpAlgorithm_Impl	PTP Broken Algorithm: C1-BRKPTPNGP	This algorithm is used to perform additional processing when the status of a PTP is set to Broken. Customer Contacts can be generated via this algorithm. Following parameters used to perform processing--- 1) contactTypeForLetter -- Contact Type for Letter.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						2)contactClassForLetter -- Contact Class for Letter. 3)contactMethodForLetter -- Contact Method for Letter. (Value should be --OTBL(Outbound Letter)) 4)contactTypeForSMS--- Contact Type for SMS. 5)contactClassForSMS--- Contact Class for SMS. 6)contactMethodForSMS--- Contact Method for SMS. (Value should be--OTBS (Outbound Short Message Service))

RuleFactsPopulationAlgorithmSpot	This Algorithm spot is used for populating facts which are required for rule engine.	void setInputKeyValue1(String inputKayValue1);
		void setInputKeyValue2(String inputKayValue2);
		void setInputKeyValue3(String inputKayValue3);
		void setInputKeyValue4(String
			com.splwg.ccb.domain.collection.RuleFactsPopulation	com.splwg.ccb.domain.collection.RuleFactsPopulation_Impl	Rule facts populating algorithm: C1-BRLSR	This algorithm is used to populate the facts required for rule engine.
						Input Key
						Input Key 1 to 5 represent primary key of BO(used in Input BO name 1 - 5)
						Note:
						Currently you can use only Input key 1,2 and 3 and Input BO 1,2 and

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		inputKayValue4);
		void setInputKeyValue5(String inputKayValue5);
		void setRuleId(String ruleId);
		void setRuleSetExecution(Bool isRuleSet);
		void setRuleEffectiveDate(String ruleEffectiveDate);
		void setFactDetails(CollectionsFactDetailsLoader collectionsFactDetailsLoader);
		
		RuleFactParameters getRuleFactParameters();
		String getGivenFactCodes();
						3
						
						Valid values in Input key and Input BO
						Input key 1 (Mandatory) Account Id (No other input value allowed)
						Input key 2 Party UID (No other input value allowed)
						Input key 3 Main Customer PER_ID for given account ID
						(No other input value allowed)
						Input key 4 NA
						Input key 5 NA
						Input BO name 1 (Mandatory) C1-ACCT-EXTN
						Input BO name 2 BO having primary key as input key 2 i.e. Party UID
						Input BO name 3 BO having primary key as input key 3 i.e. Per ID
						Input BO name 4 NA
						Input BO name 5 NA

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssociateDelinquentAccount	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssociateDelinquentAccount_Impl	Borrower Centric Case Lifecycle-C1-ASSODELAC	Associate new delinquent account of the customer to the case.
PreprocessBusinessObjectRequestAlgorithmSpot		void setAction(BusinessObjectActionLookup boAction); void setBusinessObject(BusinessObject bo); void setRequest(BusinessObjectInstance boRequest);	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPreProcess	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPreProcess_Impl	Preson Address Update - Pre Processing - C1-PADDPRE	Person Address PreProcessing algorithm. Attached on BO pre processing spot. This is a hook provided to customization. This can be utilized to validate the address data.
PreprocessBusinessObjectRequestAlgorithmSpot			com.splwg.ccb.domain.collection.address.PersonCollectionAddressPreProcess	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPostProcess_Impl	Update Collection Address on Borrower Panel-C1-PERADDPP	This is a reference implementation of Post processing Algo. Customization team can utilize this hook.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
PreprocessBusinessObjectRequestAlgorithmSpot			com.splwg.ccb.domain.collection.address.ContactPreferencePreProcess	com.splwg.ccb.domain.collection.address.ContactPreferencePreProcess_Impl	Update Collection Contact Point-C1-PCONTPRE	Contact Point PreProcessing algorithm. Attached on BO pre processing spot. This is a hook provided to customization. This can be utilized to validate the contact point data.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.CheckBankruptcyCaseExist	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.CheckBankruptcyCaseExist_Impl	Check if Special Case Already Exist on the Customer- Enter Processing: C1-CKSPLCASE	Check if any active case is present of a given case category or case type on the customer Processing steps are as below 1. If only Case Category is specified check if any active case is running on the customer whose a. Case category is same as the parameter set for the algorithm 2. If Case Type is specified check if any active case is running on the customer whose

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						a. Case type is same as the parameter set for the algorithm 3. If yes validation should fail 4. If Consider Enterprise Id = Y perform the check for all the parties with same Enterprise Id. Consider Enterprise Id value should be "YES" or "NO"
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyPullNonDelinquentAcc	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyPullNonDelinquentAcc_Impl	Pull all the non delinquent accounts of the customer into collections - Enter Processing: C1-PullNDAcc	Processing steps are as below • Pull all Not in Collections accounts into OB Collections (from OBP) where the associated customer is one of the borrower. • If Account Relationships = MC consider only the accounts where the customer is primary owner. If Account Relationships = FO consider all

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						accounts where the customer is a financial owner. If Account Relationship = All consider all accounts where the customer is a financial or non financial owner. • If Consider Enterprise Id = Yes; Determine the Enterprise Id corresponding the party id; then determine the party id corresponding to OBP host and then proceed to pull the accounts. Possible Values of Account Relationships MC, FO, ALL Possible Values fro Consider Enterprise Id Yes/No

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyAssociateAcc	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyAssociateAcc_Impl	Associate all accounts to the case where customer is a primary borrower-Enter Processing: C1-ASSCTEACC	Associate all accounts to the case where customer is a primary borrower For the primary customer associated with the case • Get all accounts where this customer is primary owner and the accounts are In Collections. (Fetch accounts based on Enterprise Id if Consider Enterprise ID = Y) • Shortlist the accounts that are not yet associated with the case. • Associate the shortlisted accounts with the case Consider Enterprise Id value should be "YES" or "NO"
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the	void setCase(ToDoCase toDoCase)	com.splwg.ccb.domain.collection.caseType.specialise	com.splwg.ccb.domain.collection.caseType.specialise	Exclude all the associated accounts from	For all the accounts associated with

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	dCollections.bankruptcy.BankruptcyExcludeAccDlr	dCollections.bankruptcy.BankruptcyExcludeAccDlr_Impl	Dialer- Enter Processing: C1-ExcAccDlr	the case • Call the Dialer Exclusion Service to exclude the accounts from feed to Dialer
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyInitiateCollateralValuation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyInitiateCollateralValuation_Impl	Initiate Collateral Valuation for all collaterals whose last valuation was done 'X' days before- Enter Processing: C1-IniCltVal	For each collateral on the associated account if last valuation was done 'X' days before than create a Collateral Valuation Task. Enter the Collateral Code; Collateral Type and Collateral Description as Remarks Exclude Collaterals with Collateral Types specified in parameter. Also Exclude Collaterals that have been already Repossessed or Sold.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Values of Validation Date: POSTING DATE, SYSTEM DATE
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorChargeOffDelinquency	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorChargeOffDelinquency_Impl	Monitor if any of the associated account need to be charged off and monitor delinquency-Monitoring: C1-MTRCRGDQY	If any of the associated account has delinquency Start Date = Today's posting date Create Bankruptcy Notification as: 'Account <Account Number> has become Delinquent' Set Display Date of the case to current business date. Monitor Charge Off: If any of the associated account has DPD= Charge Off Threshold Create Bankruptcy Notification as 'Account <Account Number> can be Charged Off' Set Display Date of the case to current business date.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						If Secured Accounts = Yes than associated accounts with Secured Switch = Y should also be considered. Monitor Delinquency = "Y" or "N" ,Monitor Charge Off = "Y" or "N" ,Secured Accounts = "Y" or "N" Values of Validation Date: POSTING DATE, SYSTEM DATE
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitor341Hearing	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitor341Hearing_Impl	Notify the Bankruptcy Specialist on Hearing Dates-Monitoring: C1-MTR341HRG	If 341 Hearing Date has been captured and is in future Create a notification for the Bankruptcy Specialist when the 341 Hearing date has been passed. i.e. when Business Date = 341 Hearing Date + 1 Notification: "Capture details of 341 Hearing" Set Display Date

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						of the case to current Business Date If Objection Hearing Date has been captured and is in future Create a notification for the Bankruptcy Specialist when the Objection Hearing date has been passed. i.e. when Business Date = Objection Hearing Date + 1 Notification: "Capture details of Objection Hearing for Debtors Proposed Plan" Set Display Date of the case to current Business Date Values of Validation Date: POSTING DATE, SYSTEM DATE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorPaymentPlan	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorPaymentPlan_Impl	Monitor if the payment plan on any of the associated accounts is Broken for more than x days-Monitoring: C1-MTRPYMPLN	If for any of the associated account on the case the days since the last PTP Broken reaches X days a notification should be created on the case. The PTP Type specified in the parameter should be considered Notification: <PTP Type> broken for account <Account Number>. Days since plan broken <Days Since PTP Broken>. Set Display Date of the case to current business date. Values of Validation Date: POSTING DATE, SYSTEM DATE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorAssetLiquidation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorAssetLiquidation_Impl	Notify the Bankruptcy Specialist if the Liquidation reaches a specific status- Monitoring: C1-MNTRASLQD	Notify the Bankruptcy Specialist if the Liquidation reaches a specific status. If for any of the associated account if the liquidation case reaches a specific status than create a notification for the Bankruptcy Specialist. Notification: "Liquidation for Account <Account Number>; Collateral <Collateral Code> has reached status <Case Status> Set Display Date of the Bankruptcy Case to Business Date

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorHearingDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyMonitorHearingDate_Impl	Notify the Bankruptcy Specialist on RFS Hearing Date-Monitoring: C1-MTRHRNGDT	If for any of the associated account on the case if the RFS Hearing Date is reached Create Notification: "Capture details for RFS Hearing for Account <Account Number> When Business date = Hearing Date + 1 Set Display Date of the case to current Business Date Values of Validation Date: POSTING DATE, SYSTEM DATE

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		void setActionSourceId(String actionSourceId);
		void setActionSourceSt
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.DetermineBankruptcyTreatment	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.DetermineBankruptcyTreatment_Impl	Determine in which status the case should proceed for Bankruptcy Treatment- Post Processing C1-DTMBKTRTM	Bankruptcy Chapter Field should be passed as a Filing Information Chapter(FC) or Converted to Chapter(CC) as an input

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		atusCode(String actionSourceStatusCd);
		void setActionId(String actionId);
		void setActionType(ActionType actionType);
		void setResultType(ResultType resultType);
		boolean getIsProcessingComplete();
						parameter
						
						If Bankruptcy Chapter = Chapter 7
						Then Transition to Manage Chapter 7 Bankruptcy Status
						
						If Bankruptcy Chapter = Chapter 13
						Then Transition to Manage Chapter 13 Bankruptcy Status
						
						If Bankruptcy Chapter = Chapter other than 7 or 13
						Then Transition to Other Bankruptcy Status
						
						Bankruptcy Chapter Field = "FC" or "CC
						Where "FC" = Filing Chapter and "CC"=Convert to chapter

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of	void setActionEntity(String actionEntity); void setActionSourceId	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.ValidateBan	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.ValidateBankruptcyCaseData_	Validate if appropriate Case Details have been entered by the user- Post Processing C1-	Validate if the Dynamic Panel Fields mentioned for the corresponding Dynamic panels

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

	result.	(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			kruptcyCaseData	Impl	VLDBCDATA	have some values for the case.
						
						If yes the Follow Up is saved successfully.
						If no system should throw an error message for the first blank field that it will encounter.
						
						Error Message: "<Field Name> cannot be blank"
						
						Possible values for Panel Names and Panel fields belonging to that Panel are as follows:
						
						Panel Name : bankruptcyTrusteeInfoPanel
						Corresponding Panel Fields: ENTITY_NAME,PHONE,EMAIL,FAX_NUMBER,CONTACT_POINT_NAME,CONTACT_POINT_PHONE_NUM,CONTACT_POINT_EMAIL,CO

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						NTACT_POINT_FAX Panel Name : bankruptcyProcessingInfoPanel Corresponding Panel Fields : HEARING_DATE,HEARING_LOCATION,LENDER_COLL_VAL_DATE,LENDER_COLL_VAL,DISCHARGE_DATE,DISMISSED_DATE,CHAPTER_CODE,COVERSION_REMARKS,CONVERSION_DATE,HEARING_ADD_INFO Panel Name : bankruptcyDebtorAttorneyPanel Corresponding Panel Fields : FIRM_NAME,PHONE,ENTITY_NAME,DEBTOR_ADDRESS Panel Name : bankruptcyFilingInfoPanel Corresponding Panel Fields : DATE_OF_BNKPT_CASE_FILE,BNKPT_CASE_NUM

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						,COURT,CHAPTER Panel Name : bankruptcyConfirmPlanInformation Panel Corresponding Panel Fields : RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT Panel Name : bankruptcyDebtorProposedPlanInfo Panel Corresponding Panel Fields : RECEIVE_DT,TOTAL_AMMOUNT,LAST_PAYMENT_DT,OBJECTION_DATE,OBJECTION_OUTCOME,HEARING_DATE Panel Name : bankruptcyLegalCounselInfoPanel Corresponding Panel Fields : ASSIGNED_DATE,COUNSEL_NAME,CONTACT_POINT_NAME,EMAIL,PHONE,ALTERNATE_PHONE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
BusinessObjectEnterStatusAlgorithmSpot			com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyPaymentPlanKept	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyPaymentPlanKept_Impl	Notify Bankruptcy Specialist when a Payment Plan status becomes Kept- Post Processing C1-NTPYMPLNK	Create Notification Notification: <PTP Type> Kept for account <Account Number>. Set Display Date of the case to current business date.
ToDoTypeToDoPostProcessAlgorithmSpot	This Algorithm spot is used for notifying task completion and also for allocating task to vendor.		com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyTaskCompletion	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BankruptcyNotifyTaskCompletion_Impl	Notify Bankruptcy Specialist of Task Completion- Post Processing C1-NTFTSKCMP	Create Notification Notification: <Task Id> - <Task Name> complete for <Account Number>. Set Display Date of the case to current business date. Notification should be created on the latest case associated on the Account
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatu	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.JointBnkptcyAssociateCust	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.JointBnkptcyAssociateCust_Impl	Joint Bankruptcy - Associate other customers to the Bankruptcy case - C1-ASSCUSTJB	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		s() String getNextTransCondition()				OB Collections from OBP through UI or through pull non delinquent accounts)
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BkptcyPayPlanCreation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BkptcyPayPlanCreation_Impl	Create Pay Plan for a Bankruptcy Case - Enter Processing: C1-CRTATP	This algorithm will create a dummy pay plan for all accounts associated with a bankruptcy case. The pay plan is created with pending status in the following tables : 1. CI_BKPTCY_PAY_PLAN_INFO 2. CI_BKPTCY_PAY_PLAN_DTLS

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BkptcyPayPlanClosure	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.BkptcyPayPlanClosure_Impl	Close Pay Plan for a Bankruptcy Case - Exit Processing: C1-CLDATP	This algorithm will close the pay plan for all accounts associated with a bankruptcy case. The pay plan is marked with close status in the following tables :
						
						1. CI_BKPTCY_PAY_PLAN_INFO

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						2. CI_BKPTCY_PAY_PLAN_DTLS 3.CI_BKPTCY_PAY_PLAN_SCHED
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BanckruptcyMonitorArrerageAlanNotification	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.BanckruptcyMonitorArrerageAlanNotification_Impl	Notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days - Monitoring: C1-MTRARPLNT	Algorithm to notify the Bankruptcy Specialist for Arrearage Overdue Amount and Overdue Days of an account, if these values are above the threshold values provided as parameters. The required parameters are : 1. Arrearage Plan Threshold Days 2. Arrearage Plan Threshold Amount 3. Confirmed Plan Threshold Days 4. Confirmed Plan Threshold Amount 5. Notification Date Type Notification is generated as -> <Arrearage/Confirmed Plan> amount for Account Number <account no> of

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						<currency symbol> <overdue amount> is overdue by <overdue no of days> Days
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.algorithm.RiskIndicatorSetResetEnterProcessing	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.arrearage.algorithm.RiskIndicatorSetResetEnterProcessing_Impl	Set or Reset Account level Warning Indicator for Bankruptcy - Enter Processing: C1-SETWI	This Algorithm Set or Reset the Account level Warning Indicators of all the associated accounts of Bankruptcy. Note this will exclude the Charge-Off Accounts.(Based on RECOVERY_SW in CI_ACCT_EXTN table) Risk Indicator Codes should be comma separated. Values: Risk Indicator = SET or RESET Risk Indicator Code = <Risk Indicator Code1,Risk Indicator

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Code2,...>
ToDoTypeToDoPostProcessAlgorithmSpot	This Algorithm spot is used for notifying task completion and also for allocating task to vendor.		com.splwg.ccb.domain.collection.vendor.VendorManagementAutomaticTaskAllocation	com.splwg.ccb.domain.collection.vendor.VendorManagementAutomaticTaskAllocation_Impl	Vendor Management - Automatic Allocation of tasks to Vendors - TO DO Type – Post Processing C1-TSKVNDR	On creation of task check if task is already allocated to a member. If Yes no action required. If No allocate the case to the member with lowest number of tasks of that task type in the queue.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipAssociation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipAssociation_Impl	Hardship - Associate Accounts of Main Customer - Enter Processing C1-HARASOPND	This algorithm associates the Party on whom the hardship case is created.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateContact	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateContact_Impl	Transition to contact state if First Contact Date has reached and set the Re Allocation Switch Case Type Auto Transition Algo C1-ECIC	Transition to contact state if First Contact Date has reached If First Contact Date has reached (based on the parameters below) Or Account is Direct Debit and Immediate Transition if Direct Debit = Yes/No Transition to Contact RM status if Relationship Manager exists and Contact RM status has been specified Transition to Contact Alternate status if Contact Alternate Flag = Y and Contact Alternate Status has been specified Else Transition to Contact Status

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Set Re-Allocation Switch = Y for the case post case transition Possible Values First Contact Calculation Parameter: DPD, DIA, Days Since Case Start Immediate Transition if Direct Debit: Y,N Validation Date : POSTINGDATE, SYSTEMDATE
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.ParkSmallBalanceAccounts	com.splwg.ccb.domain.collection.caseType.earlyCollections.ParkSmallBalanceAccounts_Impl	Park accounts with small balances to a separate status Case Type - Auto Transition C1-ECPSBA	Park accounts with small balances to a separate status so that no contacts are initiated for the account If Net Arrear Amount <= Small Balance Threshold And Net Arrear Amount > 0 Then transition to

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						small balance status. Net Arrear Amount = (Overdue Amount - Unclear Amount) If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the calculations. Set Re-Allocation Switch = Y for the case post case transition. Possible Values : Use Overdue Amount : Y,N
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatu	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracing	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracing_Impl	Transition to skip tacing status if no telephone number exists for any of the account holder Case Type - Enter Status Algo C1-ECISTNTN	If no contact points exists then move the case to Skip Tracing status Check if one of the Contact Points as specified in the parameters exists for any of the account holder.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		s() String getNextTransCondition()				If no contact point exists than move the case to Skip Tracing Status. Set Re-Allocation Switch = Y for the case post case transition.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToSuspendedStatus	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToSuspendedStatus_Impl	Transition to suspended status if the account has one of the warning indicator set Case Type - Auto Transition C1-ECTTSS	If the Account has one of the Account Risk Indicators specified in the parameter Transition to Suspended status. Create a task if Task Type has been mentioned and assign it to the Specified Queue Set Re-Allocation Switch = Y for the case post case transition. Set Suspend Reason = Risk Indicator Exit. If either of the financial owners have one of the Party Indicators mentioned in the parameter than transitionto

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Suspended status. Create a task if Task Type has been mentioned and assign it to the Specified Queue Set Re-Allocation Switch = Y for the case post case transition. Set Suspend Reason = Risk Indicator Exit. If there is at least one financial owner with no Risk indicators mentioned in the parameter 'Party Risk Indicators – Contact Alternate' than transition the case to the Contact Alternate Status. If case already in Contact Alternate Status don't initiate the transition but perform the other activities. Create a task if Task Type has been mentioned

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						and assign it to the Specified Queue Set Re-Allocation Switch = Y for the case post case transition. Set Alternate Contact Flag = Y Set Alternate Contact Reason = Risk Indicator Exit.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.ValidateContactCap	com.splwg.ccb.domain.collection.caseType.earlyCollections.ValidateContactCap_Impl	The algorithm will hold the case when the contact cap is reached Case Type - Auto Transition C1-ECVCC	Check if the contact cap has reached for the case If case is not already on Hold and Display Date <= Business Date And the number of successful contacts linked to the case in last X number of days >= Contact Cap Hold the case for Y number of days with the given Hold Reason.. Logic for considering

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						successful contacts: All contacts with given contact methods that have Authentication Status = Green Possible Values for Validation Date {SYSTEMDATE,POSTINGDATE}
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.ScheduleContact	com.splwg.ccb.domain.collection.caseType.earlyCollections.ScheduleContact_Impl	This algorithm will Schedule Contact for the case as per the given intensity Case Type - Auto Transition C1-ECSC	Schedule Contact for the case as per intensity • If case is not on Hold • And Display Date <= Business Date or Display Date is Blank • Set Display Date = Max((Last Successful Contact Date + Contact Intensity), Business Date) Consider Contact Intensity from Algorithm parameter if specified else picks up Contact

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Intensity from case level field. Logic for considering successful contacts: Last contact with given contact methods that have Authentication Status = Green Validation Date can be POSTINGDATE or SYSTEMDATE
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracingInvalidTelNumber	com.splwg.ccb.domain.collection.caseType.earlyCollections.InitiateSkipTracingInvalidTelNumber_Impl	Transition to skip tacing status if 'X' number of consecutive calss fails Case Type - Auto Transition C1-ECISTITN	"Transition to skip review if 'X' number of consecutive failed contacts • If last X number of consecutive contacts has been unsuccessful, transition to Skip Tracing Status. Logic for considering unsuccessful contacts: If last X consecutive contacts with given contact

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						methods have Authentication Status other than 'Green' Set Re-Allocation Switch = Y for the case post case transition Possible Values for Validation Date are POSTINGDATE and SYSTEMDATE"
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToUnderResolutionStatus	com.splwg.ccb.domain.collection.caseType.earlyCollections.TransitionToUnderResolutionStatus_Impl	Transition to under resolution status if Net Arrear Amount <=0 Case Type - Auto Transition C1-ECTTURS	Transition to under resolution status if Net Arrear Amount <=0 • Transition the case to Under Resolution Status if Net Arrear Amount <= 0 or PTP is running on the account. • Set Re-Allocation Switch = Y for the case post case transition Net Arrear Amount =

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						(Overdue Amount - Unclear Amount) If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the calculations. Possible values: Use Overdue Amount: Y,N
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactFromUnderResolution	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactFromUnderResolution_Impl	Resume Contact From Under Resolution Status Move case to contact status if the Net Arrear Amount is greater than zero Case Type - Auto Transition C1-ECRCFUR	Resume Contact From Under Resolution Status • If there is no more active PTP on the account and • If the Net Arrear Amount > 0 Than transition the case to Contact RM Status if RM exists and Contact RM status has been configured

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Contact Alternate Status If Contact Alternate Flag = Y Else Contact Status Set Re-Allocation Switch = Y for the case post case transition If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the calculations. Net Arrear Amount = (Overdue Amount - Unclear Amount) Use Overdue Amount can be Y/N or Yes/No

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactfromSmallBalance	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeContactfromSmallBalance_Impl	This algorithm is used to resume contact from small balance status. Case Type - Auto Transition C1-ECRCSB	This algorithm is used to resume contact from small balance status. If Net Arrear Amount > Small Balance Threshold Then transition the case to Contact RM Status if RM exists and Contact RM status has been configured Contact Alternate Status If Contact Alternate Flag = Y Else Contact Status Set Re-Allocation Switch = Y for the case post case transition If Use Overdue Amount = Yes than use Overdue Amount instead of Net Arrear Amount in the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						calculations. Net Arrear Amount = (Overdue Amount - Unclear Amount) Possible Value: Overdue Amount : Y,N
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.DetermineContactIntensity	com.splwg.ccb.domain.collection.caseType.earlyCollections.DetermineContactIntensity_Impl	Determine Contact Intensity and Contact Intensity Review Date Case Type Auto Transition Algo C1-ECDCI	"Determine Contact Intensity and Contact Intensity Review Date • If case is not on Hold • And Business Date >= Contact Intensity Review Date or Contact Intensity Review Date is Blank • Call Rule Specified in the parameter • Set Contact Intensity and Contact Intensity Review Date Validation Date Can be POSTINGDATE or

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						SYSTEMDATE"

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseTransitionandTraskCreationPostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseTransitionandTraskCreationPostProcessingAlgo_Impl	Generic Result Post Processing Algorithm for Case Transition and Task Creation Result Post processing Algorithm C1-CTRANTCRE	Generic Result Post Processing Algorithm for Case Transition and Task Creation
						
						Transition the case to given Case Status if Case Status is configured and the current status is present in one of the Valid Current Statuses. Display an error 'The selected result <Result Type> is not allowed in current Status.' If the current status is not present in one of the valid status.
						• Create Task of given Task Type and assign it to the give Task Queue if Task Type is configured.
						• Map the created task with the Follow up Id of the Follow Up that created the task.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						• Set Re-Allocation Switch = Y if Re-Allocate = Y • Copy the common characteristics of result into the case. (here the char codes need to be maintained at both the result type and case type level) • Task Creation Logic: If Task For = 'Account' Create Task on the primary associated account on the case If Task For = Customer Create Task on the primary associated Customer of the case If Task For = Case Create Task on the case If Task For = Admin Create Admin level Task

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Note: Task For is the mandatory characteristic at Task Level. Task For = Customer is an invalid configuration for Account level Case and vice versa. Possible Values of Re-Allocate Switch and Copy Characteristics to Case are : Y/N "Event Name" and "Action Flag" fields are introduced to update Cease_Desist\Contact_Alternate\Dispute Flags, where:- "Event Name" will be provided depending on the FLAG which you need to update. So, it can have one of the values:- Event Name :- "CEASE_DESIST" Event Name :-

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						"CONTACT_ALT" Event Name :- "DISPUTE" And "Action Flag" value will be SET\RESET. To set Cease_Desist\Contact_Alternate\Dispute Flags to "Y", provide Action Flag :- "SET". To set Cease_Desist\Contact_Alternate\Dispute Flags "N", provide Action Flag :- "RESET".

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void
			com.splwg.ccb.domain.collection.caseType.earlyCollections.SupervisorReferralPostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.SupervisorReferralPostProcessingAlgo_Impl	This algorithm will transfer the case to the given status if the current staus of the case is present in Valid Current status list Result Type Post Processing Algo C1-ECRTS	Supervisor Referral Algorithm
						
						• If case is present in one of the status's specified in 'Valid Current Status' than
						
						Proceed with further actions
						
						Else
						
						Display an error 'The selected

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
						result <Result Type> is not allowed in current Status.'
						
						And don't proceed with further actions.
						
						• Transition the case to given Case Status
						
						• Create Task of given Task Type and assign it to the Supervisor Queue (Queue of Task) of the Case Queue
						
						• Map the created task with the Follow up Id of the Follow Up that created the task.
						
						• Set Re-Allocation Switch = Y if Re-Allocate = Y
						
						Re-Allocate can be Y/N

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeCollectionsPostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.ResumeCollectionsPostProcessingAlgo_Impl	Resume Collections
					
					Transits the case to
					
					Contact status
					
					Result Type Post Processing Algo C1-RESCOLL
						Resume Collections
						
						Transition the case to
						
						Contact RM Status if RM exists and Contact RM status has been configured
						
						Contact Alternate Status If Contact Alternate Flag = Y
						
						Else Contact Status
						
						Set Re-Allocation Switch = Yes if Re-Allocate = Y
						
						Re-Allocate can be Y/N

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseCreationonFollowupPostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.CaseCreationonFollowupPostProcessingAlgo_Impl	Create Required Case on Follow Up
					Result Post processing Algorithm C1-CRETCSFL
						Create Required Case on Follow Up
						
						If Account Level Case Type creates case on account, If Customer level Case Type creates case on the main customer of the account. Queue to which the case should be allocated if provided else the case should remain unallocated with Re-Allocation Switch as Y

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.earlyCollections.HoldCasePostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.HoldCasePostProcessingAlgo_Impl	Hold Case for Days as provided in Characteristic Type provided in Hold Period or if that is blank Hold Period should be referred from Hold Period parameter.
					And
					Hold Reason should be set as provided in characteristic type provided in Hold Reason or if that is blank Hold Reason should be referred from Hold Reason parameter.
					
					Result Type Post Processing Algo C1-HOLDCASE
						Hold Case for Days as provided in Characteristic Type provided in Hold Period or if that is blank Hold Period should be referred from Hold Period parameter.
						And
						Hold Reason should be set as provided in characteristic type provided in Hold Reason or if that is blank Hold Reason should be referred from Hold Reason parameter.
						Validation Date Can be POSTINGDATE or SYSTEMDATE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateCaseData	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateCaseData_Impl	Update Case Level Data when a case enters a new status - C1-ECUPCASE	Update Case Level Data when a case enters a new status Set Case Characteristics to specific values provided in algorithm parameters. On entering the value the corresponding characteristic validation algorithm should be triggered. If type is mentioned but value is not than the char type needs to be made blank.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus()	com.splwg.ccb.domain.collection.caseType.earlyCollections.CeaseDesistAccountSuspension	com.splwg.ccb.domain.collection.caseType.earlyCollections.CeaseDesistAccountSuspension_Impl	This algorithm will transition the case status to the Suspension status if Cease and Desist = Y - C1-CSETRANS	Additional algorithm in Pending Status: Enter Processing to transition to Suspend Status if Cease and Desist = Y.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		String getNextTransCondition()				

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.algorithms.ScheduleCallPostProcessingAlgorithm	com.splwg.ccb.domain.collection.algorithms.ScheduleCallPostProcessingAlgorithm_Impl	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.
					
					Code - C1-SCHCALL
						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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collectionevt.ResetCaseWarningIndOnHost	com.splwg.ccb.domain.collectionevt.ResetCaseWarningIndOnHost_Impl	Reset WI in the host C1-RESETWI	This algorithm resets WI in the host. Call the Host Account Warning Indicator Service to set the WI mentioned in the parameter
CaseTypeEnterStatusValidationAlgorithmSpot	The purpose of the algorithm spot is to execute the validation logic before Case is moved into specific status.	void setCase(ToDoCase toDoCase); void setOriginalCaseStatus(CaseStatus caseOriginalStatus);	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateral	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateCollateral_Impl	The input collateral is associated with the account on which the repossession case is being created. The collateral belongs to the collateral type and collateral category specified in the parameters. If collateral type and collateral category are not mentioned no validation will be done. The collateral status is not 'Sold'.	The input collateral is associated with the account on which the repossession case is being created. The collateral belongs to the collateral type and collateral category specified in the parameters. If collateral type and collateral category are not mentioned no validation will be done. The collateral status is not 'Sold'.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
					Date of Sale is blank. There is no repossession case active on the collateral (IS_REPO_SW = N) Code - C1-VALDCOLL	Date of Sale is blank. There is no repossession case active on the collateral (IS_REPO_SW = N)
CaseTypeEnterStatusValidationAlgorithmSpot	The purpose of the algorithm spot is to execute the validation logic before Case is moved into specific status.	void setCase(ToDoCase toDoCase); void setOriginalCaseStatus(CaseStatus caseOriginalStatus);	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ValidateDemandLetterandAccelerationLetter	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ValidateDemandLetterandAccelerationLetter_Impl	If DL Template Code has been mentioned validate if Demand Letter has been sent in last X days. If AL Template Code has been mentioned validate if Acceleration Letter has been sent in last X days. If X Days is not specified just check if the letters have been sent on the account. Checks will be done for all associated	If DL Template Code has been mentioned validate if Demand Letter has been sent in last X days. If AL Template Code has been mentioned validate if Acceleration Letter has been sent in last X days. If X Days is not specified just check if the letters have been sent on the account. Checks will be done for all associated

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
					accounts unless 'Only Primary Account = Yes' in which case the check will be only on primary associated account. Code - C1-VALIDDLAL	accounts unless 'Only Primary Account = Yes' in which case the check will be only on primary associated account.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AssociateCustAssRepo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AssociateCustAssRepo_Impl	Associate all financial owners on the associated accounts to the Repossession case. Code - C1-ASSOCUST	Associate all financial owners on the associated accounts to the Repossession case.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ChkBkpcyOnAssociateCust	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ChkBkpcyOnAssociateCust_Impl	If Repossession Reason <> Bankruptcy For each customer associated with the case Check if the Bankruptcy_Switch = Y. If yes Case Creation will be rolled back and below error message will be displayed. "One or more of the collateral owners have claimed Bankruptcy. Repossession process should be initiated from Bankruptcy process" Code - C1-CHKBKPTCY	If Repossession Reason <> Bankruptcy For each customer associated with the case Check if the Bankruptcy_Switch = Y. If yes Case Creation will be rolled back and below error message will be displayed. "One or more of the collateral owners have claimed Bankruptcy. Repossession process should be initiated from Bankruptcy process"

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.MonitorDemandLetterandAccelerationLetterExpiry	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.MonitorDemandLetterandAccelerationLetterExpiry_Impl	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 Code - C1-MNTRDLAL	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutoApprovalCheckforRepossession	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutoApprovalCheckforRepossession_Impl	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	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
					Balance Number of accounts associated with the collateral Code - C1-REPOAPRV	Balance Number of accounts associated with the collateral

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepossessionTransition	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepossessionTransition_Impl	If Repossession Reason = "Voluntary Repossession" transition to 'Repossession In Progress - Voluntary Surrender'
					else transition to 'Repossession in Progress"
					
					Code - C1-RSTUPCMPL
						If Repossession Reason = "Voluntary Repossession" transition to 'Repossession In Progress - Voluntary Surrender'
						else transition to 'Repossession in Progress"

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		boolean getIsProcessingComplete();				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AutoTaskCreationForVendor	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.AutoTaskCreationForVendor_Impl	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. Code - C1-AUTOTASKC	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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ToDoTypeToDoPostProcessAlgorithmSpot		void setOldToDoEntryDTO(ToDoEntry_DTO oldDTO);
		
		void setNewToDoEntry(ToDoEntry newToDoEntry);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.NotifyOnTaskCompletion	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.NotifyOnTaskCompletion_Impl	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.
					
					Code - C1-NOTRSTSK
						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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutomaticSendingofRedemptionLetters	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.AutomaticSendingofRedemptionLetters_Impl	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. Code - C1-REDEMPLTR	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.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForRedemptionProc	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForRedemptionProc_Impl	When the outstanding amount of all the associated accounts becomes zero move the case to Closed Status. Code - C1-REDEPROC	When the outstanding amount of all the associated accounts becomes zero move the case to Closed Status.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateRepoCaseData	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.ValidateRepoCaseData_Impl	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"
					
					Code - C1-VALDATAPR
						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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorForLiquidationSetUpComplete_Impl	When Repo Title Received Date and Vehicle at Sale Location Date is available the case is moved to the next status. Code - C1-LIQSETCMP	When Repo Title Received Date and Vehicle at Sale Location Date is available the case is moved to the next status.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionAssignmentAlert	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionAssignmentAlert_Impl	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 Code - C1-REPOASAL	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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
LetterTemplateLetterExtractCollectionAlgorithmSpot	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.	void setCustomerContact(CustomerContact customerContact); LetterTemplateInfoBean getLetterTemplateInfo(); ReportDefinition getReportDefinition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ExtractRepossessionAssignmentAlgorithm	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.ExtractRepossessionAssignmentAlgorithm_Impl	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. Code - C1-REPEMTEMP	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.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorRedemptionClearDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MonitorRedemptionClearDate_Impl	When the redemption clear date is reached transition the case to the Liquidation Setup Status. Code - C1-REDCLRDT	When the redemption clear date is reached transition the case to the Liquidation Setup Status.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionApprovalResultPostProcessingAlgorithm	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionApprovalResultPostProcessingAlgorithm_Impl	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
					
					Code - C1-RAPRVRSLT
						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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

AdhocCharacteristicValueValidationAlgorithmSpot	This algorithm spot is invoked on characteristic adhoc values in order to:
	1) validate that the value is correct
	2) possibly perform a reformat of the value prior to storing on the table
		void setFormatOnly(boolean formatOnly);
		void setCharacteristicType(CharacteristicType type);
		void setAdhocValue(String value);
		String getReformattedValue();
		boolean isValidAdhoc();
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDate_Impl	Result Characteristic Value Date field Validation - PASTDATE_VAL	This algorithm is used to validate format enter by user for result characteristics during taking follow up.
						1. 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.
						2. The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD,

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDateCal	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.RepossessionClosureRedemptionClearDateCal_Impl	Redemption Clear Date Value Date field Calculation - C1-RDEEMDATE	This algorithm is used to calculate the Redemption Clear Date.
						By Default Redemption Clear Date will be caculated 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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateReviewDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateReviewDate_Impl	Update Review Date for associated accounts - C1-UPDRVWDT	For all accounts associated with the case this process will update the review date. Below parameters should be available for the process • Update Type o Set Review Date - This will set the Review Date for the account o 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 o Yes - System will update existing account review date i.e. in case a review date is already present, system will override the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						same o 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
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.CaseAutoTransition	com.splwg.ccb.domain.collection.caseType.CaseAutoTransition_Impl	Case Monitoring - CS-MONITOR	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Days. 4. Transition to another Status. The Number parameters cannot be changed. This Algorithm type is hard wired with the product and implementation can only override the 'Program Name' using Feature Config. It is expected that is one time set up per implementation.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateWarningIndicator	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateWarningIndicator_Impl	Update warning indicator for the customer - C1-UPDWARN	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						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)
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.TransitionDefaultNextStatus	com.splwg.ccb.domain.collection.caseType.TransitionDefaultNextStatus_Impl	Transition to Default Next Status - C1-TRAN-STAT	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 -

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Mention the transition condition for the next status.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collectionevt.SetCaseWarningIndOnHost	com.splwg.ccb.domain.collectionevt.SetCaseWarningIndOnHost_Impl	Set Account Warning Indicator	Set Account Warning Indicator
TimeZoneDerivationAlgorithmSpot			com.splwg.ccb.domain.collection.batch.algorithm.TimeZoneDerivationAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.TimeZoneDerivationAlgorithm_Impl	Timezone derivation field update algorithm - C1-TZDRFLD	This algorithm will update timezone of a person if it is blank
CaseTypeEnterStatusValidationAlgorithmSpot	The purpose of the algorithm spot is to execute the validation logic before Case is moved into specific status.	void setCase(ToDoCase toDoCase); void setOriginalCaseStatus(CaseStatus caseOriginalStatus);	com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletionClosure	com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletionClosure_Impl	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	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

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
					open tasks on the accounts associated with the case. Case Type – Enter Validation C1-VALTASKCM	open tasks on the accounts associated with the case.
CaseTypeExitStatusValidationAlgorithmSpot	The purpose of the algorithm spot is to execute business logic when a Case transitions out of a specific status.	void setCase(ToDoCase toDoCase); void setPreviousCaseStatus(CaseStatus caseStatus);	com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletion	com.splwg.ccb.domain.collection.tasks.algo.ValidateTaskCompletion_Impl	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. Case Type – Exit Validation C1-VALTASKEX	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 Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase); void setCaseOriginalStatus(CaseStatus caseStatus); Bool getShouldAutoTransition(); String getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.tasks.algo.AutomaticTaskCreatiomn	com.splwg.ccb.domain.collection.tasks.algo.AutomaticTaskCreatiomn_Impl	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. Case Type – Enter Status - C1-CREATTASK	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.
GenericEventHostUpdateAlgorithmSpot	This is generic algorithm which will be invoke for generic event outcome from event handler	void setPerson(Person person); void setToDoCase(ToDoCase toDoCase); void setAccount(Account account);	com.splwg.ccb.domain.collectionevt.SetWarningIndOnHost	com.splwg.ccb.domain.collectionevt.SetWarningIndOnHost_Impl	Set Account Warning Indicator	Set Account Warning Indicator
VendorServiceTypeAllocationAlgorithmSpot	This algorithm type is used to perform Legal vendor Allocation.	void setVendorServiceType(VendorServiceType vendorServiceType); void setCase(ToDoCas	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.LspVendorAllocationAlgorithm	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.LspVendorAllocationAlgorithm_Impl	Legal vendor Allocation - C1-LGLVNDRAL	Legal vendor Allocation

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		e toDoCase);				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ExtendExpiryDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ExtendExpiryDate_Impl	This algorithm will invoke the host service to extend the hardship expiry date - C1-EXT-EXPDT	This algorithm will invoke the host service to extend the hardship expiry date.Possible Values for Extended Expiry Date Char Type : C1-EXPDT and Exception Transition Condition : EXP
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureEnterStausUpdateDateTime	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureEnterStausUpdateDateTime_Impl	This algorithm will store Case Status Update Date/Time for current status into the element specified by xpath in algorithm soft parameter - C1-CASE-STAT	This algorithm will store Case Status Update Date/Time for current status into the element specified by xpath in algorithm soft parameter.Possible Values for Exception Transition Condition : EXP and Xpath to Date Element : /applicationForm/main/reliefEffectiveDt

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.CreateToDo	com.splwg.ccb.domain.collection.caseType.CreateToDo_Impl	This common algorithm creates a To Do using the values from algorithm parameters - C1-TO-DO	This common algorithm creates a To Do using the values from algorithm parameters. Possible Values for To Do Type,Message Category,Message Number,Characteristic Type For Log Entry and Exception Transition Condition
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckCustomerEligiblity	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckCustomerEligiblity_Impl	Check customer eligibility - C1-CHKCUST	This process will check warning indicators for a customer. This check will be done by a call to rule engine for each customer. Processing logic will be as below Primary entity for the case is account. Based on ownership type parameter for the process, system should consider the customers for

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						eligibility check • If ownership type parameter is set to "financial owner" o Get all financially responsible customers for the account o For each customer, system should call the rule engine to check for customer eligibility • If ownership type parameter is set to "primary" o System should call the rule engine to check for primary customers eligibility Customers' facts should be used for rule engine decision. For each call • Rule will return output as "Validation Status". Possible values can be

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						"Success" OR "Failure" • If validation status = Failed, process should return result as validation failed. o Check Validation failure option § Validation failure option = Fail case creation/transition. Case should not get created or should not transition status. § Validation failure option = Transition status. Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters). • If validation status = Success, process should return result as validation

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						successful. Parameters - • Ownership Type - Ownership type can be FINANCIAL_OWNER(Financial Owner) or PRIMARY(Primary Owner). • Rule ID - Defined rule id to check customer eligibility. Rule should return output validation status in fact 'SuccessOrFailure', which can have value true or false. • Validation Failure Option - This option is use to determine action to be taken in case of validation failure. Permissible values are FAIL_CASE_CREATION(fail case creation) and

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						TRANSITION_STATUS(transition status). • Validation Failure Transition Status - Case transition status in case of validation failure. • Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status. • Cancel Reason Char Value - Characteristic value for the defined characteristic type.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureHardshipApprovalDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CaptureHardshipApprovalDate_Impl	Capture Hardship Approval Date - C1-HARAP-DT	This algorithm will store Case Status Update Date/Time for current status into the element specified by xpath in algorithm soft parameter.Possible Values for Xpathto Date Element and Exception Transition

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		String getNextTransCondition()				Condition

CaseTypeEnterStatusValidationAlgorithmSpot	The purpose of the algorithm spot is to execute the validation logic before Case is moved into specific status.	void setCase(ToDoCase toDoCase);
		
		void setOriginalCaseStatus(CaseStatus caseOriginalStatus);
			com.splwg.ccb.domain.collection.caseCreation.AdapterTest	com.splwg.ccb.domain.collection.caseCreation.AdapterTest_Impl	Algorith that will interface with Rule Engine - C1-RULEADAPT	Algorith that will interface with Rule Engine.

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CollateralVerification_Impl	This algorithm type will perform below validations for the collateral with the case	This algorithm type will perform below validations for the collateral with the case

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckTargetAccountEligiblity	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CheckTargetAccountEligiblity_Impl	Check target account eligibility - C1-CHKTRGT	o System should call the rule engine for eligibility check. Output of rule engine will be "Validation Status" o If validation status is "Success" , § Set set-off status as "Pending" § Compute maximum amount allowed to Debit = Target account Balance - Minimum residual amount § Clear the values in the "Exclude Target Till Date" and "Exclude Reason" fields, if populated o If validation status is "Fail", § Set set-off status for target account as "Not eligible" § "Exclude Reason" should be set as "Not Eligible" § Get offset days

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						for exclude reason from the look-up § Set "Exclude target till" date for the target account to current business days + offset day. § If no offset days are returned, "Exclude target till" date should not be updated • Once all target accounts have been processed and for this case, if none of the target accounts has set-off status as "Pending". o Case should be created and transitioned to the status specified in parameters. o Set given char value for the given char type (as defined in parameters)

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RosoApprovalCheck	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RosoApprovalCheck_Impl	Approval check for set-off transaction - C1-ROSOAPPR	This process will check if approval is required for a set-off transaction. Approval will be required if • Asset classification = Value set as parameter for the process • Accrual status = Value set as parameter for the process • Sum of Debit Amounts for all target accounts >= Specified threshold Based on whether approval is required or not, transition the case to a status as set in the parameters of the process.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.FetchTargetAccounts	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.FetchTargetAccounts_Impl	Get target accounts - C1-GETTRGT	This algorithm gets all savings accounts and term deposit accounts having the same set of owners as owners of the delinquent account.Processing logic for this will be as below • Get all savings accounts and term deposit accounts having the same set of owners as owners of the delinquent account (i.e. primary account associated with the case). Ownership types can however be different. • "Same owners" - indicates that all owners of delinquent accounts are one the savings account / term deposit and there is no additional owner. • If no such

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						accounts are found, Case should be created and transitioned to the status specified in parameters. Set given char value for the given char type (as defined in parameters). Parameters - • Validation Failure Transition Status - Case transition status in case of validation failure. • Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status. • Cancel Reason Char Value - Characteristic value for the defined characteristic type.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						• Casa Account Type Identifier List - Comma separated savings account(CASA) identifiers. • Td Account Type Identifier List - Comma separated term Deposit account(TD) identifiers. • Casa Account Exclude Status List - Comma separated savings account(CASA) status to be excluded while fetching account data from host. • Td Account Exclude Status List - Comma separated term Deposit account(TD) status to be excluded while fetching account data from host. • Exclude

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Blocked Td Account - Flag to exclude blocked Term Deposit account (Y or N). • Exclude Blocked Deposit - Flag to exclude blocked Deposit (Y or N).
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateHardshipStatusToExpire	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateHardshipStatusToExpire_Impl	Update status of relief to Expired in Hardship - C1-UPDHDSTAT	Update status of relief to Expired in Hardship details table. Possible values for Hardship Expire Status.
CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase); void setNextCaseStatus(CaseStatus caseStatus);	com.splwg.ccb.domain.collection.caseType.GenericToDoCompletionForCase	com.splwg.ccb.domain.collection.caseType.GenericToDoCompletionForCase_Impl	To Do Completion for case - C1-TO-DO-END	This common algorithm will complete all To Do's with Drill Keys = Current Case Id and To Do's To Do Type is not excluded from auto

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						completion
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateMarketingConsentFlag	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateMarketingConsentFlag_Impl	Update Marketing Consent flag - C1-MKT-FLG	This is a generic algorithm that will make a service call to Host to update the Marketing Consent flag.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CheckDefaultNoticeForVoluntaryPossession	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CheckDefaultNoticeForVoluntaryPossession_Impl	Check Default Notice for Voluntary possession - C1-CHKDFLT	Check Default Notice for Voluntary possession. Possible value for Check Expiry Status.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDateExitProcessing	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckSubmissionDateExitProcessing_Impl	Check Submission Date: C1-CHKSUBDT2	Check Submission Date

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateFinancialHardshipFlag	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateFinancialHardshipFlag_Impl	Update Financial Hardship flag - C1-FNHRD-FLG	This algorithm will make a service call to Host to update the Financial Hardships flag for Primary Customer and corresponding joint account holders

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String
			com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypeCaseTransitionAlgo	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypeCaseTransitionAlgo_Impl	Result Type Case Transition Algorithm - C1-RTCT	If specified on the Result Type, this algorithm will be invoked when the corresponding result is recorded for a Case (Action/Result UI).
						
						This can be used to transition the case from the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
						current status to the next possible status as follows,
						
						- This algorithm has a parameter Output Status i.e. next possible status, so for case transition, it will be checked whether Output Status is one of the next possible status. If YES, it will transition the case to that status.
						
						- This algorithm has a parameter Input Status, which will be checked against the current status of the Case. This is an optional parameter. If specified, Case transition will happen only when the current status of the case matches with this parameter.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckActiveArsCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.CheckActiveArsCase_Impl	Algorithm to see if case is running before closing - C1-CHKCASE	The algorithm sees if the case is running in the child case category before closing the case from the parent case category

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.CheckDeceasedStatusForCustomer	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.CheckDeceasedStatusForCustomer_Impl	Check Deceased status for the customer - C1-CHKDCD	For the customer for whom the deceased case is being initiated check if Deceased warning indicator is already set OR An active deceased case is present If either of above is true, case creation should fail.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.AssociatedAccWithDeceasedCustomer	com.splwg.ccb.domain.collection.caseType.specialisedCollections.deceased.AssociatedAccWithDeceasedCustomer_Impl	Associated accounts with deceased customer case - C1-DCDACCTS	For the primary customer associated with the case Get all accounts where this customer is primary owner and the accounts are in collections Associated those

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getNextCaseStatus() String getNextTransCondition()				accounts with the case
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ExecuteFundTransfer	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ExecuteFundTransfer_Impl	Execute Fund Transfer - C1-FUNDTRFR	This process will execute the fund transfer. This should follow below steps for "each" target account where debit amount specified is > 0 and set-off status = "Pending" • Execute a payment transfer transaction from Target account to the delinquent account. • If transaction is successful, set set-off status = "Success" for this target account • If transaction is not successful, set set-off status = "Fail" for this target account For any target account where set-off status was "Pending", but

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						were not considered for set-off (because debit amount was specified as zero) update set-off status to "Cancelled" Once all target accounts have been processed, check if at least one payment transfer has status as "Success". • If yes, transition the case to status as set in the parameter "Execution Success Status" • If no, transition the case to status as set in the parameter "Execution Failure Status". Set the char value for the char type as specified in the parameters. Parameters - Execution

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Success Status - Case trasition status if fund tranfer is successful. Execution Failure Status - Case trasition status if fund tranfer fails. Cancel Reason Char Type - Characteristic type to set as case characteristic if fund transfer fails. Cancel Reason Char Value - Characteristic value for the defined characteristic type. Successful Fund Transfer Transaction Status - Trasaction status code to identify successful fund transfer.This values is returned from host service.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.SavePrevioustStatus	com.splwg.ccb.domain.collection.caseType.specialisedCollections.legal.SavePrevioustStatus_Impl	Algorithm to save previous state's status code - C1-SAVPRESTA	Algorithm to save the case status in CI_LSP_DTLS table from where it has come to the current status. This algorithm is must when we are using C1-RESSTATUS. C1-RESSTATUS transition the case to the status which is saved by this (C1-SAVPRESTA) algorithm.

PreprocessBusinessObjectRequestAlgorithmSpot		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObject(BusinessObject bo);
		void setRequest(BusinessObjectInstance boRequest);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignCaseTypeFromFeatureConfig	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignCaseTypeFromFeatureConfig_Impl	Attach case type from feature config attach to BO - C1-ATCHCS	Attach case type from feature config attach to BO.Possible value for Hardship Case Type Feature Config : C1_HSCATY_FC

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateCollectionPartyWarningIndicator	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdateCollectionPartyWarningIndicator_Impl	Update Collection Warning Indicator - C1-UPD-WRIND	This is a generic algorithm that will make a service call to the Host to update Party level warning indicators for the Main Customer. It has following parameters: 1. Warning Indicator Type. 2. Warning Indicator Value 3. Rule Type Code 4. Collection Column To Be Updated 5. Set In Collections On Related Accounts 6. Exception Transition Condition
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipEntityAssociation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipEntityAssociation_Impl	Hardship Entity Association to nominated accounts and financial owners of account - C1-HARDASSO	This algorithm associates all the accounts nominated for hardship and also associates related financial Owners of the accounts selected.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getNextCaseStatus() String getNextTransCondition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignApplicableReliefTypes	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AssignApplicableReliefTypes_Impl	Assign Applicable Relief Type - C1-RELIF-TYP	This algorithm will invoke Rules Engine to determine Applicable Relief Type(s) for each nominated Account.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.interaction.CreateCustomerContactAlgo	com.splwg.ccb.domain.collection.interaction.CreateCustomerContactAlgo_Impl	Create Customer Contact for Resultype Algo - C1-CREATCC	Create Customer Contact for Resultype Algo.Possible values for Customer Contact Class,Customer Contact Type and Preferred Contact Method

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.CalculateCaseStatusExpiryDate	com.splwg.ccb.domain.collection.caseType.CalculateCaseStatusExpiryDate_Impl	Calculate an expiry date when entering case status - C1-CSEXPDT	This algorithm type accepts a parameter for a characteristic type which will be used to create a Case Characteristic which contains a date that is equal to case status change plus Number of Days parameter value
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.CustomerContact	com.splwg.ccb.domain.collection.caseType.CustomerContact_Impl	Create Customer Contact - C1-CUST-CONT	This common algorithm creates a customer contact for the given customer contact type.Possible values are Customer Class,Customer Contact Type,Char Type Cust Cont Log Entry,X Path Completion Flag,Transition Condition and Contact Method

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.TransitionToNextDaysOnBeforeExpiry	com.splwg.ccb.domain.collection.caseType.TransitionToNextDaysOnBeforeExpiry_Impl	Transition to Next Status x days before expiry - C1-NXT-BX-DY	Transition to Next Status x days before expiry.Possible values are Days Before Expiry,Xpath to Expiry Date,Next Status and Next Transition Condition
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipExpiryDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipExpiryDate_Impl	Validate Hardship Expiry Date - C1-VAL-FHEXP	This validates the Hardship Expiry Date.It validates if the expiry date is greater than Posting date and the Allowed Minimum maturity date.

LeavePlanAlgorithmSpot		void setEmailId(String emailId);
		void setSubject(String subject);
		void setTemplate(Strin
			com.splwg.ccb.domain.collection.leavePlan.LeavePlanEmailNotification	com.splwg.ccb.domain.collection.leavePlan.LeavePlanEmailNotification_Impl	Leave Plan Email Notification - C1-USRMAILNT	This will be used to notify with email to User and it's supervisor for updation of Leave application details

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		g template); void setLeaveId(String leaveId);				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateAccountInCollectionFlag	com.splwg.ccb.domain.collection.caseType.specialisedCollections.UpdateAccountInCollectionFlag_Impl	Update Account in collections flag - C1-ACTINCOL	Get all accounts for the customer from the host. Relationship type to be considered will be primary or financial ownership based on parameter set for the process. For the accounts retrieved, check if the account is setup in collections i.e. an active contract is present for the account • If no, set in-collections flag to "N" for the account • If yes. No updates should be done
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CaseAssociationForAssetRepossessionCase	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.CaseAssociationForAssetRepossessionCase_I	Associate related entities with the case - C1-ARSENTITY	Associate related entities with the case.Possible values are Customer Association and Account

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()		mpl		Association
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RevalidateTargetAccount	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.RevalidateTargetAccount_Impl	Revalidate target account - C1-REVALTRGT	This algoritm validates target account (savings and term deposit) balance and computes maximum amount to be debited. Processing logic should be as below • Validate that "Total Debit Amount" is greater than zero. Else transition into the status should fail and appropriate error message be displayed OR recorded in case log (if not executed manually). • It is possible that target account balance got updated after user

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						had entered the debit amounts. System should refresh balance from host. • Re-compute maximum amount which can be debited for each target account For each of the target account with set-off status as "Pending", • If maximum amount which can be debited is < Debit amount specified by the user then o Set set-off status and exclude reason as "Not eligible". o Skip rest of the processing and move to next target account • Call Rule engine to validate the account, which will output "Success" or "Failure". o If for any of the account

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						validation status = "Failure", § Set set-off status and exclude reason as "Not eligible". Once all target accounts have been processed get sum of debit amounts for all target accounts with set-of status as "Pending". Three scenarios are possible • There are no target accounts in pending status. Go to cancel set-off step • Sum of Debit amounts of target account > Overdue amount for delinquent account. In this case check the "Excess debit" option o Adjust Debit Amounts - Proportionately reduce debit amounts from all target accounts.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						See example at bottom of section. o Cancel Set-off - Go to Cancel set-off step • Sum of Debit amounts of target account <= Overdue amount for delinquent account. In this case there is no exception and set-off process should proceed. Cancel Set-off • Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters) Example of proportionate adjustment: Say A1 is delinquent account and has $

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						120 as arrear. Say debit amounts of $ 60 and $40 have been set from target accounts TA1 and TA2. So total amount to be debited is $ 100 Now during revalidation it is found that overdue has dropped to $ 60. So now below computations should be done • X = (Overdue amount) / (Sum of debit amounts) • New Amount to debited from TA1 = Previous debit amount for TA1 * X • New Amount to debited from TA2 = Previous debit amount for TA2 * X So in this case • X = $ 60 / ($60 + $ 40) = 0.6 • New Amount to debited from TA1 = $ 60 * 0.6 = $ 36

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						• New Amount to debited from TA2 = $ 40 * 0.6 = $ 24 Parameters - • Cancel Reason Char Type - Characteristic type to set as case characteristic if validation failure option is transition status. • Cancel Reason Char Value - Characteristic value for the defined characteristic type. • Validation Failure Transition Status - Case transition status in case of validation failure. • Excess Debit Option - Can have value ADJUST_DEBIT_AMOUNTS(Adjust Debit Amounts) or CANCEL_SETOFF(Cancel Set-off).

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						• Minimum Residual Amount - Minimum amount that must be present in account after set-off. • Rule ID - Defined rule id to validate account. Rule should return output validation status in fact 'SuccessOrFailure' , which can have value true or false. • Casa Account Type Identifier List - Comma separated savings account(CASA) identifiers. • Td Account Type Identifier List - Comma separated term Deposit account(TD) identifiers.
CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status	void setCase(ToDoCase toDoCase); void setNextCaseStatus(CaseStatus	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.Initiate_LMIProcess	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.Initiate_LMIProcess_Impl	Initiate LMI - C1-INITLMIS	Initiate LMI. Possible values are No L M I Option,Lmi Insurer Code,Initiate L M I Options,Lmi Case Type and Balance

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	to the next status.	caseStatus);				Threshold
CollectionClassOverdueMonitorRuleAlgorithmSpot		void setAccount(Account account); Bool getIsProcessingComplete();	com.splwg.ccb.domain.collection.batch.algorithm.CollectionCaseCreationOverdueMonitorRuleAlgo	com.splwg.ccb.domain.collection.batch.algorithm.CollectionCaseCreationOverdueMonitorRuleAlgo_Impl	NGP Collection case creation algorithm - C1-COLLCASE	This is overdue monitor Rule algorithm used for NGP Collection Case creation. It will be invoked through the over due monitor batch process C1-ADMOV.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	CaseEnterStatusContractStopAlgoComp	CaseEnterStatusContractStopAlgoComp_Impl	Stop Contract Algorithm - C1-CONTSTOP	This algorithm will stop the contract linked to case in the CI_CASE_PARTY table.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckExistingHardship	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckExistingHardship_Impl	Check for existing Hardship - C1-CHKHRDSHP	Before creating case in Pending state,This Algorithm checks,if there is any active case of Hardship case type (By

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()				Retrieveing case type code from feature configuration). If yes It Display message ' Party is already in Hardship' If no,It will proceed with case creation. This checks for an existing Hardship application for the party.

CollectionActionAlgorithmSpot		void setPersionID(String personId);
		void setContactType(String conType);
		void setContactID(String contID);
			com.splwg.ccb.domain.collection.contacthistory.ContactProcessing	com.splwg.ccb.domain.collection.contacthistory.ContactProcessing_Impl	Algorithm for contact processing - C1-CNTCT	Algorithm for contact processing.

ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckApplicationExpiryDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckApplicationExpiryDate_Impl	check application expiry date - C1-CHKEXP	check application expiry date with allowed minimum date of nominated account and posting date

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);
		void setNewBusinessObject(BusinessObjectInstance boRequest);				

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.correspondence.CustomerContactCreation	com.splwg.ccb.domain.collection.correspondence.CustomerContactCreation_Impl	New Customer Contact Creation Algorithm - C1-CCCREATE	This AlgorithmType is used to create Customer Contact on the basis of Customer Contact class,Customer Contact Type and Preferred Contact Method on a Customer Level case or an Account Level case

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase); void setNextCaseStatus(CaseStatus caseStatus);	com.splwg.ccb.domain.collection.caseType.RemoveCaseCharacteristic	com.splwg.ccb.domain.collection.caseType.RemoveCaseCharacteristic_Impl	Removes a case characteristic on case status exit - C1-REMCSCH	This algorithm type removes a case characteristic with char type = parameter 10 value.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.TransitionToNextStatusOnDate	com.splwg.ccb.domain.collection.caseType.TransitionToNextStatusOnDate_Impl	Transition case on a date on a case characteristic - C1-TRANSDT	This algorithm type transitions the case on a date stored on a case characteristic (char type = parameter 10 value). If the case characteristic is not found, the case will be transitioned on the current date. This algorithm type accepts parameters Next Status or Next Transition Condition to determine the next status

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.SetAccountNextCreditReviewDateToCurrentDate	com.splwg.ccb.domain.collection.caseType.SetAccountNextCreditReviewDateToCurrentDate_Impl	Set Account Nxt Credit Review Date to current date - C1-NXTRVWDT	This algorithm sets the accounts next credit review date to current date
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.MarkAccountsForStrategyReview	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.MarkAccountsForStrategyReview_Impl	Mark accounts for strategy review - C1-REVIW-ACT	This algorithm will mark all accounts that are "in-collections" for the customer in hardship for review.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.WaitTimeOut	com.splwg.ccb.domain.collection.caseType.WaitTimeOut_Impl	Wait Time Out (in days) - C1-WAIT-DAYS	This algorithm times out when the Case has been on the state for too long.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipApplicationInputs	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ValidateHardshipApplicationInputs_Impl	Validate Hardship Application inputs - C1-V-FH-APP	This algorithm validates that all the mandatory fields on the Hardship Application Form are populated.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckForOperationalReliefTypes	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CheckForOperationalReliefTypes_Impl	Check for Operational Relief Types - C1-OP-RT	This algorithm checks if any of the identified stp relief types need to be operational

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AutoApprovalCheck	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.AutoApprovalCheck_Impl	Auto-Approval Check - C1-FH-AUTOAP	This algorithm invokes an Application service which in turn invokes host service which determines if the Hardship application can be auto-approved
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatu	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ApplyHardshipReliefTypes	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.ApplyHardshipReliefTypes_Impl	Apply Hardship Relief Types for accounts in Host - C1-FH-EVAL	This algorithm applies hardship relief types for the accounts in the host.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		s() String getNextTransCondition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdatePartyWarningIndicator	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.UpdatePartyWarningIndicator_Impl	Update Party Warning Indicator - C1-UPD-PRTWI	This is a generic algorithm that will make a service call to Host to update Party level warning indicators for Main Customer If a Rule Type Code is populated, it will first invoke the rule to determine if the Warning Indicator should be updated.
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.TransitionToNextStatusWhenAllReliefsApplied	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.TransitionToNextStatusWhenAllReliefsApplied_Impl	Transition to Next status when all reliefs are app - C1-RAPP	This is algorithm that will transition the case to the next status when all reliefs have been applied
InstallationEntityActivityPopulationAlgorithmSpot		void setEntityType(EntitytypeFlagLookup	com.splwg.ccb.domain.collection.caseCreation.Popula	com.splwg.ccb.domain.collection.caseCreation.Popula	Collection - Entity Activity Population - C1-	This sample algorithm is called from various

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		entityType);
		void setEntityId(String entityId);
		void setModeOfOperation(String modeOfOperation);
		void setActivityEntitytId(String ActivityEntitytId);
		void setActivityEntitytType(String ActivityEntitytType);
		void setActivityEntitytStatus(String ActivityEntitytStatus);
		void setActivityType(ActivityTypeCdLookup activityType);
		void setActivityDateTime(DateTime activityDateTime);
		void setMessageFields(String messageFields);
			teAccountActivityAlgo	teAccountActivityAlgo_Impl	ENTACTPOP	entities classes for population of Account Activity. The algorithm takes following input parameters:
						1)EntityType : Person/Account for which activity is getting created (e.g. Case can be created on Person as well as Account)
						2) EntityId : Person/Account Id
						3) ModeOfOperation: Add/Update/Delete/Cancel
						4) HostEntitytId: Activity Entity Id (e.g PTP/CC/Follow-up/Case Id)
						5)HostEntitytName: PTP/CC/FOLLOWUP/CASE

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelHardshipApplication	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelHardshipApplication_Impl	Cancel Hardship Application - C1-CXLFH	This algorithm will make a service call to host to cancel an active Hardship Application.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.rightofSetOff.PerformPaymentXferForROSO	com.splwg.ccb.domain.collection.caseType.rightofSetOff.PerformPaymentXferForROSO_Impl	Perform Payment Transfer for ROSO - C1-ROSOPMTXR	This Algorithm Type will call a web service which calls Oracle NGP Core Banking to perform a payment transfer between an eligible delinquent Account and eligible Target Account(s).

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.rightofSetOff.ValidateRosoInputs	com.splwg.ccb.domain.collection.caseType.rightofSetOff.ValidateRosoInputs_Impl	Validate ROSO Target Account inputs - C1-RS-VALIN	This Algorithm Type will validate the user inputs entered into the Target Account dynamic panel to ensure they comply with the business rules. If the inputs are not valid, the Case will transition back to the previous status and prompt the user to re-enter the inputs.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.AdhocCollectionEntityCreation	com.splwg.ccb.domain.collection.caseType.AdhocCollectionEntityCreation_Impl	Create RMB Entities from Host Data - C1-VCREATE	Create RMB Entities such as Person, Account ,Account Person, PartyCollect etc from Host Data. Input parameters : 0) Source Host Id : Host Identifier Value e.g. NGP -- Removed in R2.2- Host Id will come from UI b) Inapplicable Statuses : Comma separated Host System Statuses for Account (host_sys_acct_st

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						at_flg) c) Exclude Accrual Status Flag: Comma separated Accrual Statuses for Account (accrl_stat_flg) d) Exclude Asset Class Code: Comma separated Asset Class Codes for Account (asst_class_cd) e) Exclude User Defined Acct Status: Comma separated User Defined Account Status (usr_def_acct_stat_flg) f) Exclude Offer Id: Comma separated Offer Id (offer_id)

PostProcessBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
		void
			com.splwg.ccb.domain.collection.PopulateAccountactivityForNote	com.splwg.ccb.domain.collection.PopulateAccountactivityForNote_Impl	Populate Activity Table For Notes Creation - C1-NTACTVITY	Populate Activity Table For Notes Creation

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);				

PreprocessBusinessObjectRequestAlgorithmSpot		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObject(BusinessObject bo);
		void setRequest(BusinessObjectInstance boRequest);
			com.splwg.ccb.domain.collection.suspendActivity.SuspendActivityPreProcessing	com.splwg.ccb.domain.collection.suspendActivity.SuspendActivityPreProcessing_Impl_Impl	Suspend Activity for Account Pre Processing - C1-SPATACPRE	Suspend Activity for Account Pre Processing

TreatmentActivityMonitorAlgorithmSpot		void setCaseId(ToDoCase_Id caseId);	com.splwg.ccb.domain.collection.batch.algorithm.TreatmentActivityMonitorAlgoComp	com.splwg.ccb.domain.collection.batch.algorithm.TreatmentActivityMonitorAlgoComp_Impl	Sample TAM Algorithm Type - C1-TAMALG	This algorithm will update account and TAM review date for case.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase); void setNextCaseStatus(CaseStatus caseStatus);	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CancelApprovalReqAlgo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CancelApprovalReqAlgo_Impl	Cancel Approval Request - C1-CANAPPR	This algorithm will cancel all pending approval requests for the case. Example for parametervalues for legal Process: Composite Name:-com.ofss.fc.workflow.process.LegalProcessForApproval Instance Title:-LEGAL_CASE_ Value of the above parmeters are depends upon the SOA approval work flow
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.SetDisplayDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.SetDisplayDate_Impl	Set Display Date - C1-SETDSPDT	This process will update the display date for the account. New display date will be computed as = Current display date + offset days If a display date is already present on the account, it should be updated only if new display date is < existing display date

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.DefaultNextStatusAutoTransition	com.splwg.ccb.domain.collection.caseType.specialisedCollections.DefaultNextStatusAutoTransition_Impl	Transition to Default next status after N Days - C1-TRNDFLT	Transition the case to default next status after specific days. Days will be set as parameter for the process. Case should transition to Default next status if, difference in current date and date of entry into current status is >= specified number of days
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.AccountExclusion	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.AccountExclusion_Impl	Check current cases on account for exclusion - C1-EXCLCASE	System should maintain a lookup with list of case categories for set-off exclusion. Processing logic should be as below • Get all active cases for the account. Account can be primary or secondary entity for that case. • Get case categories for all these cases • If the case category for any of

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						the cases is from the exclusion list, validation should fail. • Check Validation failure option - Validation failure option = Fail case creation/transition. Case should not get created or should not transition status - Validation failure option = Transition status. Case status should be transitioned to the specified status. Set given char value for the given char type (as defined in parameters) • If the case category for any of the cases is not from the exclusion list, validation is successful and process should move to next step. Parameters - Cancel Reason

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Char Type: Characteristic type to set as case characteristic if validation failure option is transition status. Cancel Reason Char Value: Characteristic value for the defined characteristic type. Validation Failure Transition Status: Case transition status in case of validation failure. Validation Failure Option: This option is use to determine action to be taken in case of validation failure. Permissible values are FAIL_CASE_CREATION(fail case creation) and TRANSITION_STATUS(transition status).

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost_Impl	Update Collateral Status in the host	Update Collateral Status in the host
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.InitiateCollateralValuation_Impl	Initiate collateral valuation	Initiate collateral valuation
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.MandatoryCharacteristics_Impl	Mandatory characteristics check for Asset Repo	Mandatory characteristics check for Asset Repo

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateCollateralStatusInTheHost_Impl	Update Collateral Status in the Host: C1-UPCOLLSTS	Update Collateral Status in the host
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.UpdateSetoffExclusionDate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.UpdateSetoffExclusionDate_Impl	Set exclusion date for delinquent account - C1-EXCLROSO	This process will set set-off exclusion date for the delinquent account. Processing will be driven by parameters set for the process. Set-off Exclusion date should be

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getNextCaseStatus() String getNextTransCondition()				updated only if current exclusion date is <= business date. Else, skip all below processing If Cancel Reason char type parameters is not bank • Get the value for the specified char type • This char type should be used to get the offset days from the Lookup for set-off exclusion days • Set-off exclusion date should be set as current business days + offset days. • If mapping for the reason is not found, default value for offset days should be used. If Cancel Reason char type parameters is blank but Reason code is provided • Get the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						corresponding offset days from the lookup for the Reason code • Set-off exclusion date should be set as current business days + offset days. • If mapping for the reason is not found, default value for offset days should be used. Parameters - • Default Offset - Number of days to add to the set-off exclusion date. • Reason Code - Code to fetch offset days from lookup. • Cancel Reason - Characteristic type code to fetch offset days from lookup.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus)	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CancelSetoff	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CancelSetoff_Impl	Cancel Set-off - C1-CANROSO	This algorithm will update the set-off status as "Cancelled" for target accounts associated to the case and having

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()				set-off status as "Pending".
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CompleteSetoff	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.CompleteSetoff_Impl	Complete Set-off - C1-COMPROSO	This algorithm transitions the case to complete. Processing Logic will be as below • Validate that at least one of the target account has set-off status = "Success" and Reversed Flag = "N". • If above validation fails transition to complete should not be allowed and To-do of given To-do Type should be created.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ReverseSetoff	com.splwg.ccb.domain.collection.caseType.specialisedCollections.roso.ReverseSetoff_Im	Reverse Set-off - C1-REVROSO	This algorithm transitions the case to Reversed status. Processing Logic will be as below

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	specific status.	caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()		pl		• Validate below for each target account o Set-off status is not "Success" or If set-off status is "Success" then Reversed Flag should be "Y". o There should be at least one account with Reversed Flag as "Y". • If above validation fails, transition to this status should not be allowed.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateInsuranceCaseDetails	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.UpdateInsuranceCaseDetails_Impl	Algorithm type for update case id for Insurance - C1-UPCASFINS	Algorithm type for update case id for Insurance

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getNextTransCondition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnEnterAlgo	com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnEnterAlgo_Impl	Case Creation on enter processing - C1-CCOENTER	This Algorithm will create a new case for the given Case Type on enter processing. This algorithm accepts Case type as a string, which is required to create a case.
CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase); void setNextCaseStatus(CaseStatus caseStatus);	com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnExitAlgo	com.splwg.ccb.domain.collection.actionObject.actionHistory.CaseCreationOnExitAlgo_Impl	Collection - Case Creation On Exit of Status - C1-CCOE	This algorithm will create a case on the exit processing of the status. This algorihtm will create a case for the account in context and the provided Case type soft parameter.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo); void setBusinessObject(BusinessObject bo); void setEntityId(EntityId id); void setAction(BusinessObjectActionLookup boAction); void setBusinessObjectKey(BusinessObjectInstanceKey boKey); void setOriginalBusinessObject(BusinessObjectInstance boRequest); void setNewBusinessObject(BusinessObjectInstance boRequest);	com.splwg.ccb.domain.collection.actionObject.actionCategory.ActionCategoryValidation	com.splwg.ccb.domain.collection.actionObject.actionCategory.ActionCategoryValidation_Impl	Action category Validation algorithm - C1-ACTCAT	Action category Validation algorithm.This algorithm checks that there should be atleast on action category entity on it.
ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo); void setBusinessObject	com.splwg.ccb.domain.collection.actionObject.actionType.ActionTypeResultTypeValidation	com.splwg.ccb.domain.collection.actionObject.actionType.ActionTypeResultTypeValidation_Impl	Action Type Algorithm Type - C1-ACTTYP	Action Type Algorithm Type vvv

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);
		void setNewBusinessObject(BusinessObjectInstance boRequest);				

ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(Busines
			com.splwg.ccb.domain.collection.actionObject.caseTypeMapping.CaseTypeMappingValidation	com.splwg.ccb.domain.collection.actionObject.caseTypeMapping.CaseTypeMappingValidation_Impl	Case Type Status Mapping Algorithm Type - C1-CASETYMP	Case Type Status Mapping Algorithm Type

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		sObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);
		void setNewBusinessObject(BusinessObjectInstance boRequest);				

CollectionClosingAlgorithmSpot		void setServiceAgreementId(String saId);	com.splwg.ccb.domain.collection.caseCreation.CollectionClosingAlgo	com.splwg.ccb.domain.collection.caseCreation.CollectionClosingAlgo_Impl	Collection - Close Processing Algorithm - C1-CCALG	This algorithm will perform processing done when a Pending Stop Contract is picked up by the Overdue Monitor (collection is to be closed for an account). - It will update the financial balance of the Contract to zero through an adjustment. - Check if there is one or more active promise to pay for the account, if it does it will update

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						the promise to pay status to cancelled and provides the cancel reason - If it is required to close any cases, then it will check if the case has a next status in a final status and if it does will transition to that state. If the case has multiple next statuses which are final statuses, then it will use the default final status defined in the algorithm The following parameters are available and are required: - Adjustment Type used for the adjustment created by this algorithm. - Cancelation Reason Code used while canceling Active PTPs - Is Closing Required Flag to specify if the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						cases associated have to be closed. If this flag is Y but one or more cases cannot be closed the algorithm will generate an error. - Final Default Case Status - If the case to be closed has multiple next statuses that are final and the status specified in this parameter is one of those final statuses, the case will be moved to the status specified in this parameter.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCon	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardShipCaseListUpdate	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardShipCaseListUpdate_Impl	Algorithm type for case list update - C1-CASELIST	Algorithm type for case list update or insert in CI_LIST_MODE_UPDATE table.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		dition()				
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.earlyCollections.CopyCharacteristicsOnCaseCreate	com.splwg.ccb.domain.collection.caseType.earlyCollections.CopyCharacteristicsOnCaseCreate_Impl	Copy Case Characteristics Algorithm Case Type Enter Status Algorithm C1-COPYCHAR	Copy Characteristics Algorithm to copy the Characteristics of recently closed case of a particular Case Category to newly created Case of the same Case Category, when "CONTACT_ALT_SW" in CI_ACCT_EXTN table is set to "Y".

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

InstallationContactInformationAlgoSpot		void setContactPreference(ContactPref contactPref);
		boolean getStatus();
		void setPerson(Person person);
			com.splwg.ccb.domain.collection.collectionLandingPage.ContactInformationCallAdviceAlgo	com.splwg.ccb.domain.collection.collectionLandingPage.ContactInformationCallAdviceAlgo_Impl	Call Advice - Red/Green logic calculation - C1-CALADVICE	Call Advice - Red/Green logic calculation
						• Call Advice will be 'Green' if
						- 'Permission to Call' is Yes And
						- Current Time is within the State level Acceptable Time Limits And
						- Current Time is within the preferred times of the Customer And
						- Current Date is not within the Do Not Disturb Dates Else it will be 'Red'.

TaskCaseValidationAlgorithmSpot		void setTaskCaseMapping(CaseTaskMapping caseTaskMapping);
		void setToDoCase(ToDoCase toDoCase);
		void setToDoEntry(ToDoEntry toDoEntry);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepoDateValidation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.RepoDateValidation_Impl	Task Case Mapping Validation Algorithm - C1-TCVAL	Task Case Mapping Validation Algorithm Algorithm will validate Repossession Date cannot be greater than future date for the process field mapped to Task Type Code and Case Type Code mentioned in soft parameters.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						This algorithm will validate the Repossession Date field only if value is already present. Validation Date can be SYSTEM DATE or POSTING DATE
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateDisputeMonitor	com.splwg.ccb.domain.collection.caseType.earlyCollections.UpdateDisputeMonitor_Impl	Monitoring Algo For Dispute Resolved C1-DISMON	This algorithm is a Monitoring Algo For Dispute Resolved.Used for updating DisputeFlag to 'N'

PostProcessBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
			com.splwg.ccb.domain.collection.vendor.SLAParametersPostProcessAlgo	com.splwg.ccb.domain.collection.vendor.SLAParametersPostProcessAlgo_Impl	SLA Parameters validation algorithm - C1-SLAPARAM	SLA Parameters validation algorithm created for Recovery 2.6.2 release. This algorithm to be called along with CI-SLABO.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);				

ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(Business
			com.splwg.ccb.domain.collection.caseGroup.CaseGroupValidationAlgorithm	com.splwg.ccb.domain.collection.caseGroup.CaseGroupValidationAlgorithm_Impl	Case Group add validation algorithm - C1-CGVAL	Case Group add validation algorithm

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		ObjectInstance boRequest); void setNewBusinessObject(BusinessObjectInstance boRequest);				

GetStrategyAlgorithmSpot		void setServiceAgreementId(String saId);
		void setCaseId(String caseId);
		String getNewCaseType();
			com.splwg.ccb.domain.collection.caseCreation.GetStrategyAlgo	com.splwg.ccb.domain.collection.caseCreation.GetStrategyAlgo_Impl	Collection - Get Strategy Algorithm - C1-COLGS	This algorithm calls the Rules Engine to determine a collection strategy. It is invoked by the Collection Class Overdue Rule - Overdue Monitor Rule
						The following parameters are passed to the Rules Engine :
						- Rule Type (defined in the input parameter)
						- Case Type (if any)
						- Days Past Due
						- Overdue Amount
						- Collection Type

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CollectionClassOverdueMonitorRuleAlgorithmSpot		void setAccount(Account account); Bool getIsProcessingComplete();	com.splwg.ccb.domain.collection.caseCreation.CaseOverdueMonitorRuleAlgo	com.splwg.ccb.domain.collection.caseCreation.CaseOverdueMonitorRuleAlgo_Impl	Create/Move Collection Strategy Cases for Account - C1-COLOMR	This overdue monitor rule algorithm is used to determine the appropriate case type to be used to create a case for an account in collections. It is also responsible for creating the case or for case movement. It will first check for the Collection events (contracts) that are under the account. For Active Contracts it will call the Collection - Get Strategy Algorithm, to determine which Case Type should be used before creating a case. If one or more cases already exist for the Contract they may get closed and new cases created (case movement) if Collection - Get Strategy Algorithm indicates that the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						strategy need to be changed and the current cases can be closed. This algorithm also consider the feature configuration 'C1-NMCSTY' to determine the cases that should not be moved. For Pending Stop Contracts it will call the Collection - Close Processing Algorithm to move the Contract into a closed status. May also close the Cases attached to the contract and reduce the overdue amount on the contract to zero. All other SA statuses are ignored by this algorithm. Notes on the algorithm parameters - Final Default Case Status - If the case to be

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						closed have multiple next statuses that are final and the status specified in this parameter is one of those final statuses, the case will be moved to the status specified in this parameter. - Is Closing required - Flag indicate whether case closing is required or not (Y/N) - Collection Closing Algorithm - This is the algorithm code for Collection - Close Processing Algorithm - Get Strategy Algorithm - This is the algorithm code for Collection - Get Strategy Algorithm
AuditBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo); void setBusinessObject	com.splwg.ccb.domain.collection.caseCreation.CaseSaUpdateBoAuditAlgo	com.splwg.ccb.domain.collection.caseCreation.CaseSaUpdateBoAuditAlgo_Impl	Collection - Case SA Update for Manual Creation - C1-CSAUPD	This Algorithm will update Case SA table for Manual Case Creation

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);
		void setNewBusinessObject(BusinessObjectInstance boRequest);
		void setChangedValues(SchemaInstanceChanges changes);				

InstallationPayPlanAddlGraceDaysAlgorithmSpot		BigInteger getPayPlanAdditionalGraceDays(); void setPaymentPlan(PaymentPlan paymentPlan); void	com.splwg.ccb.domain.collection.payPlan.AdditionalGraceDaysCalculationAlgorithm	com.splwg.ccb.domain.collection.payPlan.AdditionalGraceDaysCalculationAlgorithm_Impl	Promise to Pay - Additional Grace Days Sample Algo - C1-PPADDLGRD	This sample algorithm is called by the Promise to Pay Monitor; it takes the output, which represents additional grace days that should

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		setAccount(Account account);				be added to a promise to pay's scheduled payment date. The algorithm takes the input parameter value and passes it back to the Promise to Pay Monitor as additional grace days.

InstallationPtpThresholdPcForNgpAlgorithmSpot		void setPromiseToPay(PromiseToPay promiseToPay);
		void setAccount(Account account);
		void setTotalPaidAmount(BigDecimal totalPaidAmount);
		void setProcessDate(Date processDate);
		void setLastPaidScheduleDate(Date lastPaidScheduleDate);
		void setLastPaidScheduleBalance(BigDecimal lastPaidScheduleBalance);
			com.splwg.ccb.domain.collection.payPlan.PaymentThresholdPercentageCalculationAlgorithm	com.splwg.ccb.domain.collection.payPlan.PaymentThresholdPercentageCalculationAlgorithm_Impl	Promise to Pay Threshold Percentage - C1-PPTHRESH	This algorithm is called by the Pay Plan Monitor when an expected scheduled payment is not fully met. At this point the promise to pay has been marked to be broken.
						It receives the following inputs from the pay plan monitor
						- Promise to Pay ID
						- Total Amount Paid towards the promise to pay
						- Date (Business Date - Grace Days)
						- Array of Promise to Pay Scheduled

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		void setGraceDays(Big Integer graceDays);
		String getOverrideflag();
		void setValidStartDate(Date validationStartDat e);
		void setValidEndDate(Date validationEndDate);
		;
						Payments balance The algorithm will check if the Total Amount Paid is within the threshold percentage (input parameter) of the Total Scheduled Payments expected.
						If the payments are within the threshold, then the algorithms returns a value of "Y" indicating the promise to pay that was set to be broken should be overridden and remain active/kept Else if the total payments are not within the threshold, then the algorithm returns a value of "N" indicating the promise to pay should be set to broken.

ResultTypePostPr ocessingAlgorithm Spot	This Algorithm spot decides in which status transition has to be made based on	void setActionEntity(Str ing actionEntity); void setActionSourceId	com.splwg.ccb.do main.collection.act ionObject.actionTy pe.ResultTypePos tProcCaseTransAl	com.splwg.ccb.do main.collection.act ionObject.actionTy pe.ResultTypePos tProcCaseTransAl	Result type Post Processing Case Transition Algo - C1-RTPCC	If specified on the Result Type, this algorithm will be invoked when the corresponding

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

	processing of result.	(String actionSourceId);
		void setActionSourceStatusCode(String actionSourceStatusCd);
		void setActionId(String actionId);
		void setActionType(ActionType actionType);
		void setResultType(ResultType resultType);
		boolean getIsProcessingComplete();
			go	go_Impl		result is recorded for a Case (Action/Result UI). This can be used to transiton the case from the current status to the next possible status as follow,
						- This algorithm has a parameter Output Status i.e. next possible status, so for case transition it will be checked whether Output Status is one of the next possible status, if YES, it will transiiton the case to that status
						- This algorihtm has a parameter Input Status, which will be checked against the current status of the Case. This is an optional parameter. If specified the Case transition will happen only when the current status of the case matches with this

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						parameter.
BusinessObjectEnterStatusAlgorithmSpot			com.splwg.ccb.domain.collection.inboundCustomer.CreateEntityAlgo	com.splwg.ccb.domain.collection.inboundCustomer.CreateEntityAlgo_Impl	Inbound Customer algorithm - C1-IN-CUST	This algorithm will create the Person,Account,SA, SAcollection object and Adjustment from FACT clob. This is a Business Object Status Enter algorithm. The algorithm perform the following actions - Retrieve the XML message containing the customer information, which stored on the FACT MO. - Read the XML and determine if the action is to add a new customer or update an existing customer. - It may create a combination of Person, Account, Contract, Contract, or Adjustment, depending on what was contained in the

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						XML. - If all objects are created successfully it will transition the lifecycle to the 'Created' status - Else if any of the objects experienced and error while processing it will transition the lifecycle to the "Rejected" status. It has two parameters , both optional. - Account Id Type identifies the Account Identifier Type used to locate the account in ORMB. - Person Id Type identifies the Person Identifier Type used to locate the person in ORMB.
ResultTypePreProcessingAlgorithmSpot			com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePreProcAlgo	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePreProcAlgo_Impl	Result Type Pre-processing Algorithm Type - C1-RSTPRE	Result Type Pre-processing Algorihtm Type

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity); void setActionSourceId(String actionSourceId); void setActionSourceStatusCode(String actionSourceStatusCd); void setActionId(String actionId); void setActionType(ActionType actionType); void setResultType(ResultType resultType); boolean getIsProcessingComplete();	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcAlgo	com.splwg.ccb.domain.collection.actionObject.actionType.ResultTypePostProcAlgo_Impl	Result Type Post-processing Algorithm Type - C1-RSTPOST	Result Type Post-processing Algorithm Type

AdhocCharacteristicValueValidationAlgorithmSpot	This algorithm spot is invoked on characteristic adhoc values in order to:
	1) validate that the value is correct
	2) possibly perform a reformat
		void setFormatOnly(boolean formatOnly);
		void setCharacteristicType(CharacteristicType type);
		void setAdhocValue(String value);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.CustomAdhocDateValidationAlgComp	com.splwg.ccb.domain.collection.caseType.specialisedCollections.CustomAdhocDateValidationAlgComp_Impl	Characteristic Type :Validate Date Field (Custom) - C1-ADHDATE	Custom Date validation
						This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time.
						The Parameters From Date and To

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	of the value prior to storing on the table	String getReformattedValue(); boolean isValidAdhoc();				Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is specified, they must be in the format YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field. The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include:

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS. Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
DialerResultsUploadAlgorithmSpot		void setAccountNumber(String accountNumber); void setCustomerNumber(String customerNumber);	com.splwg.ccb.domain.collection.dialerResultUpload.DialerResultUploadAlgo	com.splwg.ccb.domain.collection.dialerResultUpload.DialerResultUploadAlgo_Impl	Algorithm Type for Dialer Results Upload - C1-DLRRSUPLD	Algorithm Type for Dialer Results Upload
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCaseCreationActivity	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCaseCreationActivity_Impl	Algorithm for Hardship case creation activity - C1-CRTHDSP	This Algorithm is responsible for making a Hardship Case entry on the Party,when the Hardship case is created.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

CaseTypeExitStatusAlgorithmSpot	The purpose of the algorithm spot is to perform additional logic when a Case transitions out of the current status to the next status.	void setCase(ToDoCase toDoCase);
		
		void setNextCaseStatus(CaseStatus caseStatus);
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AbortApprovalWorkItemsAlgo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AbortApprovalWorkItemsAlgo_Impl	This Algorithm is used to abort Approval work item - C1-ABORTAPP	This algorithm is used to abort approval work item. Input to the algorithm is composite name,instance title and case status exclusion list. If next case status is present in case status exclusion list then work item instance is not aborted. caseStatusExclusionList:- comma seperated list of case status for which approval work item shouldn't be aborted. Composite Name:- Fully qualified approval class name. Instance Title:- Approval instance work item title prefix. Example - Input parameters and it's applicable value for ROSO Process, Composite

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						Name:- com.ofss.fc.workflow.process.ROSOProcessForApproval Instance Title:- ROSO_CASE_ Value of the above parmeters is dependent upon the SOA approval work flow.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelFinancialHardshipApprovalReqAlgo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.CancelFinancialHardshipApprovalReqAlgo_Impl	Cancel Process Approval Request:Financial Hardship - C1-CANFHAPPR	This algorithm will cancel all pending approval requests for the case. Example for parametervalues for hardship Process: Composite Name:- com.ofss.fc.workflow.process.FinancialHardshipProcessForApproval Instance Title:- FINANCIAL_HARDSHIP_CASE_ Value of the above parmeters depends upon the SOA approval work flow.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCharAssociation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.HardshipCharAssociation_Impl	Hardship Characteristic Association - C1-FHCHARASC	Hardship Characteristic Association
PrePopulatedRuleFactsAlgorithmSpot		List<RuleFactInfoDto> getPrePopulatedFacts();	com.splwg.ccb.domain.collection.PrePopulatedSystemFacts	com.splwg.ccb.domain.collection.PrePopulatedSystemFacts_Impl	Pre-Populated system facts to be used for Rule - C1-PPSF	This algoritm is used to populate input system fact for Rule.It used as an input to RuleFactPopulation algorithm. System Facts populated through this algoritm are SystemDate and PostingDate. This is sample implementation to provide populated facts to RuleFactPopulation algoritm.User can create his own algoritm type

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						based on his requirement (Algoritm Entity must be Rule Execution - Pre Populated Rule Facts).
ToDoTypeToDoPostProcessAlgorithmSpot	This Algorithm spot is used for notifying task completion and also for allocating task to vendor.	void setOldToDoEntryDTO(ToDoEntry_DTO oldDTO); void setNewToDoEntry(ToDoEntry newToDoEntry);	com.splwg.ccb.domain.collection.batch.algorithm.AssignTaskToQueueAlgorithm	com.splwg.ccb.domain.collection.batch.algorithm.AssignTaskToQueueAlgorithm_Impl	Assign Batch level TODOs(task) to a queue. - C1-ASGNTASK	Assign Batch level TODOs(task) to a queue.
AdhocCharacteristicValueValidationAlgorithmSpot	This algorithm spot is invoked on characteristic adhoc values in order to: 1) validate that the value is correct 2) possibly perform a reformat of the value prior to storing on the table	void setFormatOnly(boolean formatOnly); void setCharacteristicType(CharacteristicType type); void setAdhocValue(String value); String getReformattedValue(); boolean isValidAdhoc();	com.splwg.ccb.domain.collection.caseType.specialisedCollections.ProductAdhocDateValidationAlgComp	com.splwg.ccb.domain.collection.caseType.specialisedCollections.ProductAdhocDateValidationAlgComp_Impl	Validate Date Field :Custom - C1-ADHV-DTD	This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time. The Parameters From Date and To Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is specified, they must be in the format

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field. The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used: YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						YYYY-HH:MI:SS. Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system.

AdhocCharacteristicValueValidationAlgorithmSpot	This algorithm spot is invoked on characteristic adhoc values in order to:
	1) validate that the value is correct
	2) possibly perform a reformat of the value prior to storing on the table
		void setFormatOnly(boolean formatOnly);
		void setCharacteristicType(CharacteristicType type);
		void setAdhocValue(String value);
		String getReformattedValue();
		boolean isValidAdhoc();
			com.splwg.ccb.domain.collection.caseType.CharAdhocDateValidation	com.splwg.ccb.domain.collection.caseType.CharAdhocDateValidation_Impl	Characteristic Date field Validation:C1-CHARDTVAL	This algorithm is used to validate that an ad hoc characteristic value is a date or a date/time.
						
						The Parameters From Date and To Date are both optional. The algorithm will check that the date is later than the From Date (if entered) and/or earlier than the To Date (if entered). If either value is

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						specified, they must be in the format YYYYMMDD. These parameters are ignored if the characteristic value is a date/time field. The various Date Format parameters are used to control the format in which the date/time is entered by a user. You must supply at least one format in parameter 3. The other parameters exist in case you allow multiple date formats to be used. Examples of date formats include: YYYYMMDD, DD/MM/YYYY, DD-MM-YYYY, MM/DD/YYYY, YYYY-MM-DD, etc. However, only three types of date/time formats can be used:

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
						YYYY-MM-DD-HH:MI, MM-DD-YYYY-HH:MI:SS, and DD-MM-YYYY-HH:MI:SS. Regardless of the format entered by the user, the date is stored in the format defined by parameter 3. We strongly recommend this parameter be set to YYYY-MM-DD for dates and YYYY-MM-DD-HH:MI:SS for date/time fields as this is how all dates are stored in our system. Parameter 9 : valid values are true/false. When Business date validation required is true, algorithm will validate the given date to check if its a valid business date.
CaseTypeAutoTransitionAlgorithmS	This algorithm type is used to perform auto	void setCase(ToDoCase toDoCase);	com.splwg.ccb.domain.collection.caseType.specialise	com.splwg.ccb.domain.collection.caseType.specialise	Retry Case in Error - C1-	This algorithm is plugged-in on auto-transition of

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
pot	transition processing for a Case.	Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	dCollections.financialHardship.RetryCaseInErrorForHardshipApp	dCollections.financialHardship.RetryCaseInErrorForHardshipApp_Impl	RCASEE	error states and attempts to retry validation, completion or wait if the To Do Entry associated is not being worked on. The retry will be performed only until the input Maximum Number of Retries is reached.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.PerformQueueAllocation	com.splwg.ccb.domain.collection.caseType.specialisedCollections.financialHardship.PerformQueueAllocation_Impl	Allocate Queue for Customer Level Case	Allocate Queue for Customer Level Case. Only Queue Allocation would be done. User Allocation would be skipped for customer level cases.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ValidateBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo);
		void setBusinessObject(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);
		void setNewBusinessObject(BusinessObjectInstance boRequest);
			com.splwg.ccb.domain.collection.address.PersonCollectionAddressValidation	com.splwg.ccb.domain.collection.address.PersonCollectionAddressValidation_Impl	Person Address - Collection - C1-PERADDRC	This Algorithm is a reference implemenation for consulting. This algorithm will be used for validating Person address as per requirment

PostProcessBusinessObjectAlgorithmSpot		void setMaintenanceObject(MaintenanceObject mo); void setBusinessObject	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPostProcess	com.splwg.ccb.domain.collection.address.PersonCollectionAddressPostProcess_Impl	Person Address update post processing algorithm. - C1-PADDPOST	Person Address update post processing algorithm.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		(BusinessObject bo);
		void setEntityId(EntityId id);
		void setAction(BusinessObjectActionLookup boAction);
		void setBusinessObjectKey(BusinessObjectInstanceKey boKey);
		void setOriginalBusinessObject(BusinessObjectInstance boRequest);				

ICollectionContactPointPostProcessingSpot		void setOldEntity(ContactPrefColl oldEntity); void setNewEntity(ContactPrefColl newEntity);	com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot	com.splwg.ccb.domain.collection.address.CollectionContactPointPostProcessingSpot_Impl	Person Contact Point Update - Post Processing - C1-PERCONTPP	This is a reference implementation of Post processing Algo. Customization team can utilize this hook.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTra	com.splwg.ccb.domain.collection.caseType.UpdateSelfServeFlag	com.splwg.ccb.domain.collection.caseType.UpdateSelfServeFlag_Impl	Update Self Serve Flag Algorithm - C1-SELFSERVE	Action -soft parameter mentioned in algorithm type which will update the self_serve flag to Y or N. If Action = Set make Self Serve

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		nsition() String getNextCaseStatus() String getNextTransCondition()				Flag = Y If Action = Reset make Self Serve Flag = N

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
			com.splwg.ccb.domain.collection.actionObject.actionHistory.FollowUpResultTaskAlgo	com.splwg.ccb.domain.collection.actionObject.actionHistory.FollowUpResultTaskAlgo_Impl	Create Task for Self Serve Request for Assistance transaction - C1-FLWRTSK	This algorithm will be used to Create Task post Follow Up.Possible values are Task For,Task Type and Task Queue

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		boolean getIsProcessingComplete();				
CaseTypeAutoTransitionAlgorithmSpot	This algorithm type is used to perform auto transition processing for a Case.	void setCase(ToDoCase toDoCase); Bool getShouldAutoTransition(); CaseStatus getNextCaseStatus(); String getNextTransCondition();	com.splwg.ccb.domain.collection.scra.algorithm.ActiveServiceAlgorithm	com.splwg.ccb.domain.collection.scra.algorithm.ActiveServiceAlgorithm_Impl	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. Code - C1-ACTMEMCHK	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.
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is	void setCase(ToDoCase toDoCase) void setCaseOriginalSt	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.A	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.A	If any of the customers associated with the repossession case satisfy below	If any of the customers associated with the repossession case satisfy below

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	moved into specific status.	atus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	ctiveMilitaryServiceCheckonAssociatedCustomers	ctiveMilitaryServiceCheckonAssociatedCustomers_Impl	criteria block repossession initiation. The customer is a Service Member and The customer has not waived his SCRA Protection and (The customer is in Active Service or the number of days since the end date of customers last active service < X days or the service member is missing in action) Error Message: "Repossession case cannot be initiated. SCRA checks failed." Code - C1-BLOCKREPO	criteria block repossession initiation. The customer is a Service Member and The customer has not waived his SCRA Protection and (The customer is in Active Service or the number of days since the end date of customers last active service < X days or the service member is missing in action) Error Message: "Repossession case cannot be initiated. SCRA checks failed."

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		
		void setActionSourceId(String actionSourceId);
		
		void setActionSourceStatusCode(String actionSourceStatusCd);
		
		void setActionId(String actionId);
		
		void setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();
			com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatusCodePostLiquidationPostProcessing	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatusCodePostLiquidationPostProcessing_Impl	Metro 2 Reporting – Account Status Code post Liquidation C1-ASCLIQUA	This algorithm is used for Metro 2 Reporting – Account Status Code post Liquidation

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatuscodeEnterProcessingAlgo	com.splwg.ccb.domain.collection.caseType.specialisedCollections.AssetRepo.algorithms.Metro2AcctStatuscodeEnterProcessingAlgo_Impl	Metro 2 Reporting – Account Status Code C1-ASCREPO	This algorithm is used for Metro 2 Reporting – Account Status Code
ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity); void setActionSourceId(String actionSourceId); void setActionSourceStatusCode(String actionSourceStatusCd); void setActionId(String actionId); void	com.splwg.ccb.domain.collection.caseType.earlyCollections.Metro2CompliancCodePostProcessingAlgo	com.splwg.ccb.domain.collection.caseType.earlyCollections.Metro2CompliancCodePostProcessingAlgo_Impl	Metro 2 Reporting - Compliance condition code C1-COMCODE	This algorithm is used for Metro 2 Reporting - Compliance condition code

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		setActionType(ActionType actionType);
		
		void setResultType(ResultType resultType);
		
		boolean getIsProcessingComplete();				

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2CheckForOpenStatusEnterProcessing	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2CheckForOpenStatusEnterProcessing_Impl	Metro 2 Reporting - Marking Account as Close - C1-CFOSEP	The logic is incorporated for Metro Algorithm only if a Account is close than it should be marked as Close
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInformationIndicator	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInformationIndicator_Impl	Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.	Set CII = X based on Chapter entered in Filing Information for all customers associated to the case.

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
		getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()			Code - C1-CONINFOIN	
CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.	void setCase(ToDoCase toDoCase) void setCaseOriginalStatus(CaseStatus caseStatus) Bool getShouldAutoTransition() String getNextCaseStatus() String getNextTransCondition()	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInfoIndiChap13PostDis	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.Metro2ConsumerInfoIndiChap13PostDis_Impl	If any associated secured account without confirmed plan on it report CII = Q Else Report CII = G for Chapter 12 Report CII = H for Chapter 13. Code - C1-CIIPSTDIS	If any associated secured account without confirmed plan on it report CII = Q Else Report CII = G for Chapter 12 Report CII = H for Chapter 13.

ResultTypePostProcessingAlgorithmSpot	This Algorithm spot decides in which status transition has to be made based on processing of result.	void setActionEntity(String actionEntity);
		void setActionSourceId(String actionSourceId);
		void setActionSourceStatusCode(String actionSourceStatusCd);
			com.splwg.ccb.domain.collection.caseType.earlyCollections.ConsumerNowLocated	com.splwg.ccb.domain.collection.caseType.earlyCollections.ConsumerNowLocated_Impl	This algorithm will set the given CII Code for the party id provided as result characteristics Result type post processing algo C1-CGCLC	This algorithm will set the given CII Code for the party id provided as result characteristics

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary

		void setActionId(String actionId);
		void setActionType(ActionType actionType);
		void setResultType(ResultType resultType);
		boolean getIsProcessingComplete();				

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into specific status.
	The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic
		void setCase(ToDoCase toDoCase);
		void setCaseOriginalStatus(CaseStatus caseStatus);
		Bool getShouldAutoTransition();
		String getNextCaseStatus();
		String getNextTransCondition();
			com.splwg.ccb.domain.collection.caseType.earlyCollections.CreditGrantorCannotLocateConsumer	com.splwg.ccb.domain.collection.caseType.earlyCollections.CreditGrantorCannotLocateConsumer_Impl	Automatically set for all borrowers the account the CII Code in skip tracing status on entering a case status
					Enter Processing.
					C1-CNLREM
						Automatically set for all borrowers the account the CII Code in skip tracing status on entering a case status

CaseTypeEnterStatusAlgorithmSpot	The purpose of the algorithm spot is to execute the business logic when Case is moved into	void setCase(ToDoCase toDoCase); void setCaseOriginalStatus(CaseStatus	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.SetDPDOut	com.splwg.ccb.domain.collection.caseType.specialisedCollections.bankruptcy.SetDPDOutstandingAmount_I	Set the DPD and Outstanding amount to all associated accounts on entering the status	On creation of a case the algorithm will Set DPD and Outstanding amount to all associated

Algorithm Spot	Spot Detail	Spot Interface Funtions	Collections Algorithm Component	Collections Algorithm Impl	Collections Algorithm Description and Code	Algorithm Summary
	specific status. The specific sample algorithm creates To Do entry and links the Case to it as FK Characteristic	caseStatus); Bool getShouldAutoTransition(); String getNextCaseStatus(); String getNextTransCondition();	standingAmount	mpl	– Enter Status - C1-SETDPD	accounts

GenericAlgorithm Spot	This is generic algorithm spot which can be used to generate Generic Algorithm of type AlgorithmSpot	void setPerson(Person person);
		
		void setToDoCase(ToDoCase toDoCase);
		
		void setAccount(Account account);
		
		Bool getDMDCVerificationRequired();
			com.splwg.ccb.domain.collection.dmdc.VerifyDMDCDetailsAlgorithm	com.splwg.ccb.domain.collection.dmdc.VerifyDMDCDetailsAlgorithm_Impl	This algorithm is used to check whether SCRA verification request should call to DMDC or not based on number of days passed.
					
					Code - C1-DMDCREQ
						This algorithm is used to check whether SCRA verification request should call to DMDC or not based on number of days passed.