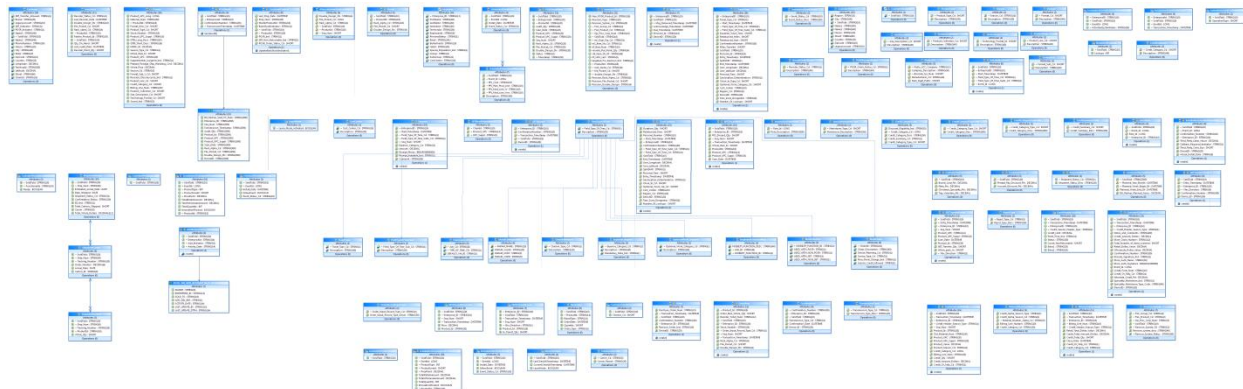


1 Detailed Design

1.1 Data Model



1.2 Data Source

The following table identifies the data sources for each MBO. In case this is a Web Service, only the relevant part of the URL is provided. For CRM, the function module and the output table are identified. Not all MBO have a data source binding. This section is only for the read operations. The create operations are defined in another section of this document. The data source in this section is the one that were used during development. Address can be different for other environment.

Given the very important data volume, the initial data load is done by a process called Bulk Loading. This process bypass existing way to load the data and it inserts the rows directly in the Cache Database. Once this is complete, a DCN process will update the Cache Database daily. The Bulk Load inserts the data in the MBO that are in a DCN cache group.

The names of the data sources are very important. SUP configuration relies on these exact names to load the data during a deployment.

Data Source Type	Name	Address
Web Service	webservice	http://apired.master.hmkad.hallmark.com:8000/api
CRM	crm	App server: adc30fe.hallmark.com System ID: DC3 System number: 62

1.2.1 Web Service Data Source

This section identifies the relevant information to locate the information from the data source. The address can be different but the information contained in the table below should remain the same, in every environment. For the Web Service, the corresponding XSD filename is provided.

There is another section in the document that contains the unbounded MBO. It is possible that a MBO from this list may also be in the unbounded section. The URL and XSD that are provided in this section were used during the development. The DCN processes were not ready until the Sprint 4, so it was necessary to have a way to get data during development, without relying on the DCN. The list of MBO in this section represents this situation. Refer to the Cache Group section to find which MBO is unbounded. Every MBO that are in a DCN cache group can be considered unbounded but they might had a data source in the Web Service earlier during the development.

MBO	Description	Data Source Details (URL and XSD)
<i>AlternateInventoryLocation</i>	Represents the alternate pocket information for a specific product that can be in different pocket for the same customer.	<i>/AlternativeInventoryLocation</i> <i>AlternativeInventory.xsd.xml</i>
BillingLineSummary	Represents a billing line summary that is sent	/BillingLineSummary BillingLineSummary.xsd.xml
<i>Confirmation</i>	<i>Represents the confirmation number data that is stored after each transmission.</i> <i>DEPRECATED</i>	<i>/Confirmation</i> <i>Confirmation.xsd.xml</i>
CreditCategory	Represents a credit category. A product belongs to a specific category. DEPRECATED	/CreditCategory CreditCategory.xsd.xml
CreditCategoryTally	Represents a credit category. A product belongs to a specific category.	/CreditCategory CreditCategoryTally.xsd.xml
CreditCategoryTypeTally	Represents the credit category type for a tally.	/CreditCategoryType CreditCategoryTypeTally.xsd.xml
CreditSummaryTally	Represents credit summary information for a tally.	/CreditCategorySummary CreditCategorySummaryTally.xsd.xml
CustomerCreditDetail	Represents the details for a credit that is specific to a product or a pocket.	/SoldToCustCreditDetail CustomerCreditDetails.xsd.xml
CustomerCreditHeader	Represents the header of a customer credit. It contains the CustomerCreditDetail and BillingLineSummary informations.	/SoldToCustCreditHeader CustomerCreditHeader.xsd.xml
<i>CustomerPlannedHours</i>	<i>Represents an interval of time in which the number of hours a customer is open for business.</i>	<i>/BuPartnerSoldToPlannedHours</i> <i>CustomerPlannedHours.xsd.xml</i>

<i>CustomerProduct</i>	<i>Represents the association between a specific product and a customer.</i>	<i>/CustomerProduct CustomerProduct.xsd.xml</i>
<i>CustomerTerms</i>	<i>Represents the customer terms in regard to his credit.</i>	<i>/SoldToCustomerTerms CustomerTerms.xsd.xml</i>
<i>CustomerTermsSeason</i>	<i>Represents the customer terms for seasonal product in regard to his credit.</i>	<i>/SoldToCustomerTermsSeason CustomerTermsSeason.xsd.xml</i>
<i>DecodeOrderStatus</i>	<i>Represents the description for the different order statuses.</i>	<i>/DecodeOrderStatus DecodeOrderStatus.xsd.xml</i>
<i>EventRole</i>	<i>Represents the role that an employee had for a specific installation at a specific customer</i>	<i>/EventRole EventRole.xsd.xml</i>
<i>Expense</i>	<i>Represents an expense that has been done by the employee.</i>	<i>/ExpenseDtl ExpensesDtl.xsd.xml</i>
<i>Format</i>	<i>Represents the description for a product format.</i>	<i>/Format Format.xsd.xml</i>
<i>FormatSub</i>	<i>Represents the description for a product sub-format</i>	<i>/FormatSub1 FormatSub1.xsd.xml</i>
<i>InstallationChecklistReporting</i>	<i>Represents a check list report after an installation has been completed.</i>	<i>/InstallationChecklistReporting InstallationCheckListReporting.xsd.xml</i>
<i>InstallationEventRole</i>	<i>Represents the description for the different roles an employee can have for an installation</i>	<i>/Roles InstallationRole.xsd.xml</i>
<i>InstallationEvent</i>	<i>Represents the installation that was done by an employee at a specific customer at a specific time.</i>	<i>/InstallationEvent InstallationEvent.xsd.xml</i>
<i>InstallationEventStatus</i>	<i>Represents the description for an installation event status.</i>	<i>/EventStatus EventStatus.xsd.xml</i>
<i>Inventory</i>	<i>Represents the inventory data for a specific product at a specific customer.</i>	<i>/Inventory Inventory.xsd.xml</i>
<i>InventoryPocket</i>	<i>Represents the master chart data for a specific product at a specific customer. This is used to activate system after an installation is complete.</i>	<i>/InventoryPocket InventoryPocket.xsd.xml</i>
<i>MaterialNumberAlpha</i>	<i>Represents the product that are eligible to a credit based on the first 4 characters.</i>	<i>/CreditAlpha Credit_Alpha.xsd.xml</i>
<i>MealAlert</i>	<i>Represents a meal alert that has been displayed on the device and acknowledged by the user.</i>	<i>/MealAlert meallaert.xsd.xml</i>

Order	Represents a complete order header done by a user. It contains multiple OrderItem	/OrderHeader OrderHeader.xsd.xml
OrderItem	Represents a specific product that is ordered by the user.	/OrderDetail OrderDetail.xsd.xml
OrderRecordInputType	Represents the description for the different way that an order can be created on the device.	/OrderInputRecordType OrderInputRecordType.xsd.xml
<i>Pocket</i>	<i>Represents the pocket information for a specific product at a specific customer.</i>	<i>/Pocket Pocket.xsd.xml</i>
POSRAudit	Represents a POSRAudit header that has been done by an employee.	/PosrAuditHdr posrAudit.xsd.xml
POSRR reorderStatus	Represents the description of the different POSRAudit reorder status.	/PosrReorderStatus PosrReorderStatus.xsd.xml
PrefixUPC	Represents UPC prefix code that determine if a product belongs to Hallmark.	/UccCode UccCode.xsd.xml
<i>Product</i>	<i>Represents a specific product. It acts as a product catalogue.</i>	<i>/Product Product.xsd.xml</i>
ProductCollection	Represents the description for the product collection.	/ProductCollection ProductCollection.xsd.xml
ProductLine	Represents the description for the product line.	/ProductLine ProductLine.xsd.xml
ProductType	Represents the description for the product type.	/ProductType ProductType.xsd.xml
RejectType	Represents the description for the different reject type used in the backorder.	/RejectType RejectType.xsd.xml
<i>RevisionSystem</i>	<i>Represents the different system that can be activated for a specific customer.</i>	<i>/RevisionSystemCode RevisionSystemCode.xsd.xml</i>
SBTDiscard	Represents a SBT Discard header. It contains SBTDiscardItem.	/SBTDiscardHeader SBtDiscardHeader.xsd.xml
SBTDiscardItem	Represents a specific product that can be discarded.	/SBTDiscardDetail SBTDiscardDetail.xsd.xml
SBTNonReturnableUPC	Represents the products that are not eligible to a SBT Discard.	/SBTNonReturnableUPCOnly NonReturnableUPC.xsd.xml
SBTTransferDetail	Represents the SBT Transfer detail for a specific product that is moved from and to a specific customer.	/SbtTransferDetails SBTTransferDetail.xsd.xml

SBTTransferHeader	Represents a SBT Transfer header that contains multiple SBT Transfer details	/SbtTransferHeader SBTTransferHeader.xsd.xml
ScheduledStoreEvent	Represents the installation events that are planned at a specific customer.	/ScheduledStoreEvent ScheduleStoreEvent.xsd.xml
ScheduledStoreEventPPL	Represents all the system that has to be activated for a planned installation	/ScheduledStoreEventPpl ScheduleStoreEventPPL.xsd.xml
Season	Represents the description for a seasonal product type.	/Season Season.xsd.xml
SizeDescription	Represents the description for the product size description	/SizeDescriptions SizeDescriptions.xsd.xml
SpecialityMarkdownType	Represents the description for the speciality markdown type.	/SpecialityMarkdownType SpecialityMarkdownType.xsd.xml
TechnologyFormat	Represents the description for the product technology format.	/TechnologyFormats TechnologyFormats.xsd.xml
TimeTracking	Represents the time report that is created on the device and records the time spent working by an employee.	/TimeTrackingExpense TimeTrackingExpense.xsd.xml
TrackCarrierCode	Represents the carrier code.	/ShpmtTrackedCarrCd TrackCarrierCd.xsd.xml
TransmissionType	Represents the description for the transmission type.	/TransmissionType TransmissionType.xsd.xml
TravelTime	Represents the mileage that is recorded by the device done by an employee.	/TimeTrackingExpense TimeTrackingExpense.xsd.xml
ApkVersionTracking	Responsible for APk version tracking of the cc application	/ApkVersionTracking ApkVersionTracking.xsd.xml

1.2.2 CRM Data Source

This section identifies the MBO that are loaded by a RFC. Each row contains the name of the MBO and the corresponding RFC with the output table. In case there is a load parameter, it is distinguished from the output table.

MBO		Function Module & Output Table
Customer	Represents an account	ZSOE1_FM_GET_ACCOUNTS – ET_ACCOUNTS1
Employee	Represents an employee	ZSOE1_FM_GET_EMPLOYEES Input: IV_NAME (Char) Username / EnterpriseID Output: ET_EMPLOYEES1
CustomerSelection	Represents all the account and this is used to add new account on the device.	ZSOE1_FM_GET_ACCOUNTS – ET_ACCOUNTS1
CustomerAssigned	Represents all the account assigned to a particular employee.	ZSOE1_FM_GET_EMPLOYEE_ACCOUNTS Input: IV_NAME (Char) Username / EnterpriseID Output: ET_EMPL_ACCTS1
CostCenter	Represents the different cost centers on which an employee can charge administration time.	ZSOE1_FM_GET_COSTCENTER – ET_COSTCENTERS1
FieldTypeOfTime	Represents the description type of time spent by an employee.	ZSOE1_FM_GET_CONFIG_TABLES – ET_TYP_TM
TravelType	Represents the description for the travel type.	ZSOE1_FM_GET_CONFIG_TABLES – ET_TRVL_TYP
FieldTypeOfTimeSub	Represents the description for the sub type of time spent by an employee.	ZSOE1_FM_GET_CONFIG_TABLES – ET_TYP_TM_SB1
EmployeeJobType	Represents the description for the employee job type.	ZSOE1_FM_GET_CONFIG_TABLES – ET_JOB_TYPET
GlobalConfiguration	Represents all the global configuration values that are used to configure the application.	ZSOE1_FM_GET_CONFIG_TABLES – ET_GBL_CONFIG
CheckinMethod	Represents the description for the check in method type used by the employee.	ZSOE1_FM_GET_CONFIG_TABLES – ET_CHKIN

ExpenseCategory	Represents the description for an expense.	ZSOE1_FM_GET_CONFIG_TABLES – ET_EXP_CATG
OptionalWorkCategory	Represents the description for optional work category type.	ZSOE1_FM_GET_CONFIG_TABLES – ET_OPTWRK_CAT
EmployeeFunctionMap	Represents the different mapping between a job type and the allowed action in the application.	ZSOE1_FM_GET_CONFIG_TABLES – ET_JOB_FUNC
StoreFunction	Represents the different mapping between an account type and the allowed action in the application.	ZSOE1_FM_GET_CONFIG_TABLES – ET_FNCTN_FLTR
StoreChain	Represents the chain for which a specific account belongs to.	ZSOE1_FM_GET_CONFIG_TABLES – ET_CHAIN_VALUES

1.2.3 Unbounded MBO

This section contains the MBO that are in the model but are not bounded to any data source. These MBO can only be loaded using the DCN process (or Bulk Loading). The list contains the MBO that can be modified directly in the Workspace and does not require a XSD file.

Unbonded MBO	Description
Inventory	<i>Represents the inventory data for a specific product at a specific customer.</i>
InventoryPocket	<i>Represents the master chart data for a specific product at a specific customer. This is used to activate system after an installation is complete.</i>
ShipmentHeader	Represents the Shipment header information. It contains multiple Shipment Carton. A Shipment is broken down in multiple shipment carton.
ShipmentProduct	Represents a specific product that is part of a Shipment Carton
ShipmentCarton	Represents a specific shipment that is part of a Shipment Header. The deliveries are based on this concept. A Shipment Header is arrived when all the Shipment Carton are arrived.
AddRemoveAccountHistory	Keep track of the add and remove account activity on the device per user.
Product	
Backorder	Represents the orders that are backordered for a specific customer.
FamilyFile	Represents the relation between an old product and a new product that are in the same family.
RevisionBkupFilePkt	Represents revision backup file pocket details for a product.

1.2.4 Local MBO

This section contains the list of the local MBO that are in the package.

Local MBO	
AddedRemovedCustomerLocal	CreditHeaderLocalInfo
CustomerCheckinTypeLocal	CustomerFunctionnalityLocal
CustomerPendingLocal	InstallationEventActivatedSystemLocal
InstallationEventOrderLocal	InstallationUnassignedProductLocal
LearnMode	NumberOfLookup
POSRAuditItem	SBTDiscardItemLocal
SBTInTransitLocal	ScheduleStoreEventLocal
StoreVisitTime	

1.3 Package

Package Name	Component
FSOHMK	The MBO package that is visible from SCC
FSOHallmark	Application Identifier to be used on the client device
com.mdc.sap.fso.mbo	Generated Code Package Name

1.4 Cache groups

The MBO package contains several cache groups in order to ensure flexibility in the data loading process. Groups that were identified to have less than 10000 records are fetched directly from the Web Service. Groups with over 10000 records are loaded with a DCN process, developed and maintained by the customer. Groups from CRM system use On-Demand or Scheduled Refresh.

Cache group name	MBOs	Cache Policy
AlternateInventoryLocationCG	AlternateInventoryLocation	DCN
AuditCG	POSRAudit	DCN
BackorderCG	Backorder	DCN
ConfirmationCG	Confirmation	DCN
CreditCG	BillingLineSummary CustomerCreditDetail CustomerCreditHeader	DCN

CustomerAssignedCG	CustomerAssigned	On-Demand; Cache validity 5 minutes; Partitioned by user and device.
CustomerCG	Customer	Scheduled Refresh; Every 24 hours.
CustomerPlannedHoursCG	CustomerPlannedHours	DCN
CustomerProductCG	CustomerProduct	DCN
CustomerSelectionCG	CustomerSelection	Scheduled Refresh; Every 24 hours.
DecodeCG	CreditCategoryTally CreditCategoryTypeTally CreditSummaryTally DecodeOrderStatus Format FormatSub InstallationEventRole OrderRecordInputType POSRReorderStatus PrefixUPC ProductCollection ProductLine ProductType RejectType Season ShipmentStatus SizeDescription SpecialMarkdownType TechnologyFormat TrackCarrierCode TransmissionType	Scheduled Refresh; Every 24 hours.
DecodeCreditCG	CustomerTerms CustomerTermsSeason	DCN
Default	AddRemoveAccountHistory CreditCategory	DCN
EmployeeCG	Employee	On-Demand; Cache validity 5 minutes; Partitioned by user and device.

InstallationCG	EventRole InstallationChecklistReporting	DCN
InventoryCG	Inventory	DCN
InventoryPocketCG	InventoryPocket	DCN
MaterialAlphaCG	MaterialNumberAlpha	Scheduled Refresh; Every 24 hours.
OrderCG	Order OrderItem	DCN
PocketCG	Pocket	DCN
ProductCG	Product	DCN
RevisionSystemCG	RevisionSystem	Scheduled Refresh; Every 24 hours.
SBTDiscardCG	SBTDiscard SBTDiscardItem SBTNonReturnableUPC	DCN
SBTTransferCG	SBTTransferDetail SBTTransferHeader	DCN
ShipmentCG	ShipmentHeader ShipmentCarton ShipmentProduct	DCN
StoreCG	ScheduledStoreEvent ScheduledStoreEventPPL	DCN
TimeDecodeCG	CheckinMethod CostCenter EmployeeFunctionMap EmployeeJobType ExpenseCategory FieldTypeOfTime FieldTypeOfTimeSub GlobalConfiguration InstallationEventStatus OptionalWorkCategory StoreChain StoreFunction TravelType	Scheduled refresh; Every 24 hours
TimeExpenseCG	Expense InstallationEvent	DCN

	MealAlert TimeTracking TravelTime	
FamilyFileCG	FamilyFile	DCN

1.5 Synchronization groups

The MBO package contains several synchronization groups in order to let the application synchronize only the data that is relevant at a particular time. The groups are organized in way to optimize transmission and communication with SUP server. The change detection interval is not relevant in the MBO package because this is Replication-Based System, there is no messaging in place that use the detection interval.

Synchronization group name	MBOs	Change detection interval
AuditSG	POSRAudit	10 minutes
BackorderSG	Backorder	10 minutes
ConfirmationSG	Confirmation	10 minutes
CreditSG	BillingLineSummary CustomerCreditDetail CustomerCreditHeader	10 minutes
CustomerAssignedSG	CustomerAssigned	10 minutes
CustomerPannedHoursSG	CustomerPlannedHours	10 minutes
CustomerProductSG	CustomerProduct	10 minutes
CustomerSelectionSG	AddRemoveAccountHistory CustomerSelection	10 minutes
CustomerSG	Customer	10 minutes
DecodeCreditSG	CustomerTerms CustomerTermsSeason MaterialNumberAlpha	10 minutes
DecodeSG	CreditCategory CreditCategoryTally CreditCategoryTypeTally CreditSummaryTally DecodeOrderStatus Format FormatSub OrderRecordInputType POSRR reorderStatus	10 minutes

	PrefixUPC ProductCollection ProductLine ProductType RejectType Season ShipmentStatus SizeDescription SpecialMarkdownType TechnologyFormat TransmissionType	
EmployeeSG	Employee	10 minutes
InstallationSG	EventRole InstallationChecklistReporting InstallationEventRole RevisionSystem	10 minutes
InventoryPocketSG	InventoryPocket	10 minutes
InventorySG	Inventory	10 minutes
OrderSG	Order OrderItem	10 minutes
PocketSG	AlternateInventoryLocation Pocket	10 minutes
ProductSG	Product	10 minutes
SBTDiscardSG	SBTDiscard SBTDiscardItem SBTNonReturnableUPC	10 minutes
SBTTransferSG	SBTTransferDetail SBTTransferHeader	10 minutes
ShipmentSG	ShipmentHeader ShipmentCarton ShipmentProduct TrackCarrierCode	10 minutes
StoreSG	ScheduledStoreEvent ScheduledStoreEventPPL	10 minutes
TimeDecodeSG	CheckinMethod CostCenter	10 minutes

	EmployeeFunctionMap EmployeeJobType ExpenseCategory FieldTypeOfTime FieldTypeOfTimeSub GlobalConfiguration InstallationEventStatus OptionalWorkCategory StoreChain StoreFunction TravelType	
TimeExpenseMileageSG	Expense InstallationEvent MealAlert TimeTracking TravelTime	10 minutes
FamilyFileSG	FamilyFile	10 minutes

1.6 Synchronization Parameters

This section lists the MBO that contains synchronization parameters that enables the middleware to send only the relevant rows to a device. The parameters are separated into two logical groups; the EnterpriseID based which means it is data relevant only to the user. The second group is Account based which means that data is relevant to a specific account. The information is presented based on the synchronization keys. All the MBO that shares the same keys are listed under the name of that key.

Synchronization Parameters	Affected MBO	Mapped Attributes
EnterpriseldSK	POSRAudit TimeTracking Expense TravelTime InstallationEvent MealAlert Employee Order OrderItem SBTDiscard	Enterprise_ID

	SBTDiscardItem SBTTransferHeader SBTTransferDetail CustomerCreditHeader CustomerCreditDetail BillingLineSummary InstallationChecklistReporting EventRole CustomerAssigned	
SoldToldSK	Customer CustomerProduct CustomerTerms CustomerTermsSeason InventoryPocket AlternateInventoryLocation Pocket Backorder Inventory ScheduledStoreEvent CustomerPlannedHours RevisionSystem ShipmentHeader	SoldTold

1.7 Operations

This section describes the operations that are implemented in the different MBO. The request and response row indicates the XSD file that is used to map the field to the Web Service corresponding URL.

CustomerProduct	
updateBackStockIndicator()	Update the backstock indicator for a product related to a customer
URL	/CustomerProduct/{id}
Request	UpdateBackStockIndicator.xsd.xml
Response	UpdateBackStockIndicator.xsd.xml
Mapping (MBO to Web Service)	BackStockIndicator – Back_Stock_Availability_Ind SoldTold – BU_Partner_Sold_To_Num SoldTold – id ProductId – Product_Id

POSRAudit	
SendAudit(POSRAuditItemStruct[])	Send a list of POSR Audit item
URL	/PosrAuditDtl
Request	AddPOSRAudit.xsd
Response	AddPOSRResponse.xsd.xml
Mapping	Structure POSRAuditItemStruct is used for the mapping. The structure is built using the POSRAuditItem Local MBO.

MealAlert	
create()	When a MealAlert is displayed on the device, this operation is called and sent to the backend.
URL	/MealAlert
Request	SubmitMealAlert.xsd.xml
Response	SubmitMealAlert.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

TravelTime	
create()	Create a mileage expense. The user input the data and it is sent to the backend at the end of a day.
URL	/TimeTrackingExpense
Request	TravelTimeCreate.xsd.xml
Response	TravelTimeCreate.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

InstallationEvent	
create()	Create a InstallationEvent record on the backend. This operation must follows a TimeTracking create operation. If the TimeTracking is not created before, the operation for InstallationEvent will fail on the backend
URL	/InstallationEvent
Request	InstallationEventCreate.xsd.xml
Response	InstallationEventCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

Expense	
create()	Create an Expense record on the backend. This operation must follows a TimeTracking create operation. If the TimeTracking is not created before, the operation for Expense will fail on the backend
URL	/ExpenseDtl
Request	ExpenseDtlCreate.xsd.xml
Response	ExpenseDtlCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

SBTDiscard	
create()	Create a header for a SBTDiscard. The header must be created prior to create SBTDiscardItem. The relationship between SBTDiscard and SBTDiscardItem is one-to-many.
URL	/SBTDiscardHeader
Request	SBTDiscardCreate.xsd.xml
Response	SBTDiscardResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

TimeTrackingExpense	
create()	Create a TimeTracking records. This operation is crucial and is used to calculate the time the user worked and determine his paycheck. The relationship between TimeTracking with Expense and InstallationEvent is one-to-many.
URL	/TimeTrackingExpense
Request	TimeTrackingExpenseCreate.xsd.xml
Response	TimeTrackingExpenseCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

SBTDiscardDetail	
create()	Create a SBTDiscardItem that represents a specific product to be discarded. This operation must follows a SBTDiscard create operation. If the SBTDiscard is not created before, the operation for SBTDiscardItem will fail on the backend
URL	/SBTDiscardDetail
Request	SBTDiscardItemCreate.xsd.xml
Response	SBTDiscardItemCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response.

EventRole	
create()	Creates an EventRole that represents what the user did for a particular event at a particular customer.
URL	/EventRole
Request	EventRoleCreate.xml.xsd
Response	No response
Mapping	All attributes are mapped for the Request. There is no response for this operation.

InstallationChecklistReporting	
create()	This operation creates an InstallationChecklistReporting record that represents the summary of an installation once it is done.
URL	/InstallationChecklistReporting
Request	InstallationCheckListReportingCreate.xsd.xml
Response	InstallationCheckListReportingResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

SBTTransferHeader	
create()	Create a header for a SBT Transfer. The header must be created prior to create SBTTransferDetail. The relationship between SBTTransferHeader and SBTTransferDetail is one-to-many.
URL	/SbtTransferHeader
Request	SbtTransferHeaderCreate.xsd.xml
Response	SbtTransferHeaderResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

SBTTransferDetail	
create()	Create a SBTTransferDetail that represents a specific product to be transfered. This operation must follows a SBTTransferHeader create operation. If the SBTTransferHeader is not created before, the operation for SBTTransferDetail will fail on the backend.
URL	/SbtTransferDetails
Request	SBTTransferDetailCreate.xsd.xml
Response	SbtTransferDetailCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

CustomerCreditHeader	
create()	Create a header for a CustomerCreditHeader. The header must be created prior to create CustomerCreditDetail and BillingLineSummary. The relationship between CustomerCreditHeader with CustomerCreditDetail and BillingLineSummary is one-to-many.
URL	/SoldToCustCreditHeader
Request	CustomerCreditCreate.xsd.xml
Response	CustomerCreditResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

CustomerCreditDetail	
create()	Create a CustomerCreditDetail that represents a specific product to be credited / calculated. This operation must follows a CustomerCreditHeader create operation. If the CustomerCreditHeader is not created before, the operation for CustomerCreditDetail will fail on the backend.
URL	/SoldToCustCreditDetail
Request	CustomerCreditDetailCreate.xsd.xml
Response	CustomerCreditDetailCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

BillingLineSummary	
create()	Create a BillingLineSummary that represents a specific product to be credited / calculated. This operation must follow a CustomerCreditHeader create operation. If the CustomerCreditHeader is not created before, the operation for BillingLineSummary will fail on the backend.
URL	/BillingLineSummary
Request	BillingLineSummaryCreate.xsd.xml
Response	BillingLineSummaryCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

Order	
create()	Create a header for an Order. The header must be created prior to create OrderItem. The relationship between Order with OrderItem is one-to-many.
URL	/OrderHeader
Request	OrderCreate.xsd.xml
Response	OrderCreateResponse.xsd.xml
Mapping	All attributes are mapped for the Request and Response

AddRemoveAccountHistory	
create()	Create an entry in the CRM system every time a user Add or Remove an account from the device.
RFC	ZSOE1_FM_ADD_ACCOUNT_IN
Input	IT_ACCOUNTS_IN
Output	None
Mapping	All attributes are mapped for the Input

1.8 Queries

This section describes all the custom queries that are implemented in the different MBO. Each table represents a specific MBO and the associated queries. The default queries that are automatically generated are omitted and only the queries that were written using the wizard from the workspace are represented

Customer		
Query name	Parameter	Definition
FindByLegacyNumber <i>Returns single object</i>	legacy – String(10)	SELECT x.* FROM Customer x WHERE x.LegacyAccount = :legacy

Product		
Query name	Parameter	Definition
findByStockNumber	stocknumber – String(18)	SELECT x.* FROM Product x WHERE x.Stock_Number = :stocknumber
findMultipleByUPC	upc – String(300)	SELECT x.* FROM Product x WHERE x.Product_UPC LIKE '%' :upc '%'
findMultipleByStockNumber	stock – String(18)	SELECT x.* FROM Product x WHERE x.Stock_Number LIKE '%' :stock '%'
FindByProductUPC	produpc – String(300)	SELECT x.* FROM Product x WHERE x.Product_UPC = :produpc
findMultipleByFormat	format_cd - Short	SELECT x.* FROM Product x WHERE x.Format_Desc_Cd = :format_cd
findMultipleByPattern	prod_cool_cd - Short	SELECT x.* FROM Product x WHERE x.Product_Collection_Cd = :prod_coll_cd
FindByProductUPAndSuppl	upc – String(20) supplemental – String(30)	SELECT x.* FROM Product x WHERE x.Product_UPC = :upc AND x.Product_UPC_Suppl = :supplemental
findMultipleByFormatAndSoldTold	Format_cd – Short SoldTold – String(20)	SELECT x.* FROM Product x, CustomerProduct cp WHERE x.Format_Desc_Cd = :format_cd AND cp.SoldTold = :soldtold AND cp.ProductId = x.ProductId
findMultipleByPatternAndSoldTold	Prod_coll_cd – Short SoldTold – String(20)	SELECT x.* FROM Product x, CustomerProduct cp WHERE x.Product_Collection_Cd = :prod_coll_cd AND cp.SoldTold = :soldtold AND cp.ProductId = x.ProductId
FindByProductUPCLong	Upc - Long	SELECT x.* FROM Product x WHERE x.Product_UPC_Long = :upc
FindByUPCAndSupplLong	Upc – Long Supplemental – String(30)	SELECT x.* FROM Product x WHERE x.Product_UPC_Long = :upc AND x.Product_UPC_Suppl = :supplemental
FindMultipleByPartialUPC	Upc - Long	SELECT x.* FROM Product x WHERE x.Product_UPC_Long = :upc

POSRAudit		
Query name	Parameter	Definition
FindMultiplePendingByTimespan	Start – Timestamp End – Timestamp pendingState - Boolean	SELECT x.* FROM POSRAudit x WHERE x.TransmissionTimestamp >= :start AND x.TransmissionTimestamp <= :end AND x.pending = :pendingState
FindMultiplePendingBySoldTold	soldTold – String(10) pendingState - Boolean	SELECT x.* FROM POSRAudit x WHERE x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultiplePendingBySoldToAndTimespan	Start – Timestamp End – Timestamp SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM POSRAudit x WHERE x.TransmissionTimestamp >= :start AND x.TransmissionTimestamp <= :end AND x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultipleByTimespan	Start – Timestamp End – Timestamp	SELECT x.* FROM POSRAudit x WHERE x.TransmissionTimestamp >= :start AND x.TransmissionTimestamp <= :end

AlternatInventoryLocation		
Query name	Parameter	Definition
FindByProductAndSoldTold	Product – String(18) Customer – String(300)	SELECT x.* FROM AlternatInventoryLocation x WHERE x.ProductId = :product AND x.SoldTold = :customer

Pocket		
Query name	Parameter	Definition
FindByPocketId	Alpha – String(10) File_pocket – String(30) Double_Design – String(4) SoldTold – String(10)	SELECT x.* FROM Pocket x WHERE x.Rack_Alpha_Cd = :alpha AND x.File_Pocket_Cd = :file_pocket AND x.Double_Design_Sfx = :double_design
FindBySoldTold	soldTold – String(10)	SELECT x.* FROM Pocket x WHERE x.SoldTold = :soldTold ORDER BY x.Rack_Alpha_Cd, x.File_Pocket_Cd, x.Double_Design_Sfx
FindByProductId	Product – String(18)	SELECT x.* FROM Pocket x WHERE x.ProductId = :product

FindByProductAndSoldTold	Product – String(18) SoldTold – String(10)	SELECT x.* FROM Pocket x WHERE x.ProductId = :product AND x.SoldTold = :soldTold
FindMultipleByPocketIdAndSoldTold	Rack_alpha – String(10) File_pocket – Short Double_Design – String(4) SoldTold – String(10)	SELECT x.* FROM Pocket x WHERE x.Rack_Alpha_Cd = :rack_alpha AND x.File_Pocket_Cd = :file_pocket AND x.Double_Design_Sfx = :double_design AND x.SoldTold = :soldTold

POSRAuditItem		
Query name	Parameter	Definition
FindBySoldToldAndProductId	SoldTold – String(20) productId – String(20)	SELECT x.* FROM POSRAuditItem x WHERE x.SoldTold = :soldTold AND x.ProductId = :productId
FindMultipleByStatusAndSoldTold	SoldTold – String(20) Status – String(1)	SELECT x.* FROM POSRAuditItem x WHERE x.SoldTold = :soldTold AND x.Status = :status
FindMultipleByStatus	Status – String(1)	SELECT x.* FROM POSRAuditItem x WHERE x.Status = :status
FindMultipleByParentPK	soldTold – String(10) enterpriseID – String(20) timestamp – String(50)	SELECT x.* FROM POSRAuditItem x WHERE x.SoldTold = :soldTold AND x.EnterpriseID = :enterpriseID AND x.Timestamp = :timestamp

InventoryPocket		
Query name	Parameter	Definition
FindBySoldToldAndProductId	soldTold – String(300) productId – String(300)	SELECT x.* FROM InventoryPocket x WHERE x.SoldTold = :soldTold AND x.ProductId = :productId
FindMultipleBySoldToldAndProduct	soldTold – String(300) productId – String(300)	SELECT x.* FROM InventoryPocket x WHERE x.SoldTold = :soldTold AND x.ProductId = :productId
FindMultipleByReorderTicket	Reorder_ticket – String(10)	SELECT x.* FROM InventoryPocket x WHERE x.eT_Item_Num = :reorder_ticket
FindMultipleByGroupAndProductAndSoldTold	Fg1_group_cd – String(2) Fg1_product_cd – String(2)	SELECT x.* FROM InventoryPocket x WHERE x.FG1_Group_Cd = :fg1_group_cd

	SoldTold – String(10)	AND x.FG1_Product_Cd = :fg1_product_cd AND x.SoldTold = :SoldTold
FindMultipleByGroupProductDisplayAndSoldTold	Display – String(1) Fg1_group – String(2) Fg1_product – String(2) soldTold – String(10)	SELECT x.* FROM InventoryPocket x WHERE x.mC_Dspl_Typ_Cd = :display AND x.FG1_Group_Cd = :fg1_group AND x.FG1_Product_Cd = :fg1_product AND x.SoldTold = :soldTold

TravelTime		
Query name	Parameter	Definition
findMultipleByTimespan	Start_time – DateTime End_time – DateTime	SELECT x.* FROM TravelTime x WHERE x.Start_Timestamp >= :start_time AND x.Start_Timestamp <= :end_time
FindOverlap	Start_time – DateTime End_time – DateTime	SELECT x.* FROM TravelTime x WHERE (:start >= x.Start_Timestamp AND :start <= x.End_Timestamp) OR (:end >= x.Start_Timestamp AND :end <= x.End_Timestamp)
FindMultipleByTimespanDescending	Start_time – DateTime End_time – DateTime	SELECT x.* FROM TravelTime x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end Order By x.Start_Timestamp DESC
FindMultipleByTimespanAndTravelCode	Start_time – DateTime End_time – DateTime Travel_cd – String(1)	SELECT x.* FROM TravelTime x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end AND x.Travel_Type_Cd = :travel_cd
FindMultipleByTimespanAscending	Start_time – DateTime End_time – DateTime	SELECT x.* FROM TravelTime x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end Order By x.Start_Timestamp ASC

CustomerSelection		
Query name	Parameter	Definition
FindByAccountNumber	Account_number – String(10)	SELECT x.* FROM CustomerSelection x WHERE x.LegacyAccount = :account_number
FindMultipleByNames	Name1 – String(40) Name2 – String(40)	SELECT x.* FROM CustomerSelection x WHERE x.Name1 LIKE '%' :name1 '%'

		OR x.Name2 LIKE '%' :name2 '%'
FindMultipleByCityOnly	City – String(40)	SELECT x.* FROM CustomerSelection x WHERE x.City = :city
FindMultipleByCityAndState	City – String(40) State – String(3)	SELECT x.* FROM CustomerSelection x WHERE x.City = :city AND x.Region = :state
FindBySoldTold	SoldTold – String(10)	SELECT x.* FROM CustomerSelection x WHERE x.SoldTold = :soldTold

PrefixUPC		
Query name	Parameter	Definition
FindByManufacturerCode	Manufacturer – String(10)	SELECT x.* FROM PrefixUPC x WHERE x.Manufacturer_Cd = :manufacturer

InstallationEvent		
Query name	Parameter	Definition
FindByParentPK	Start – DateTime enterpriseld – String(10) field_type_time – String(1) field_type_sub – String(1)	SELECT x.* FROM InstallationEvent x WHERE x.Start_Timestamp = :start_timestamp AND x.EnterpriseID = :enterpriseld AND x.Field_Type_Of_Time_Cd = :field_type_of_time AND x.Field_Type_Of_Time_Sub1_Cd = :field_type_of_sub1

Expense		
Query name	Parameter	Definition
FindMultipleByTimespan	Start – DateTime End – DateTime	SELECT x.* FROM Expense x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end
FindMultipleByTimespanDescending	Start – DateTime End – DateTime	SELECT x.* FROM Expense x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end Order By x.Start_Timestamp DESC
FindByParentPK	Start – DateTime enterpriseld – String(10) field_type_time – String(1) field_type_sub – String(1)	SELECT x.* FROM Expense x WHERE x.Start_Timestamp = :start_timestamp AND x.EnterpriseID = :enterpriseld

		AND x.Field_Type_Of_Time_Cd = :field_type_of_time AND x.Field_Type_Of_Time_Sub1_Cd = :field_type_of_sub1
--	--	---

SBTDiscard		
Query name	Parameter	Definition
FindMultiplePendingByTimespan	Start – DateTime End – DateTime pendingState - Boolean	SELECT x.* FROM SBTDiscard x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.pending = :pendingState
FindMultiplePendingBySoldTold	SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM SBTDiscard x WHERE x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultiplePendingBySoldToAndTimespan	Start – DateTime End – DateTime SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM SBTDiscard x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultipleByTimespan	Start – DateTime End – DateTime	SELECT x.* FROM SBTDiscard x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end

TimeTracking		
Query name	Parameter	Definition
FindMultipleByTimespan	Start – DateTime End – DateTime	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end
FindOverlap	Start – DateTime End – DateTime	SELECT x.* FROM TimeTracking x WHERE (:start >= x.Start_Timestamp AND :start <= x.End_Timestamp) OR (:end >= x.Start_Timestamp AND :end <= x.End_Timestamp) AND x.Field_Type_Of_Time_Sub1_Cd != '4'
FindMultipleByTimespanAndTimeCode	Start – DateTime End – DateTime	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start

	Field_sub1 – String(1)	AND x.Start_Timestamp <= :end AND x.Field_Type_Of_Time_Sub1_Cd = :field_sub1
FindMultipleByTimespanForPaidTime	Start – DateTime End – DateTime	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end AND (x.Field_Type_Of_Time_Sub1_Cd = '1' OR x.Field_Type_Of_Time_Sub1_Cd = '3')
FindMultipleByTimespanAndConfirmation	Start – DateTime End – DateTime Confirmation – String(40)	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end AND x.ConfirmationNumber = :confirmation
FindMultipleByTimespanAndTimeCodeAscending	Start – DateTime End – DateTime Sub1_code – String(1)	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end AND x.Field_Type_Of_Time_Sub1_Cd = :sub1_code Order By x.Start_Timestamp ASC
FindMultipleByTimespanAndExpense	Start – DateTime End – DateTime	SELECT x.* FROM TimeTracking x WHERE x.Start_Timestamp >= :start AND x.Start_Timestamp <= :end AND x.Field_Type_Of_Time_Sub1_Cd = '4'

SBTDiscardItem		
Query name	Parameter	Definition
FindByParentPK	Timestamp – DateTime Enterpriseld – String(10) SoldTold – String(10)	SELECT x.* FROM SBTDiscardItem x WHERE x.Transaction_Timestamp = :Timestamp AND x.Enterprise_ID = :enterpriseID AND x.SoldTold = :SoldTold

EventRole		
Query name	Parameter	Definition
FindMultipleBySoldToldAndEventId	SoldTold – String(300) eventId - Long	SELECT x.* FROM EventRole x WHERE x.SoldTold = :soldTold AND x.Event_Id = :eventId

InstallationChecklistReporting		
Query name	Parameter	Definition
FindMultiplePending	pendingState - Boolean	SELECT x.* FROM InstallationChecklistReporting x WHERE x.pending = :pendingState
FindMultiplePendingBySoldTold	SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM InstallationChecklistReporting x WHERE x.SoldTold = :soldTold AND x.pending = :pendingState

CustomerPlannedHours		
Query name	Parameter	Definition
FindMultipleBySoldToldOrderedASC	soldTold – String(300)	SELECT x.* FROM CustomerPlannedHours x WHERE x.SoldTold = :soldTold ORDER BY x.Planned_Week_Begin_Dt ASC

SBTTTransferHeader		
Query name	Parameter	Definition
FindMultiplePendingByTimespan	Start – DateTime End – DateTime pendingState - Boolean	SELECT x.* FROM SBTTTransferHeader x WHERE x.Entry_Timestamp >= :start AND x.Entry_Timestamp <= :end AND x.pending = :pendingState
FindMultiplePendingBySoldTold	SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM SBTTTransferHeader x WHERE x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultiplePendingBySoldToAndTimespan	Start – DateTime End – DateTime SoldTold – String(10) pendingState - Boolean	SELECT x.* FROM SBTTTransferHeader x WHERE x.Entry_Timestamp >= :start AND x.Entry_Timestamp <= :end AND x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultipleByTimespan	Start – DateTime End – DateTime	SELECT x.* FROM SBTTTransferHeader x WHERE x.Entry_Timestamp >= :start AND x.Entry_Timestamp <= :end

SBTTransferDetail		
Query name	Parameter	Definition
FindByParentPK	Entry_timestamp – DateTime enterpriseld – String(10) soldTold – String(10) xfer_direction – String(1)	SELECT x.* FROM SBTTransferDetail x WHERE x.Entry_Timestamp = :entry_timestamp AND x.Enterprise_ID = :enterpriseld AND x.SoldTold = :soldTold AND x.Xfer_Direction = :xfer_direction

CustomerCreditHeader		
Query name	Parameter	Definition
findMultipleByPrimaryKeyAndTimespan	SoldTold – String(300) Credit_header – String(300) Start – DateTime End – DateTime Credit_tally – String(300)	SELECT x.* FROM CustomerCreditHeader x WHERE x.SoldTold = :soldTold AND x.Credit_Header_Season_Type = :credit_header_season_type AND x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.Credit_Or_Tally_Cd = :credit_or_tally
FindByPrimaryKeyAndCreditTally	SoldTold – String(300) timestamp - DateTime Credit_header – String(300) enterpriseld – String(300) Credit_tally – String(300)	SELECT x.* FROM CustomerCreditHeader x WHERE x.SoldTold = :soldTold AND x.Transaction_Timestamp = :transaction_timestamp AND x.Credit_Header_Season_Type = :credit_season_header_type AND x.Enterprise_ID = :Enterpriseld AND x.Credit_Or_Tally_Cd = :credit_tally_ind
FindMultipleBySoldToldAndEntldAndCreditIndAndCreditType	enterpriseld – String(10) soldTold – String(10) credit_tally_cd – String(1) credit_header_cd – String(1)	SELECT x.* FROM CustomerCreditHeader x WHERE x.Enterprise_ID = :enterpriseld AND x.SoldTold = :soldTold AND x.Credit_Or_Tally_Cd = :credit_tally_cd AND x.Credit_Header_Season_Type = :credit_header_cd
FindMultiplePendingBySoldTold	soldTold – String(10) everyday_season – String(1)	SELECT x.* FROM CustomerCreditHeader x WHERE x.SoldTold = :soldTold

	credit_tally – String(1) pendingState - Boolean	AND x.Credit_Header_Season_Type = :everyday_season AND x.Credit_Or_Tally_Cd = :credit_tally AND x.pending = :pendingState
FindMultiplePendingBySoldToldAndTimespan	Start – DateTime End – DateTime soldTold – String(10) everyday_season – String(1) credit_tally – String(1) pendingState - Boolean	SELECT x.* FROM CustomerCreditHeader x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.SoldTold = :soldTold AND x.Credit_Header_Season_Type = :everyday_season AND x.Credit_Or_Tally_Cd = :credit_tally AND x.pending = :pendingState
FindMultipleByTimespan	Start – DateTime End - DateTime	SELECT x.* FROM CustomerCreditHeader x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end

CustomerCreditDetail		
Query name	Parameter	Definition
FindMultipleByProductId	SoldTold – String(300) Credit_header – String(300) Start – DateTime End – DateTime ProductId - String(20)	SELECT x.* FROM CustomerCreditDetail x WHERE x.SoldTold = :SoldTold AND x.Credit_Header_Season_Type = :Credit_Header_Season_Type AND x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.Product_Id = :ProductId
FindByParentPK	Timestamp – DateTime enterpriseld – String(10) soldTold – String(10) credit_header – String(1)	SELECT x.* FROM CustomerCreditDetail x WHERE x.Transaction_Timestamp = :transaction_timestamp AND x.Enterprise_ID = :enterpriseld AND x.SoldTold = :soldTold AND x.Credit_Header_Season_Type = :credit_header_type

BillingLineSummary		
Query name	Parameter	Definition
FindByParentPK	Entry_timestamp – DateTime enterpriseld – String(10) soldTold – String(10) credit_header – String(1)	SELECT x.* FROM BillingLineSummary x WHERE x.Transaction_Timestamp = :transaction_timestamp AND x.Enterprise_ID = :enterpriseID AND x.SoldTold = :soldtold AND x.Credit_Header_Season_Type = :credit_header_season

RevisionSystem		
Query name	Parameter	Definition
FindByInventoryPocketData	Fg1_group – String(2) Fg1_product- String(2) Op_proc – String(2) soldTold – String(10) revision_system – String(2)	SELECT x.* FROM RevisionSystem x WHERE x.FG1_Group_Cd = :Fg1_group AND x.FG1_Product_Cd = :fg1_product AND x.Op_Proc_Corp_Num = :op_proc AND x.SoldTold = :soldTold AND x.Revision_System_CD = :revision_system_cd
FindMultipleByGroupandProductAndSoldTold	Fg1_group – String(2) Fg1_product- String(2) soldTold – String(10)	SELECT x.* FROM RevisionSystem x WHERE x.FG1_Group_Cd = :FG1_Group AND x.FG1_Product_Cd = :FG1_Product AND x.SoldTold = :SoldTold
FindMultipleBySoldToldAndDescription	SoldTold – String(10) Description – String(40)	SELECT x.* FROM RevisionSystem x WHERE x.SoldTold = :soldtold AND x.Revision_system_desc = :description
FindMultipleBySoldToldAndStatus	Status – String(30) soldTold – String(10)	SELECT x.* FROM RevisionSystem x WHERE x.Revision_System_Status = :status AND x.SoldTold = :soldtold

Order		
Query name	Parameter	Definition
FindMultipleBySoldToldAndPONumber	soldTold – String(10) po_number – String(10)	SELECT x.* FROM Order x WHERE x.SoldTold = :soldTold AND x.Purchase_Order_Num = :po_number
FindMultiplePendingByTimespan	Start – DateTime End – DateTime pendingState - Boolean	SELECT x.* FROM Order x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.pending = :pendingState

FindMultiplePendingBySoldTold	SoldTold – String(10) pendingState -Boolean	SELECT x.* FROM Order x WHERE x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultiplePendingBySoldToAndTimespan	Start – DateTime End – DateTime soldTold – String(10) pendingState - Boolean	SELECT x.* FROM Order x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end AND x.SoldTold = :soldTold AND x.pending = :pendingState
FindMultipleByTimespan	Start – DateTime End - DateTime	SELECT x.* FROM Order x WHERE x.Transaction_Timestamp >= :start AND x.Transaction_Timestamp <= :end

OrderItem		
Query name	Parameter	Definition
FindMultipleBySoldToldAndProductId	soldTold – String(10) productId – String(12)	SELECT x.* FROM OrderItem x WHERE x.SoldTold = :soldtold AND x.Product_Id = :productId
FindMultipleByReorderTicket	reorderTicket – String(12)	SELECT x.* FROM OrderItem x WHERE x.Reorder_Ticket_Num = :reorderTicket
FindByParentPK	SoldTold – String(10) Transaction – DateTime enterpriseld – String(10)	SELECT x.* FROM OrderItem x WHERE x.SoldTold = :soldTold AND x.Transaction_Timestamp = :transaction_timestamp AND x.Enterprise_ID = :enterpriseld
FindByParentPKAndProductId	SoldTold – String(10) Transaction – DateTime enterpriseld – String(10) productId – String(12)	SELECT x.* FROM OrderItem x WHERE x.SoldTold = :soldTold AND x.Transaction_Timestamp = :transaction_timestamp AND x.Enterprise_ID = :enterpriseld AND x.Product_Id = :ProductId

InstallationUnassignedProductLocal		
Query name	Parameter	Definition
findMultipleBySoldToldAndEventIdAndType	EventId – Long Product_type – Int SoldTold – String(10)	SELECT x.* FROM InstallationUnassignedProductLocal x WHERE x.EventId = :EventId AND x.ProductType = :product_type AND x.SoldTold = :SoldTold
findMultipleBySoldToldAndEventIdAndProductId	SoldTold – String(10) eventId – Long ProductId – String(12)	SELECT x.* FROM InstallationUnassignedProductLocal x WHERE x.SoldTold = :soldTold AND x.EventId = :eventId AND x.ProductId = :ProductId

ScheduleStoreEventLocal		
Query name	Parameter	Definition
FindMultipleBySoldToAndNewEvent	SoldTold – String(10) newEvent - Boolean	SELECT x.* FROM ScheduleStoreEventLocal x WHERE x.SoldTold = :soldTold AND x.IsNewEvent = :newEvent

ShipmentHeader		
Query name	Parameter	Definition
FindMultipleByArrivalTimespan	Start_timestamp – Date End_timestamp – Date SoldTold – String(10)	SELECT x.* FROM ShipmentHeader x WHERE x.Estimated_Arrival_Date >= :start_timestamp AND x.Estimated_Arrival_Date <= :end_timestamp AND x.SoldTold = :soldTold
FindMultipleByShippedTimespan	Start_timestamp – Date End_timestamp – Date SoldTold – String(10)	SELECT x.* FROM ShipmentHeader x WHERE x.Date_Shipped >= :start_timestamp AND x.Date_Shipped <= :end_timestamp AND x.SoldTold = :soldTold
findByShipNum	Ship_Num – String(9)	SELECT x.* FROM ShipmentHeader x WHERE x.Ship_Num = :ship_num

ShipmentCarton		
Query name	Parameter	Definition
FindMultipleBySoldToldAndShipNum	SoldTold – String(10) Ship_Num – String(9)	SELECT x.* FROM ShipmentCarton x WHERE x.SoldTold = :Soldtold AND x.Ship_Num = :Ship_Num
FindMultipleBySoldToAndTimespan	Soldtold – String(10) Start – Date End - Date	SELECT x.* FROM ShipmentCarton x WHERE x.SoldTold = :soldTold AND x.Arrival_Date >= :start AND x.Arrival_Date <= :end
findByShipNum	Ship_Num – String(9)	SELECT x.* FROM ShipmentCarton x WHERE x.Ship_Num = :shipNum

ShipmentProduct		
Query name	Parameter	Definition
FindMultipleBySoldToAndProductId	SoldTold – String(10) ProductId – String(18)	SELECT x.* FROM ShipmentProduct x WHERE x.SoldTold = :soldTold AND x.ProductId = :productId

1.9 Importing the MBO model into Mobile Workspace

Follow this procedure to import the MBO project into the Mobile Workspace. This procedure applies for any version of the Workspace.

1. Copy the XSD files to this folder (create it if it does not exist):
C:\Users\C5178588\Desktop\HALLMARK\web_services (this is a known issue with the Workspace; it does not let the XSD files to be relative to the project. The reference is the complete path. Make sure you create this exact directory)
2. Copy the project files from this location on Perforce:
\onDevice\Hallmark_Dev\dev\Hallmark_Model_Xlink\src\MBOModel\FSOHallmark (this folder contains the sup.model and sup.model_diagram, which are the required files)
3. Start the Mobile Workspace.
4. Under the Enterprise Explorer section, right click on REST Web Service → New...
5. Under the Name section, enter “webservice” (this is very important to have this name because all SUP configurations rely on this name) and click Next.
6. Enter the information required the wizard. Refer to the technical documentation to know what data has to be entered. Click Finish when done.
7. Under the Enterprise Explorer section, right click on SAP Server → New...
8. Under the Name section, enter “crm” (this is very important to have this name because all SUP configurations rely on this name) and click Next.
9. Enter the information required by the wizard. Refer to the technical documentation to know what data has to be entered. Click Finish when done.
10. Click on File → Import.
11. Choose the category General → Existing project into Workspace and click on Next.
12. Choose the root directory that contains the sup.model and sup.model_diagram files. The rest of the information will be filled automatically by the wizard. Once this is completed, click on Finish
13. The project “hallmark_workshop” should be available in the Project list in the Workspace.
14. Right-click on “hallmark_workshop” and select Open Diagram Editor.
15. The MBO model should be visible and be editable.

