Oracle Utilities Customer Self Service

Implementation Guide

Release 2.2.0.0

E78219-02

August 2016 (Revised November 2018)

ORACLE

Oracle Utilities Customer Self Service Implementation Guide

Release 2.2.0.0

E78219-02

August 2016

Table of Contents

Chapter 1

Product Overview

Oracle Utilities Customer Self Service is a flexible and user-friendly packaged utility portal that is pre-integrated with Oracle Utilities applications. This solution provides consumers with the ability to manage their accounts, take control of their consumption, and receive alerts and updates. It increases utility efficiency by facilitating interaction with consumers and highlighting incentives to optimize energy usage and reduce costs.

The application can provide both unsecured public access for finding general information and utility offerings, and secured access for registered and enrolled users to perform account specific operations.

Functional Overview

Oracle Utilities Customer Self Service modules include the following functionality:

- Account Management Module:
  - User registration
  - Password and User ID management
  - Self-service information management
  - Account information management
  - Alerts and notifications
  - Forms Management
- Billing and Payment Management Module:
  - Billing notification preferences
  - Account charges summary
  - View bill/payment history

  - Service charges to-date
  - Compare rate plans and analysis
  - Setup electronic billing
  - One-time payments
  - Schedule Payment
  - Automatic recurring payments
  - View rate plans and products
  - View promotions
  - Payment Arrangement
  - Budget Management and Billing
  - Prepaid Customer Enhancements
- Customer Service Management Module:
  - Add scalar meter read data
  - Detailed service usage
  - Download Usage Data (Usage Download)
  - Start, Stop, or Transfer Service for a new or existing customer
- Outage Module:
  - Outage Table - Display outage information for the utility as text.
  - Outage Map - Display a geographic map showing outage information for the utility.
  - My Outage Details - To show the current outages and planned outages for a given account
  - Report Public Outage - To report an outage for a public location
  - Report Premise Outage - To report an outage at a customer's premise for a given account.
- Commercial Account Management
  - Manage Sets
  - Multiple Account Data Download
  - Multiple Account Financial History
  - Multiple Account Aggregation
  - Multiple Account Usage Comparison

Two additional secured areas are available to provide the following capabilities:

- Administration
  - View and manage metadata used by the application (labels, messages, other entities)
  - View and manage access roles and security rules
- Customer support
  - Allow a CSR login and view core modules as selected customer

In addition, the system provides a web service to enroll multiple users to a set of Accounts.

Accessibility Support

Oracle Utilities Customer Self Service modules include access key (hot key) support. An access key is indicated by an underlined letter of a button, link, or field in a page, and typing the "shift+alt+<underlined letter>" key combination provides the user with access to the associated field, link, or button functionality.

Technical Overview

Oracle Utilities Customer Self Service is based on service oriented architecture and leverages industry leading Oracle application development technology.

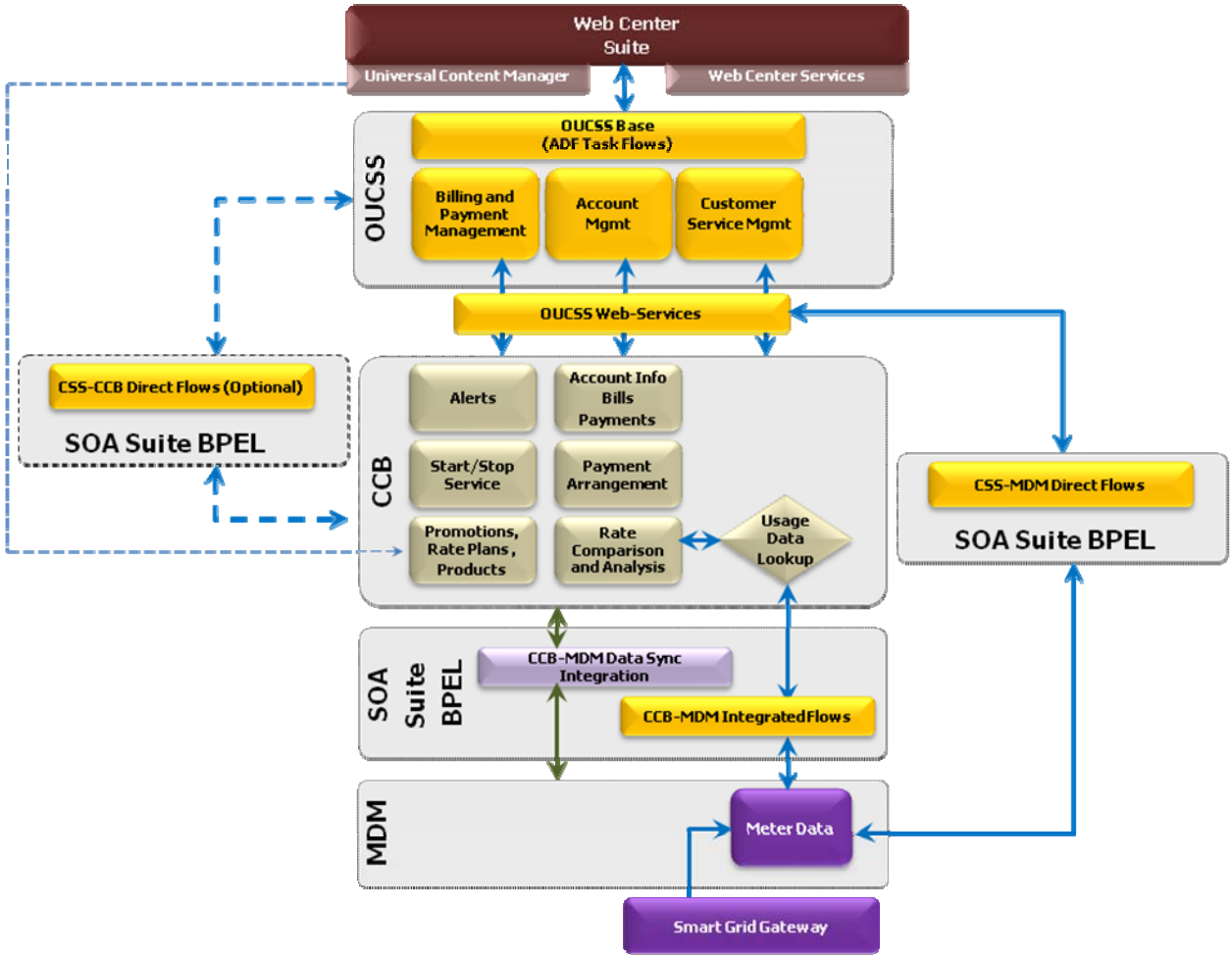
- Portal/Taskflow components are developed using Oracle Application Development Framework (ADF) 12*c* and are packaged as ADF shared libraries.
- Taskflows/Portlets are pre-integrated with Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management, and Oracle Utilities Network Management System applications using a standards-based web service API and Oracle SOA Suite.
- OUCSS taskflows are consumed directly in WebCenter Portal 12c for the portal solution. Please refer https://docs.oracle.com/middleware/1221/wcp/develop/GUID-796D444D-9613-408A-A3FE-D73B066113C3.htm#JPSDG15371 for more deailts on extending WebCenter Portal 12c.
    - OUCSS solution provides Portals with pages containing OUCSS taskflows with the release package for quick start purposes.
    - Implementations can extend and customize the provided solution by following the procedures described in the "Customizing and Extending the OUCSS Custom Portal" whitepaper (available on the Oracle Technology Network (OTN) web site (http://docs.oracle.com/cd/E72219_01/documentation.html, then choose the Oracle Utilities Customer Self Service link).

Security

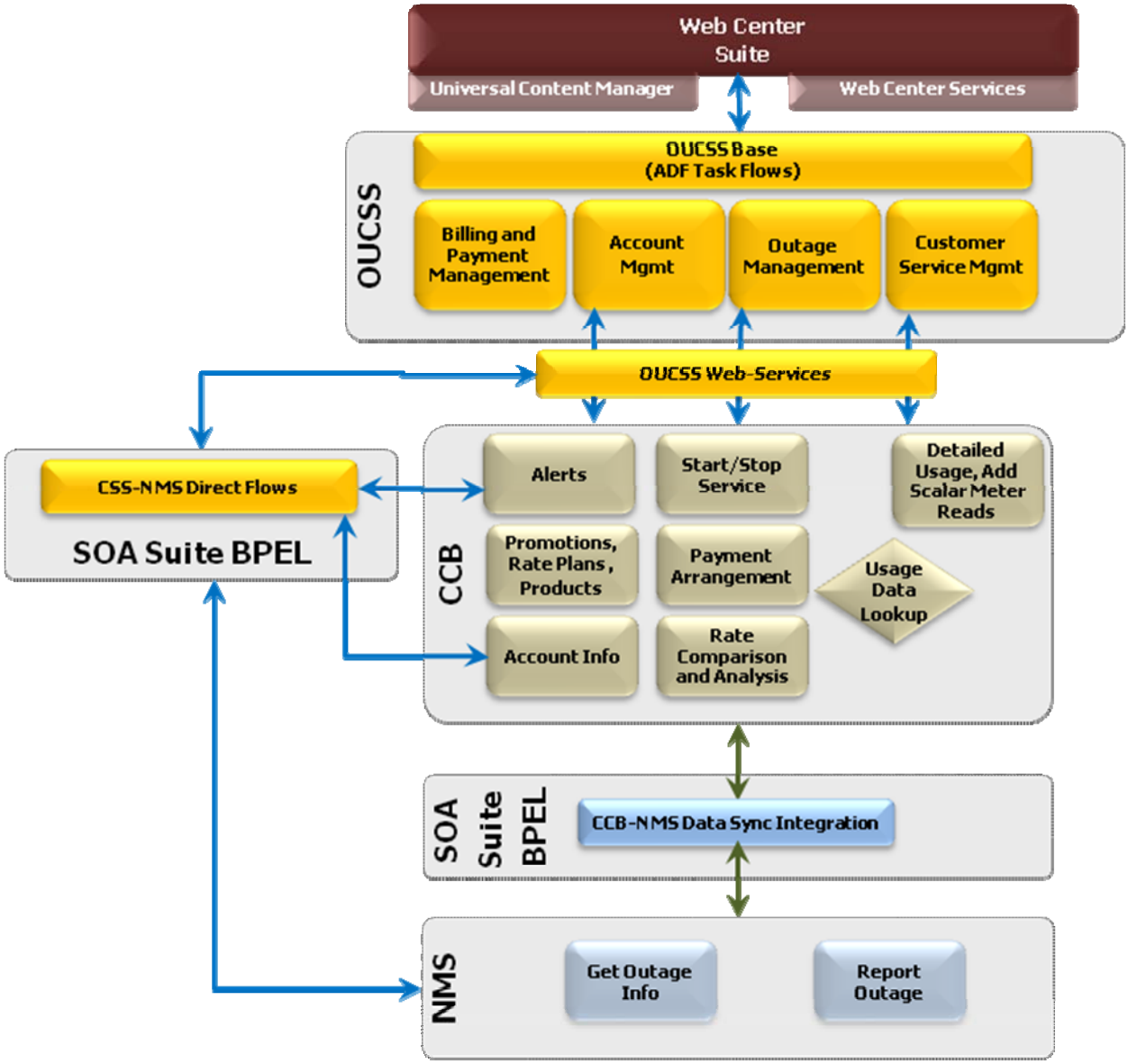
OUCSS offers Tier-1 and Tier-2 security.

- Tier-1 Security: OUCSS installs three Portals. All pages in the Self Service portal are public accessible. All pages in OUCSS Portal are secured and are accessed by authenticated users. The OUSSS Admin Portal is accessible only by users from the WSSAdminGroup or OUCSSCSRGroup enterprise groups. For more information on this type of security see the *Oracle Utilities Customer Self Service Security Guide*.
- Tier-2 security controls actions and fields on taskflows/portlets. For more information on this type of security, see the "Verify the Security Configuration" section in the *Oracle Utilities Customer Self Service Installation Guide*.

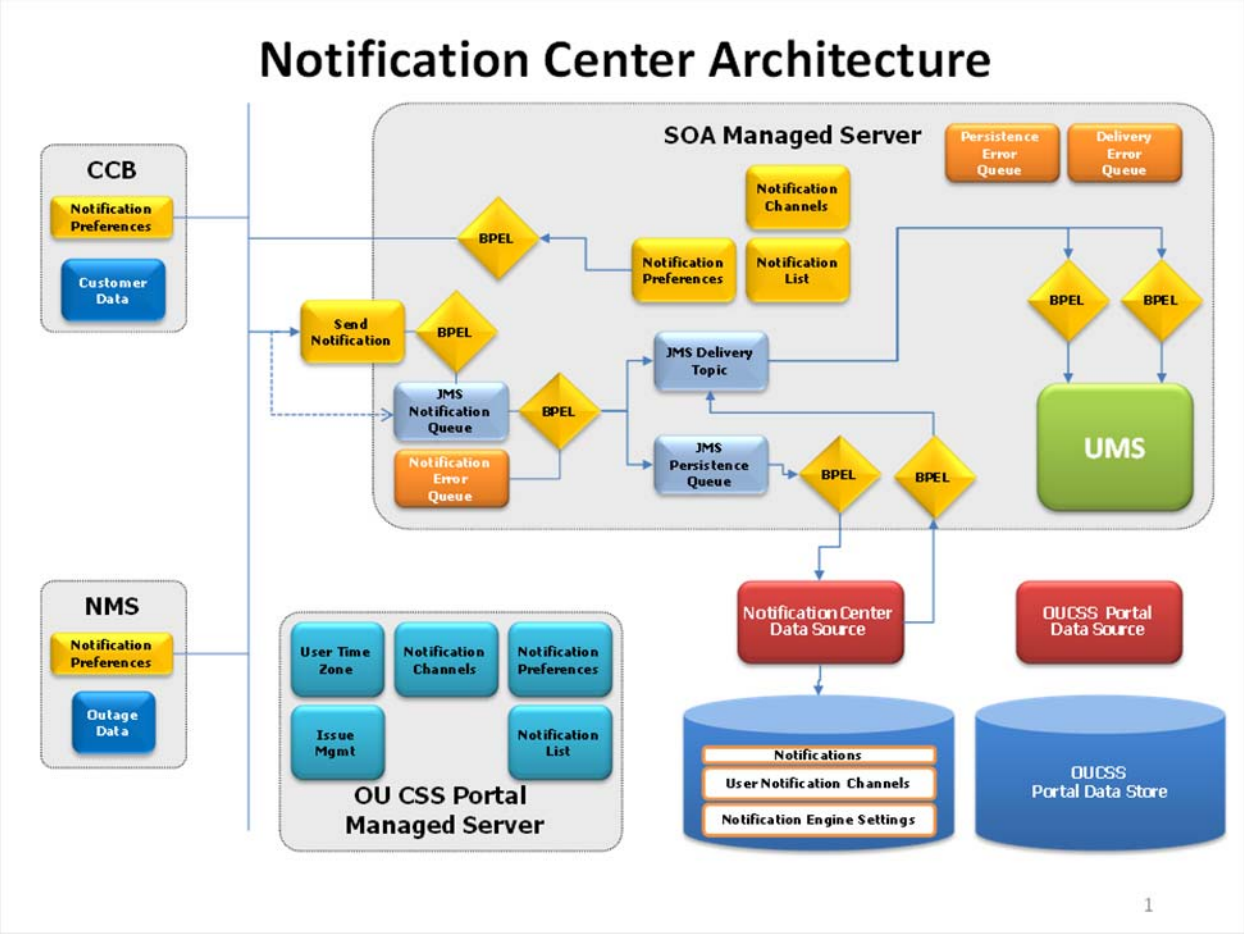
OUCSS Architecture



OUCSS Architecture diagram with CSS-MDM direct flows



OUCSS Architecture diagram with CSS-NMS direct flows



OUCSS Architecture diagram of Notification module

Additional Resources

Resource	Location
WebCenter Custom Portal Application Developer's Guide	http://docs.oracle.com/cd/E29542_01/webcenter.1111/e63259/toc.htm
Oracle Fusion Developers Guide (JDeveloper and ADF)	http://docs.oracle.com/cd/E41362_01/web.1111/b31974/toc.htm
Customizing Taskflows : Oracle WebCenter Spaces	http://docs.oracle.com/cd/E28280_01/webcenter.1111/e10148/jpsdg_taskflows.htm
Customize and Extend the OUCSS Portal Customizing and Extending the OUCSS Custom Portal Whitepaper	Available for download in the Oracle Utilities Customer Self Service section of the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://docs.oracle.com/cd/E72219_01/documentation.html).

Note: This document and the documentation mentioned above are subject to revision and updating. For the most recent version of this and related documentation, as well as information on functionality and known issues for other Oracle products that may be required for installation and proper functionality of this product, check the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://docs.oracle.com/cd/E72219_01/documentation.html, then choose the Oracle Utilities Customer Self Service link).

Chapter 2

OUCSS Implementation

OUCSS Web Services

The following are the base services invoked by Oracle Utilities Customer Self Service.

Notes:

- For more information on configuring CCB services, see Chapter 3, Customer Care and Billing Configuration, and the Oracle Utilities Customer Care and Billing user documentation.
- For more information on configuring MDM services, see Chapter 4, Meter Data Management Configuration, and the Oracle Meter Data Mnagement user documentation.
- For more information on configuring NMS services, see Chapter 5, Network Management System Configuration, and the Oracle Network Management System user documentation.
- For more information on configuring BPEL services, see Chapter 7, CSS Direct BPEL Flows.
- For more information on Optional OUCSS-OUCCB BPEL flows Web Services refer to Appendix A, Optional OUCSS-OUCCB BPEL Flows Web Services.

OUCSS Service Name Module Name	Service Description	CCB Service	BPEL Service	Notes
SSInvitePersonList Login	It is responsible for retrieving person information related to an account.	WXInvitePersonList		
SSVerifyAccount Login	This service is used to enroll an account to a self-service user for web access.	WXVerifyAccount		

	The list of verification fields per line of business are defined on the Self-Service Integration master configuration.	
SSViewAccountService Login	This service retrieves account information to display in the self-service application.	WXViewAccount
SSViewAccountList Login	This service accepts a list of accounts from self-service and returns corresponding account information. It uses the information scripts defined on the Self-Service Integration master configuration. In addition, it may also invoke the account list filter script defined on the self-service master configuration to apply filter criteria to the account list.	WXViewAccountList
AccountSummaryService AccountSummary	This service retrieves account charge summary information to display in the self-service application. The service returns • The most recent bill for the account (excludes off cycle bill generated bills) • Account's current balance	WXAccountChargesSummaryRetriever
SSAutoPaySetupUpdateService AccountManagement, AutoPay	This service is used for retrieving and maintaining an account's auto pay details. For auto pay updates, the service merely creates an instance of the Auto Pay Setup business object defined on the Self-Service Integration master configuration.	WXAutoPaySetup
WSSEBillUpdateService	This service is	WXEBillSetup

AccountManagement Electronic Billing	responsible for retrieving and maintaining a customer's bill routing method, e.g., postal, email, etc. This service supports the following modes: • On READ action, it retrieves the account's current bill routing method setting • On UPDATE action, it updates the account's current bill routing method setting	
SSMaintainMailingAddressUpdateService AccountManagement, Account AddressInfo	This service is used for retrieval and updating of customer's mailing address. This service supports the following modes: • On READ action, it retrieves the account's current mailing address • On UPDATE action, it updates the account's mailing address. Note that when a mailing address is updated, it is stored on the person correspondence information	WXMaintainMailingAddressInfo
WSSAccountPhoneInfoUpdateService AccountManagement, AccountPhoneInfo	This service is responsible for retrieval and updating of customer's phone numbers. This service supports the following modes: • On READ action, it retrieves the customer's current phone information • On UPDATE action, it updates the customer's phone information	WXMaintainPhoneInfo
ViewBillService	This service retrieves	WXBillView

BillingHistory	account's bill details to display in the self-service application.			
SSCreateScalarMeterAddService Scalar Meter	It is responsible for retrieving and adding manual or scalar meter reads. When adding a new meter read, the service merely creates an instance of the Meter Read Creation business object defined on the Self-Service Integration master configuration.	WXCreateMeterRead	OUCSSOUMDM2AddScalarMeterReadEBF (For more information see Add Scalar Meter Read Integration Flow)	This module can be configured to used either CCB service or BPEL service to connect to MDM
AlertsService Alerts	This service retrieves a list of alerts to display in the self-service application. The list of alert types and corresponding scripts are defined on the Self-Service Integration master configuration.	WXGetCCBAlerts	OUCSSGetAlertsEBF (This BPEL service retrieves alerts information from CCB and NMS. For more information see Get Alerts Integration Flow)	This module can be configured to call CCB Service or to BPEL service to get alert information from CCB and NMS
SSConsumptionSummaryService ConsumptionSummary	This service retrieves consumption information for display in the self-service application. It retrieves consumption information for service agreements that do not require MDM bill determinants.	WXGetConsumptionSummary	OUCSSOUMDM2GetConsumptionSummaryEBF (For more information see Get Consumption Summary Integration Flow)	This module can be configured to call CCB service or BPEL service to connect to MDM
SSGetRatedSAsService RateAnalysis	This service retrieves the rated service agreements of an account. It also returns the valid rate schedules for every SA's SA Type.	WXGetRatedSAs	(Optional OUCSS-OUCCB BPEL Flow)	
SSRateAnalysisService RateAnalysis	This service receives an SA and a new rate schedule and does a comparison of the bill segments of the SA versus what the charges would have been if the SA was billed using the new rate schedule.	WXRateAnalysis		• CCB Service is calling the CCB-MDM Self Service Usage Request Integration Flow. For more information, refer to Chapter 6, CCB-MDM

				Integrated Flows
SSOneTimePaymentService Payment/OneTime	This service is used for creation of online payments.	WXMakePayment		
BillNotificationUpdateService BillNotification Login Update Profile	This service is responsible for maintaining the billing notification preferences of a self-service user.	WXSetBillNotifyPreference		
SSServiceChargesToDate ServiceChargeToDate	This service retrieves the charges to date for a self-service user's account. The system only attempts to calculate unbilled charges to date for service agreements that require bill determinants from MDM.	WXUsageChargesToDate		• CCB Service is calling the CCB-MDM Self Service Usage Request Integration Flow. For more information, refer to Chapter 6, CCB-MDM Integrated Flows
UsageDetailService UsageDetail	This BPEL service retrieves usage details (interval) for a self-service user's account for some period (e.g., year, month or day). The system will attempt to retrieve usage information from MDM for each of the account's service agreements that require bill determinants. This service may also return temperature information.		OUCSSOUMDM2Direct UsageDetailReqEBF (For more information see Get Usage Detail Integration Flow)	•
SSUsageOverviewService UsageOverview	This BPEL service retrieves x-day usage overview for a self-service user's account. The number of days (x) is provided as input to this service. The system will attempt to retrieve usage information from MDM for each of the account's service agreements that require bill determinants.		OUCSSOUMDM2Direct UsageOverviewEBF (For more information see Direct Usage Overview Integration Flow)	•
CustomerMgmtService	This inbound service is used to process	WXProcessStartStopRequest		

ServiceMgmt	start, stop and transfer service requests. The service supports the following modes: • On READ action, it invokes the start service criteria script defined on the self-service master configuration • On UPDATE action, the service merely creates an instance of the start service task type defined on the master configuration. The start, stop or transfer request is handled within the service tasks's lifecycle.	
PremiseSearchService ServiceMgmt	This service is used to search for a premise. It invokes the existing CCB premise search page to search for a premise by the address field constituents.	WXPremiseSearch
PaymentArrangement PayArrangement	This service is used to process a pay arrangement request. The service supports the following modes: • On READ action, it invokes the payment arrangement eligibility script defined on the self-service master configuration. The eligibility script is responsible for returning the list of available pay arrangement service task types. • On UPDATE action, the service merely creates an instance of the pay arrangement service task type	WXProcessPayArrangementRequest

	selected by the end customer. The pay arrangement request is handled within the service tasks's lifecycle.			
GreenButtonService Usage Download	This BPEL service is used for the Usage Download functionality for both commercial and residential customers. This service calls the MDM service to retrieve usage data and pass it to OUCSSOUMDM2FormatGreenButtonDataEBF which in turn formats the data to CSV or XML format and returns it to the calling service ,OUCSSOUMDM2GetUsageOverviewEBF.		OUCSSOUMDM2GetUsageOverviewEBF OUCSSOUMDM2FormatGreenButtonDataEBF (For more information see Usage Download Integration Flow)	•
SSLookupService Admin/Lookup	This service retrieves data for populating dropdown lists in the self-service application. The following is returned: • Valid credit card types as defined on the Self-Service Integration master configuration • Valid payment types as defined on the Self-Service Integration master configuration • Valid bill route types • Valid phone types	WXGetSelfServiceDropdowns		
SSLabelService Admin/Labels	This service retrieves data for populating field labels in the self-service application.	WXGetSelfServiceLabels		
OutageSummaryService Outage Map (Outage Map and Outage Table screens)	This BPEL service to get all outages from NMS aggregated by Zip, County or City. Out of box, it is aggregated by Zip.		OUCSSOUNMSOutageSummaryEBF (For more information see Outage Summary Integration Flow)	

SSTroubleCallService SSReportOutage	This BPEL service is used to save the outages (public and premise outages) information in NMS database.		OUCSSOUNMSTrouble CallInterfaceEBF (For more information see Trouble Calls Interface Integration Flow)
SSAccountInfoTroubleCodesService SSReportOutage	This BPEL service retrieves the information displayed on the Report Public Outage and Report Premise Outage Screen. It retrieves premise , account information and service related information from CCB and trouble codes information from NMS		OUCSSGetOutageScreenInfoEBF (For more information see Get Outage Screen Info Integration Flow)
OutageMyDetailService OutageMap (My Outage Details screen)	This BPEL service retrieves current and planned outages for a given account from NMS		OUCSSOUNMSOutage DetailEBF (For more information see Outage Detail Integration Flow)
OutagePublicDetailService OutageMap	This process is used to get the outage summary information for the selected Area for the Outage Summary information screen in CSS from the Utility's network management system (e.g., NMS).		OUCSSOUNMSOutage PublicDetailEBF (For more information see Outage Public Detail Integration Flow)
SSContextInfoService AccountContextInfo	This service accepts account id and person id and returns a list of the context properties for the account	WXContextInfo	
SSAccountSearchService CSRAccountSearch	This service accepts account information, together along with search information The web service will return the list of accounts as a response	WXAccountSearch	

SSBudgetDetailService BudgetBilling	The web service accepts account information The list of eligible SAs along with their recommended budget amounts will be sent back to CSS. Ineligible SAs will also be sent, identifying them as such	WXBudgetDetails
SSBudgetRequestUpdateService BudgetBilling	After sending an update request to the service, the service will cancel the account's budget billing plan.	WXProcessBudgetRequest
SSFinancialHistory Financial History	The web service accepts account information and returns bills and payments. Under each bill row, more details related to the bill will be supplied (total adjustments and corrections as well as the individual bill segments reported on the bill). Also, transactions not yet reported on a bill will be displayed individually.	WXFinancialHistoryRetriever
PrepaidBalanceAndChargesService PrePaid	The web service is used for fetching the balance and charges associated with a particular prepaid account	WXRetrievePPBBalanceAndCharges
PrepaidEstimatesAndCostService PrePaid	This web service is used for fetching the estimated number of days available for a particular prepaid customer before his service is cut.	WXRetrievePPBEstimatesAndCosts
ScalarUsageDetailService Scalar Usage Details	The web service accepts on read account information. It returns the list of service agreements with their	WXRetrieveScalarUsage

	corresponding scalar usage details measurement		
OUCSSUploadProcessService FormsMgmt	This BPEL service accepts the CSS request message to upload Document to file server.		OUCSSUploadDocumentEBF (For more information see Upload Document Integration Flow)
OUCSSReadProcessService FormsMgmt	This BPEL service accepts the CSS request message to read Document from file server		OUCSSReadDocumentEBF (For more information see Read Document Integration Flow)
SSFormsListService FormsMgmt	This web service has one action which is READ, when this web service is invoked, the edge application will return the list of form entries for the FORMTYPE.	WXFormsList	
SSFormsManagementUpdateService FormsMgmt	On UPDATE the web service is passed with the updated Form details. If the transaction is successful a reference number is returned back	WXFormsManagement	
WXRetrieveAccountDocumentsService SSReadAccountDocuments	The web service returns a list of valid documents related to the given account and the description of the documents	WXRetrieveAccountDocuments	
OUCSSReadDocumentService SSReadAccountDocuments	The web service will be used for reading a particular document from the list of document URLs returned by CCB. Integration retrieves the content either from the file system or through a third party document management system.		OUCSSReadDocumentEBF (For more information see Read Document Integration Flow
SSNotifPreferences Notification Preferences	This service is used to retrieve preferences for a given account to a self-service user. The notification types	WXSetNotificationPreferences	OUNCWXNotificationPreferences (For more information see Notification Management OUCSS Integration Services)

	for the preferences are retrieved dynamically for CCB and from the OUNC database for NMS. This service is called only when oucss.notifications.owner = 'OUNC' i.e., when Notification center maintains the delivery channels and preferences. See SSCommPreferencesUpdate for details on how CCB maintains preferences and channels.		
SSDeliveryChannelsService SSDeliveryChannels	Responsible for retrieving delivery channels information for the self-service user.It is the Profile page under Notifications. This service is called only when oucss.notifications.owner = 'OUNC' i.e., when Notification center maintains the delivery channels and preferences. See SSCommPreferencesUpdate for details on how CCB maintains preferences and channels.		OUNCWXDeliveryChannels (For more information see Notification Management OUCSS Integration Services)
SSNotificationListService Notification List	This service retrieves all notifications sent out from edge application for a given account. Notifications previously sent from edge applications are retrieved from the OUNC database.		OUNCWXGetNotificationsService (For more information see Notification Management OUCSS Integration Services)
SSCommPreferencesUpdate Notification Preferences	This service is used to retrieve preferences for a given account to a self-service user. The notification types and the preferences are retrieved	WXMaintainCommPreferences	

	dynamically from CCB . This is called when oucss.notification.owner configuration is set to CCB	
SSCancelSchedPay One Time Pay	Cancel Scheduled One Time Payment	WXCancelSchedPay
WXUsageChargesProjectedService BillCharges Projection	Responsible for getting projected service charges until next bill date, and the average service charges for prior bills.	WXUsageChargesProjected
LangTimeZoneService LangTimeZonePref	Responsible for maintaining a customer's preferred language and time zone. This service is used to read and update CCB with the customer's preferred language and timezone. This service supports the following modes: •- On Read action, it retrieves the person's language and time zone. •- On Update action, it updates the person's language and time zone. Note that when a language and time zone is updated, it is stored on the person correspondence information.	WXMaintainPersonLanguageTimeZone
MailingAddressValidation AccountAddressInfo	This service is responsible for maintaining a customer's mailing address. This service supports the following modes: •- On Read action, it retrieves the account's current mailing address •- On Update action,	WXMaintainMailingAddressInfo

	it updates the account's mailing address. Note that when a mailing address is updated, it is stored on the person correspondence information	
UserLinkService Login	This service supports Enroll, Add, Update, and Remove actions. The actions reflect the actions of the account/person relationship. An account and person are either associated with each other or no longer associated with each other in CSS. The update action can indicate a change to person information such as email or a change to account/person information such as access role.	WXMaintainCSSUserAccount
Commercial Web Services (Web Services that are exclusive to commercial context)		
SSCommercialAccountList AcctManagement	This is the same service used for the This service accepts a list of accounts from self-service and returns corresponding account information. It uses the information scripts defined on the Self-Service Integration master configuration. In addition, it may also invoke the account list filter script defined on the self-service master configuration to apply filter criteria to the account list.	WXViewAccountList
CommercialBillHistoryService SSCommercialBillHistory	This service accepts a list of accounts and returns the financial history for the accounts. The base financial history	WXMultipleAccountFinancialHistory

	service is called	
SSCommercialUsageComparisonService SSCommercialUsageComparisonService	This BPEL service accepts list of accounts. It returns usage information as a list, with one list entry per account, service type, uom and sqi. The usage information has TOU amounts summed for each period	OUCSSOUMDMMultiAccountUsageComparisonEBF (For more information see Multiple Account Usage Comparison Integration Flow)
SSCommercialUsageAggregationService SSCommercialUsageAggregationService	The BPEL web service accepts list of accounts. It summarizes the usage by account, service type, uom, sqi, overlay uom and returns the aggregated results	OUCSSOUMDMMultiAccountUsageAggregationEBF (For more information see Multiple Account Usage Aggregation Integration Flow)

OUCSS Portal Solution

The reference WebCenter Portal archives contains OUCSS taskflows. The taskflows are spread across different pages and logically grouped in separate Portals. The WebCenter Portal uses the default WebCenter security model to restrict access only to certain groups and users.

Responsive Design

Based on device type used to access the Portal (desktop/laptop/tablet or mobile device), the Portal automatically loads the views and resources targetted for the device type. On desktop/laptop/tablets, Portal pages are loaded, but when accessed from a smart/mobile phones, Mobile pages will be loaded. Mobile view does not support all functaionalities as the desktop version.

Taskflow that support Responsive design expose a parameter “deviceAgent”. This parameter can be used to override the automatic detection of device type.

Parameter	Comment
DESKTOP	Renders the desktop view even on mobile devices. (Default)
MOBILE	Renders the mobile view even on desktop and tablets.

Portal Pages

OUCSS solution imports three Portals (public, oucss and OUCSSAdmin) with pages containing OUCSS taskflows. **Self Service** portal (public) contains all pages accessible by everyone/public users. The OUCSS Portal (oucss) contains pages that are secured and accessible only by authenticated/loggedin users. The **Admin** (OUCSSAdmin) contains admin and configuration pages that are accssible by Admin and CSR users.

The pre-configured **WSSAdmin** user is the moderator of all the three Portals. This user has complete control on all the portal and can add, modify, delete any resource of the portals.

Portal Title	Portal Name	Accessibility	Description
Self Service	public	Public/Everyone	Portal containing public pages accessible by all users.
OUCSS Portal	oucss	Authenticated/Logged In users	Portal containing account/user related pages.
OUCSS Admin	OUCSSAdmin	WSSAdminGroup (all Pages) WSSCSRGroup (only the Customer Search page)	Portal containing admin/csr related pages.

Self Service Portal (public)

The **Self Service** Portal is a public portal to allow any user in the World Wide Web access to public pages like Home, Login, Register, etc. This Portal is defined public and has no secured page.

Visible Public Pages

- Home
- Login
- Retrieve User
- Register
- Outage
- Outage Map
- Outage Table
- Report Outage

Hidden Public Pages

Apart from the public system pages provided by WebCenter Portal, the following are hidden pages in this Portal

- Validate Email

OUCSS Portal (oucss)

The **OUCSS** Portal is a secured portal to allow access only to authenticated users. This portal allows users to access account related pages like Dashboard, View Bill, Personal Information, etc. This Portal is defined private and has no public page.

Visible Secured Pages

- Home
- Accounts
- Details

  - Dashboard
  - Financial History
  - Budget Billing (Post Paid accounts only)
  - View Bill
  - Make Payment
  - Payment Arrangement (Post Paid accounts only)
  - Compare Rates
  - Usage Details
  - Personal Information
  - My Outages
  - Report Outage (Premise)
  - Start Service
  - Stop Service
  - Transfer Service
  - Account Documents
- Multi-Account
  - Set Accounts
  - Financial History
  - Usage Compare
  - Usage Aggregate
- Notification
  - Inbox
  - Profile
  - Preferences
- Forms
  - Log an Issue
  - Form List
- Outages
  - Display Map
  - Display List
  - Report Public Outage
- New Customer
- User Profile

Hidden Secured Pages

OUCSS Portal contains the following are hidden pages

- Add Meter Read

- Manage Address
- Manage Phone
- Manage Bill Notification
- Manage Electronic Bill
- Form Update
- Language & Timezone (for Update)

Mobile Secured Pages

OUCSS Portal contains the following are pages that are displayed when accessing the Portal using a mobile/smart phone. These pages breakdown complex pages (e.g. Dashboard) and display individual taskflows for better rendering.

- Change Account Access
- Account Summary
- Alerts
- Service Charges to Date
- Consumption Summary
- Usage Overview
- Manage Auto Pay
- Prepaid Balance and Changes
- Prepaid Estimates and Cost
- Promotions & Offers

OUCSS Admin Portal (OUCSSAdmin)

The **OUCSS Admin** Portal is a secured portal to allow access only to Admin and CSR users. This portal allows CSR users to access Customer Search page and Admin users to admin/configuration related pages like Resources, Labels, Lookups, etc. This Portal is defined hidden and has no public page.

Visible Secured Pages

- Customer Search
- Configuration Options
- Resources
- Access
- Security
- Edge Application
- Line of Business
- Portlets
- Language
- Labels

- Lookup
- Messages
- Train
- Offers

OUCSS Taskflows dropped on each Page
Public Pages

Home

- No Taskflows. This page is left empty.

Login

- Login

Register

- Registration

Outage

Outage List

- Outage Map (with displayMode 1 to display Outages in tabular format)

Outage Map

- Outage Map (with displayMode 0 to display Outages in map format)

Report Outage (Public)

- Report Outage (with premiseFlg as N to report Public Outage).

Reset Password

- Forgot Password (with mode param set to RESET_PASSWORD)
- Forgot Password (with mode param set to RETRIEVE_USERID)

Validate Email (Hidden)

- Validate Email

Secure Pages

Accounts

- Account Search
- Account List (mode param set to LIST)
- Remove Account
- Invite
- Account Nickname
- Enroll
- User Link (used as taskflow call in Invite, Enroll & Remove Account taskflows)

Details

Dashboard

- Account List (mode param set to DROPDOWN)
- View Account Information
- Account Charges Summary (Visible for Post Paid accounts only)
- Prepaid Balance and Charges (Visible for Pre Paid accounts only)
- Alerts
- Manage Budget Billing (Visible for Post Paid accounts only)
- Prepaid Estimates and Cost (Visible for Pre Paid accounts only)
- Service Charges to Date
- Bill Charges Projection
- Usage Overview
- Scalar Consumption Summary
- Banner Promotion

Financial History

- Account List (mode param set to DROPDOWN)
- View Account Information
- Financial History

Budget Billing

- Account List (mode param set to DROPDOWN)
- View Account Information
- View Budget Billing

View Bill

- Account List (mode param set to DROPDOWN)
- View Account Information
- View Bill

Payment Arrangement

- Account List (mode param set to DROPDOWN)
- View Account Information
- Payment Arrangement

Compare Rates

- Account List (mode param set to DROPDOWN)
- View Account Information
- Rate Analysis Control
- Rate Analysis Table View
- Rate Analysis Graph View

Usage

- Account List (mode param set to DROPDOWN)
- View Account Information
- Usage Detail (New with Time Selector)
- Scalar Usage Detail
- Usage Download

Personal Information

- Account List (mode param set to DROPDOWN)
- View Account Information
- View Mailing Address
- View Phone Information
- Displays Phone Information if isUsingLegacyContct is true.
- Displays Person Contacts if isUsingLegacyContct is false.
- View Billing Notification Preferences
- View Bill Delivery Method
- View Automatic Payment Option
- Language and TimeZone (mode param set to READ or Blank)

My Outage Details

- Account List (mode param set to DROPDOWN)
- View Account Information
- Outage Details

Report Outage (Premise)

- Account List (mode param set to DROPDOWN)
- View Account Information
- Report Outage (with premiseFlg as Y to report premise Outage).

Start Service

- Account List (mode param set to DROPDOWN)
- View Account Information
- Start Service

Stop Service

- Account List (mode param set to DROPDOWN)
- View Account Information
- Stop Service

Transfer Service

- Account List (mode param set to DROPDOWN)
- View Account Information
- Transfer Service

Account Documents

- Account List (mode param set to DROPDOWN)
- View Account Information
- Read Account Documents

Multi-Account

Set Accounts

- Set Accounts
- Create Set

Financial History

- Business Context

- Commercial Financial History

Usage Compare

- Business Context
- Usage Comparison

Usage Aggregate

- Business Context
- Usage Aggregation
- Usage download

Notification

Inbox

- Account List (mode param set to DROPDOWN)
- View Account Information
- Inbox

Profile

- Account List (mode param set to DROPDOWN)
- View Account Information
- Profile

Preferences

- Preference

New Customer

- New Customer Service

User Profile

- Change User Profile
- Change Password

Forms

Log an Issue

- Enter Form

Note. Current page supports only one form type Issues. A new portlet page needs to be developed to support additional form types.

Forms List

- Form List

Secure Hidden Pages

Scalar Read

- Account List (mode param set to DROPDOWN)
- View Account Information
- Add Meter Reading

Pay Now

- Account List (mode param set to DROPDOWN)
- View Account Information
- Make One Time Payment

Manage Address

- Account List (mode param set to DROPDOWN)
- View Account Information
- Update Mailing Address

Account Address Management(Invoked if Address Validation is enabled) Manage Phone

- Account List (mode param set to DROPDOWN)
- View Account Information
- Update Phone Information

Manage Auto Pay

- Account List (mode param set to DROPDOWN)
- View Account Information
- Update Automatic Payment Option

Manage Bill Notify

- Account List (mode param set to DROPDOWN)

- View Account Information
- Update Billing Notification Preferences

Manage EBill

- Account List (mode param set to DROPDOWN)
- View Account Information
- Update Electronic Bill Option

Language and TimeZone

- Account List (mode param set to DROPDOWN)
- View Account Information
- Language and Timezone (mode param set to UPDATE)

Form Update

- Update Form

Mobile Pages

Portal when accessed on a Mobile device loads mobile pages that are designed for smartphones or small screens. User can access mobile public pages such as Home, Register, Outage, etc and secured pages related to their accounts.

Visible Public Pages

- Home
- Login
- Register
- Retrieve User
- Outage
- Display Map (Outage)
- Display List (Outage)
- Report Public Outage

Hidden Public Pages

These pages are not part of the Navigation model.

- Validate Email

Visible Secured Pages

- Accounts
- Dashboard
  - Change Account Access
  - Account Summary (Postpaid Accounts)
  - Alerts
  - Service Charge To Date
  - Consumption Summary
  - Usage Overview
  - Manage Auto Pay
  - Financial History
  - View Bill
  - Make Payment
  - Prepaid Balance And Charges (Prepaid Accounts)
  - Prepaid Estimate And Cost (Prepaid Accounts)
  - My Outages
  - Report Outage (Premise)
  - Promotion & Offers
  - Log an Issue
- Forms
  - Log an Issue
  - Form List
- Notifications
  - Inbox
  - Profile
  - Preferences

Hidden Secured Pages

These pages are not part of the Navigation model.

- Add Scalar Read
- Update Form

OUCSS Taskflows Dropped on Each Page

Public Pages

Home

- No Taskflows. This page is left empty.

Login

- Login

Register

- Registration

Outage

Display List

- Outage Map (with displayMode 1 to display Outages in tabular format)

Display Map

- Outage Map (with displayMode 0 to display Outages in map format)

Report Outage (Public)

- Report Outage (with premiseFlg as N to report Public Outage).

Reset Password

- Forgot Password (with mode param set to RESET_PASSWORD)
- Forgot Password (with mode param set to RETRIEVE_USERID)

Validate Email (Hidden)

- Validate Email

Secure Pages

Accounts

- Account List
- Enroll

Dashboard

Change Account Access

- Account Access
- Remove Account
- Invite
- Account Nickname

Alerts

- Alerts

Account Summary

- Account Charges Summary (Visible for Post Paid accounts only)

Prepaid Balance and Charges

- Prepaid Balance and Charges (Visible for Pre Paid accounts only)

Prepaid Estimates and Cost

- Prepaid Estimates and Cost (Visible for Pre Paid accounts only)

Service Charges To Date

- Service Charges to Date

Usage Overview

- Usage Overview

Consumption Summary

- Scalar Consumption Summary

Promotions & Offers

- Standard Promotion

Manage Auto Pay

- View Automatic Payment Option

Financial History

- Financial History

View Bill

- View Bill

Make Payment

- One Time Payment

My Outage Details

- Outage Details

Report Outage (Premise)

- Report Outage (with premiseFlg as Y to report premise Outage).

Inbox

- Inbox

Profile

- Profile

Preferences

- Preference

Log an Issue

- Enter Form

Note. Current page supports only one form type Issues. A new portlet page needs to be developed to support additional form types.

Forms List

- Form List

Secure Hidden Pages

Scalar Read

- Add Meter Reading

Form Update

- Update Form

OUCSS Context

OUCSS can be configured to fetch consolidated accounts from all LOBs or filter accounts by LOB. By default, Portal is set to fetch consolidated account. This is controlled by the configuration option property “oucss.context.selection”.

The following table summarizes the available options/

Value	Behavior
A	All user Accounts are fetched irrespective of the LOB. Login taskflow does not display the drop down to select LOB. Default value.
S	Accounts are fetched based on LOB. Login taskflow displays the drop down to select LOB.

OUCSS Portal supports login to multiple context based on the LOB (e.g. Residential, Commercial) that allows customizing separate Page Templates, Navigation and pages for each context. OUCSS Login supports 2 modes Consolidated Account and LOB Specific.

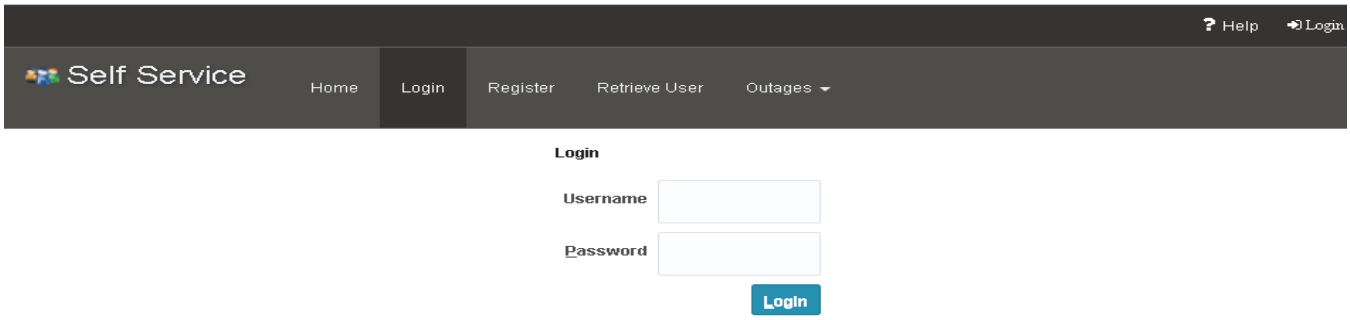
Login Configuration

OUCSS Portal supports consolidated accounts login or LOB based context login (e.g. Residential, Commercial etc.). Each context can be customized to use separate Page Templates, Navigation and pages.

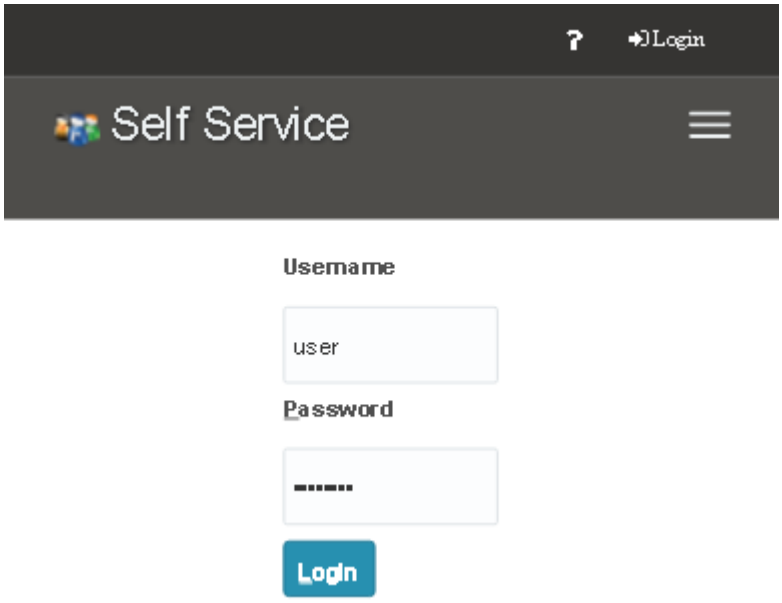
Consolidated Account Login

Login taskflow with consolidated account hides the selection of Login context. The context is null (no context) in this mode. The selection of Portal Resources is determined based on the account selected by the user. For example, if the user clicks on Residential Account, then the Portal Resources configured for Residential LOB are loaded.

Desktop



Mobile



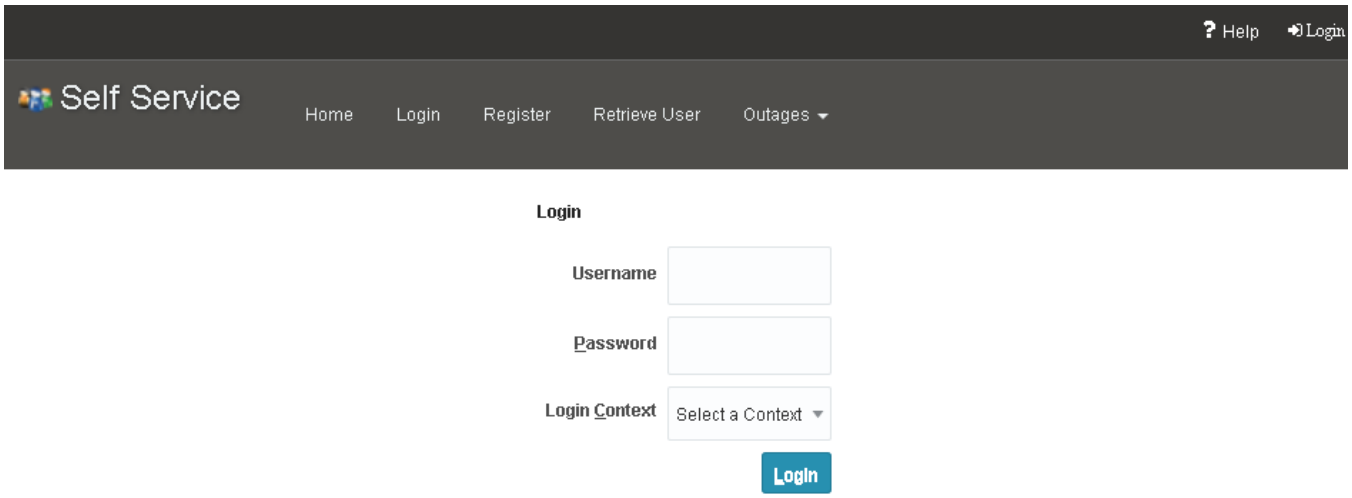
LOB Context Login

If the consolidated account mode is disabled, then login taskflow enables selections of an LOB to filter the accounts and pre-select the Portal resources (Page templates, navigation model, skins etc.) based on the selected LOB.

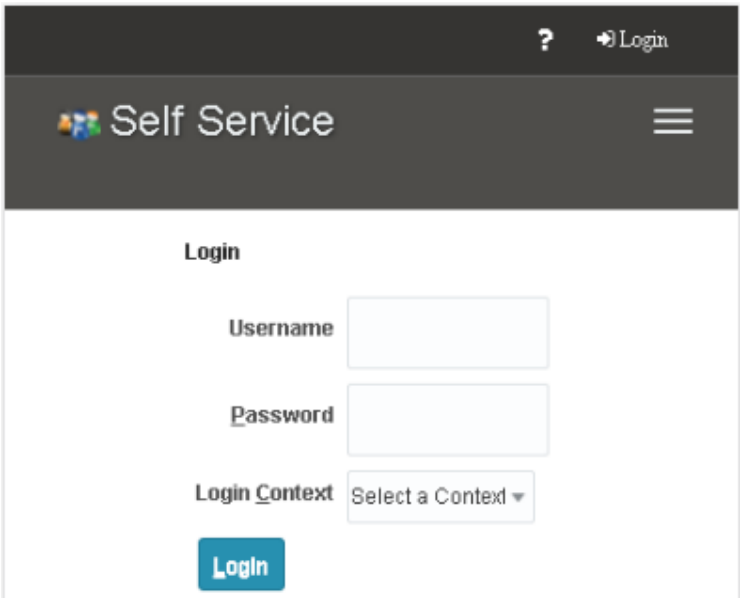
Generic Context Login

By default when Login taskflow is generic (the LOB taskflow parameter is empty). In this mode, a list box is shown to allow users select the Context along with entering Username and password before login.

Desktop



Mobile

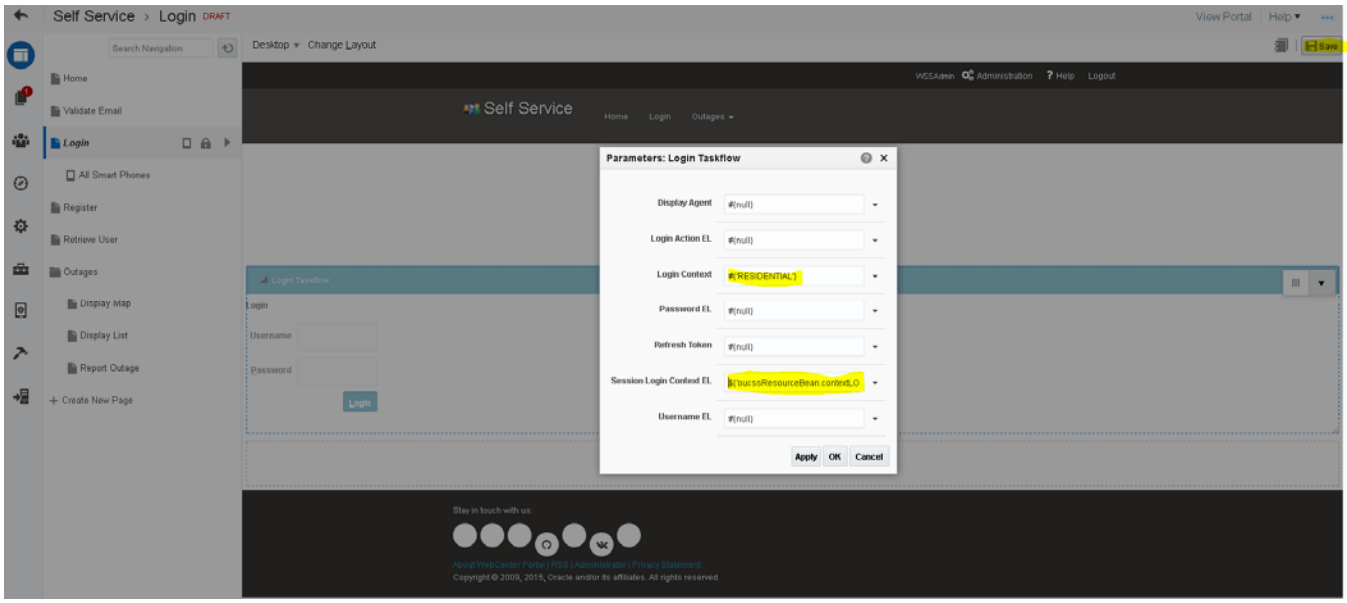


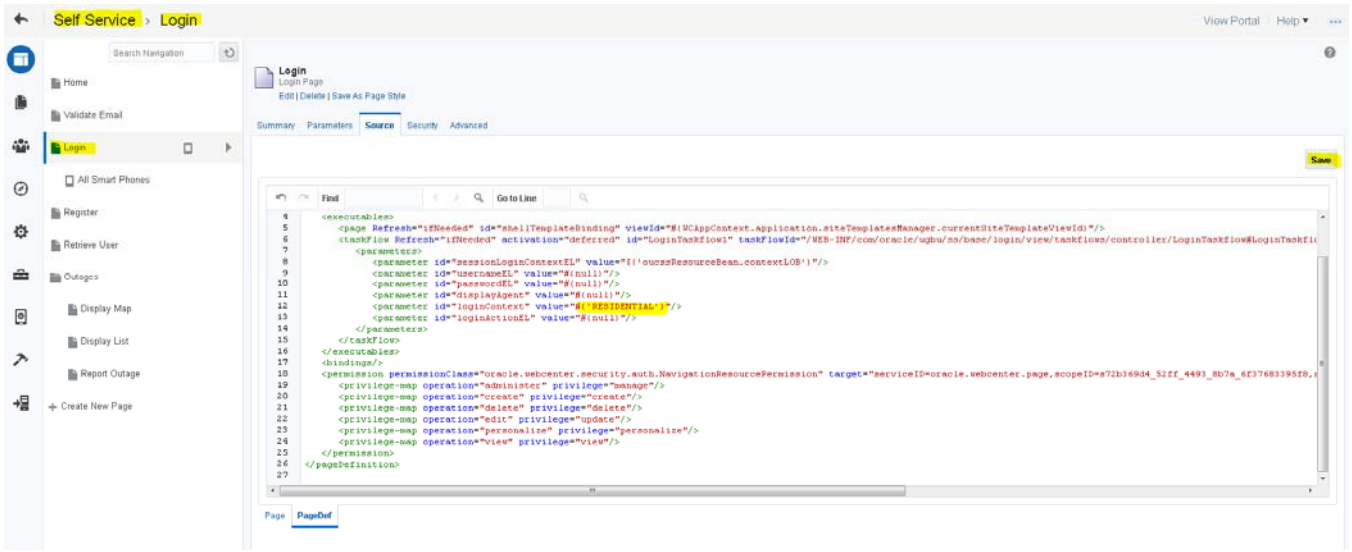
Context Specific Login

To create a Context Specific taskflow, drop the Login Taskflow and update the “Login Context” input parameter with the LOB code defined in OUCSS LOB table. This will enable the Login to login with the specific context.

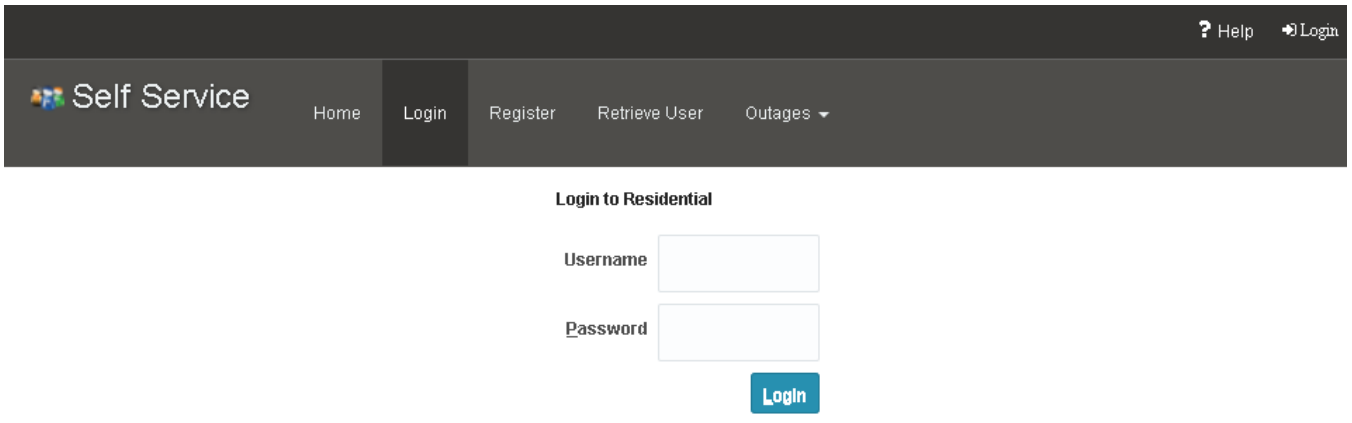
Also, to set the Context into session, update the “Session Login Context EL” with EL to store the context. For OUCSS Portal, use the EL ${'oucssResourceBean.contextLOB'}.

1Set Login Context parameter for context specific login.





Alternative approach to update the Login Context using Source



CSS-CCB User Linking/Syncing

A new configuration property “oucss.link.ccb.person” is added to enable linking CSS users with CCB person. If CCB notification is used, then implementation should set the value of “oucss.link.ccb.person” property to “Y” to enable user linking.

If user link is enabled, every time a CSS users enrolls to account, invites another user, removes an account, removes and invited account or updates email address then the CCB person is added, removed or updated to be in sync with the CSS user.

Portal Resources

An OUCSS Portal solution allows configuration of separate Portal Resources for each LOB and for each display type (Desktop and Mobile). WebCenter provides many out-of-box Portal resources (Page Templates, Skins, Resource Catalog , etc.); however, customized portal resources are provided with the OUCSS solution.

Page Template

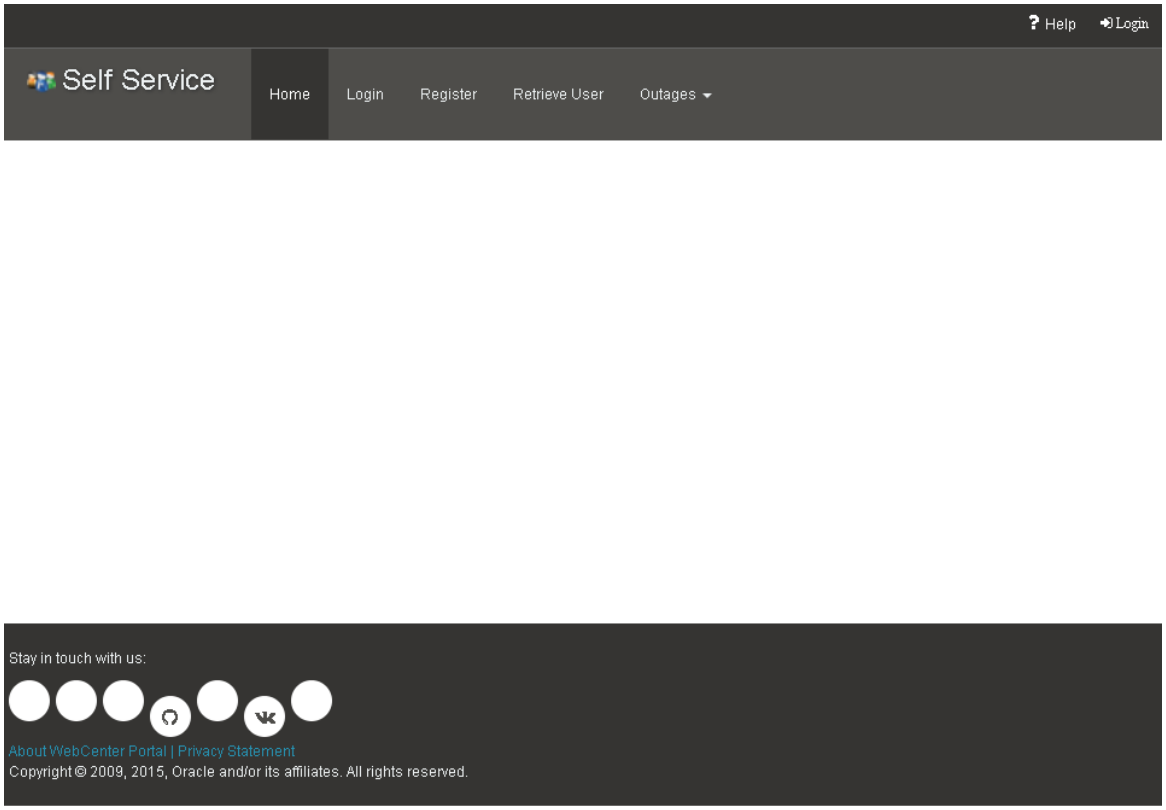
For Residential and Public users, the Portal is configured to use the **OUSS Residential Template** that controls the layout, navigation (both main menu and sub-menu), as well as the links on the boilerplate. Portal Administrators can add more page templates or update the existing one to change the look and feel of the Portal.

For Commercial users, the Portal is configured to use the **OUCSS Commercial Template**.

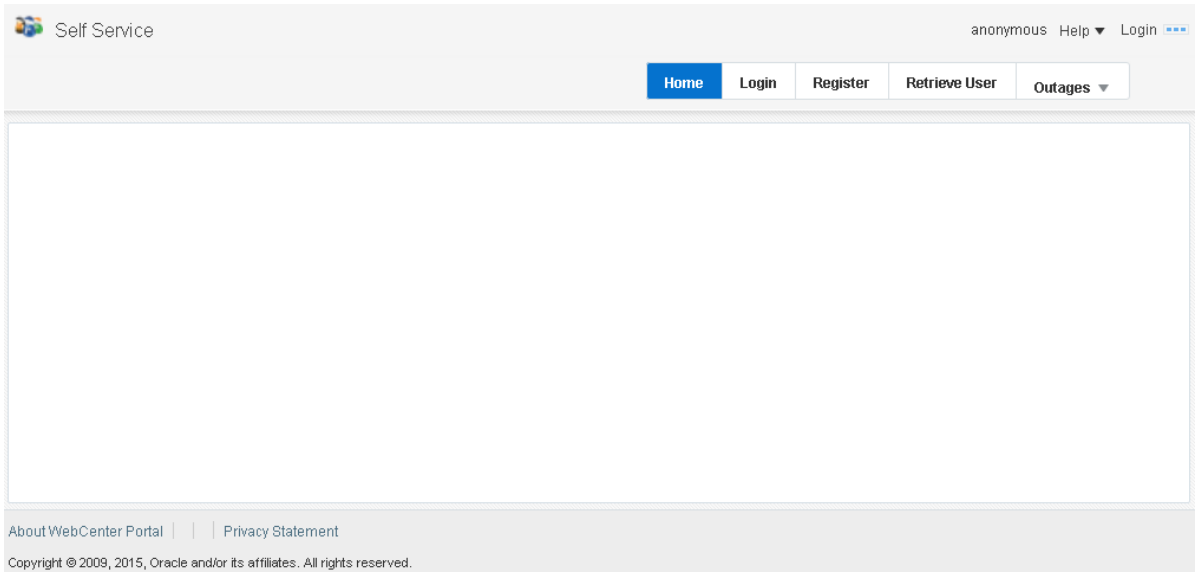
For Mobile devices, the Portal is configured to use the **OUSS Residential Template** for all types of users. This template is responsive and its layoutchanges based on the display type. Implementors can configure a separate template for each LOB/Context to be used for mobile devices.

Page Templates for Portal can be configured using the **OUCSS Admin > Lookup Screen**. See the Portal Resource Management section later in this guide for more details.

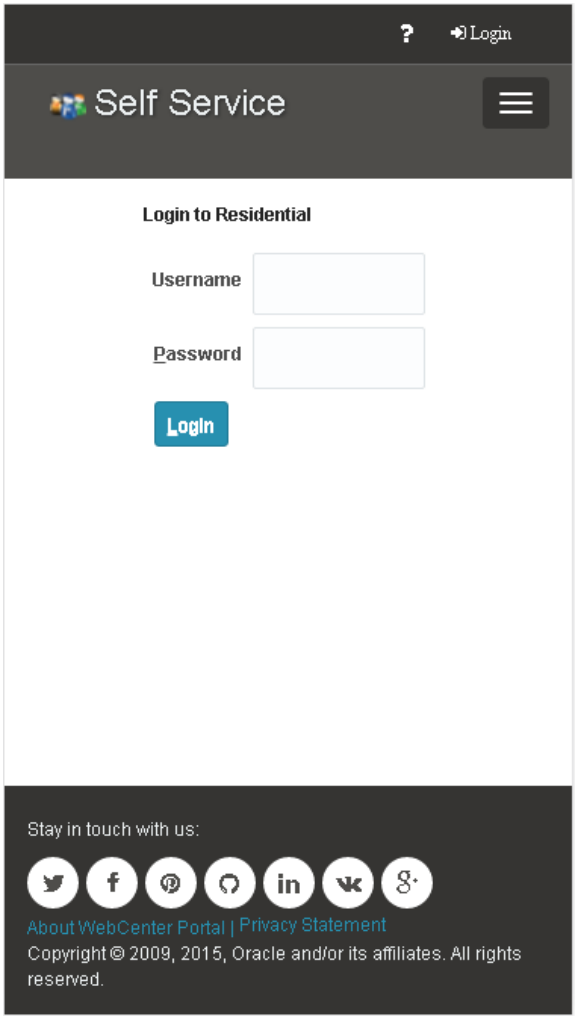
Preview of the OUSS Residential Template



Preview of the OUCSS Commercial Template



Preview of the OUCSS Residental Template on Mobile



Navigation

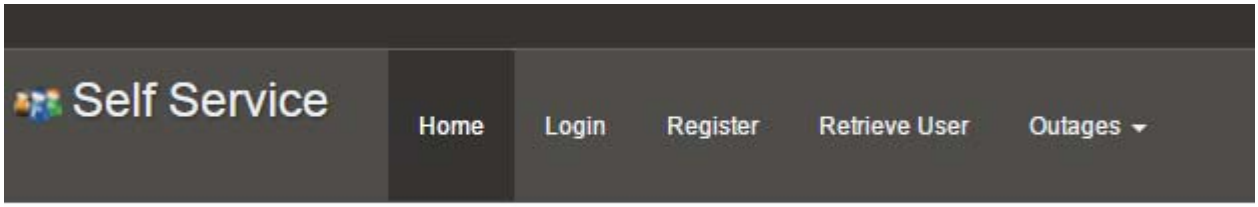
Navigation in each OUCSS Portal is configured to hide and show link based on device type, login information, account type etc using EL expression in the “Visible” attribute.

Navigation for OUCSS Portal can be configured by directly editing the Portal and adding or deleting pages, subpages, links etc. Please refer to Editing Portal Navigation documentation http://docs.oracle.com/middleware/12211/wcp/build/GUID-B4D173AA-5FCC-4E15-BAB4-2D6A52263C81.htm#WCPAA14959 for more information.OUCSS Portal can be customized to hide or remove pages and taskflows invoking MDM or NMS if not necessary

Desktop Navigation

Self Service Portal Navigation (Public)

Public users can access the portal without logging in, and access Public pages such as Home or Registration. Self Service (public) Portal is configured for public users. This navigation contains links to Public Pages.



Navigation Model for Authenticated Users

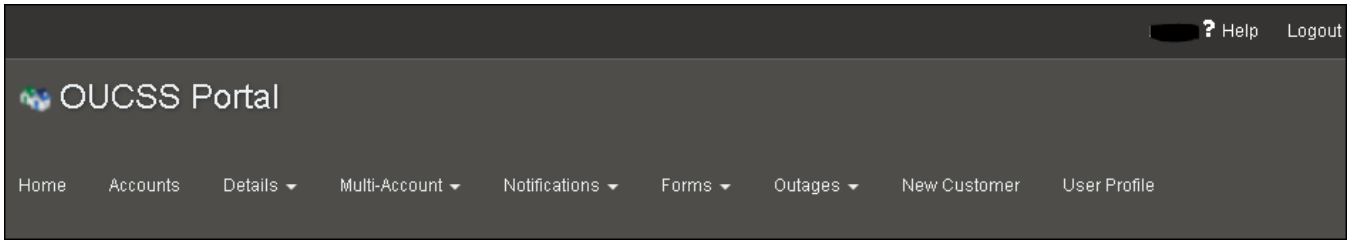
Authenticated users are users who login to Portal with a valid user name and password.

Upon login, the user has access to all public and secured pages. If the user is enrolled to one or more accounts, the Details, Multi-Account, Notification and Forms links are visible.

OUCSS Portal can be configured to use separate navigation model for each supported LOB/context. to secured pages. The Navigation model are customized to restrict links based on the criteria like valid users (that is users who are registered using OUCSS the Registration), enrolled to one or more accounts, or user with Administrator or CSR privileges.

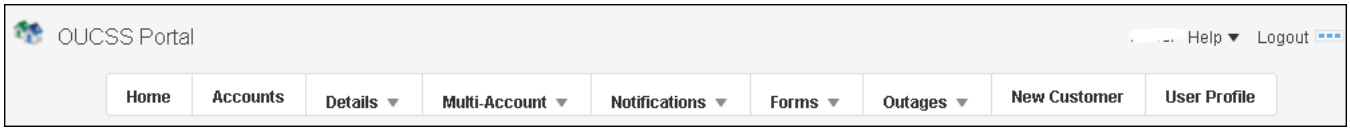
Residential Accounts

OUCSS Portal is configured with the following Navigation links for residential accounts to allow access to secured pages.



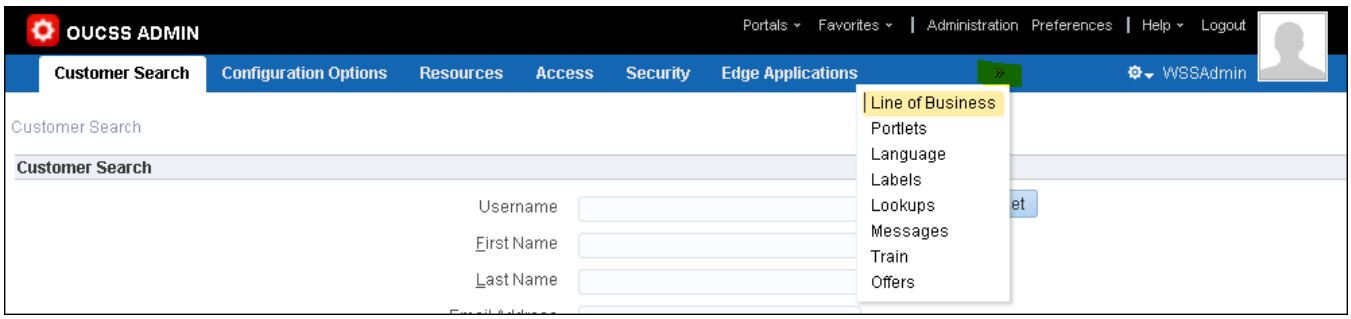
Menu with Commercial Template for Commercial Accounts

OUCSS Portal navigation rendered using commercial site template. The navigation links for commercial accounts are same as residental accounts in this release.



OUSS Admin Portal Navigation

Users who are member of the WSSAdminGroup are referred as OUCSS Admin. OUCSS Admins are allowed access to the OUCSS Admin Portal and admin pages menu.



Navigation for WSSCSR User

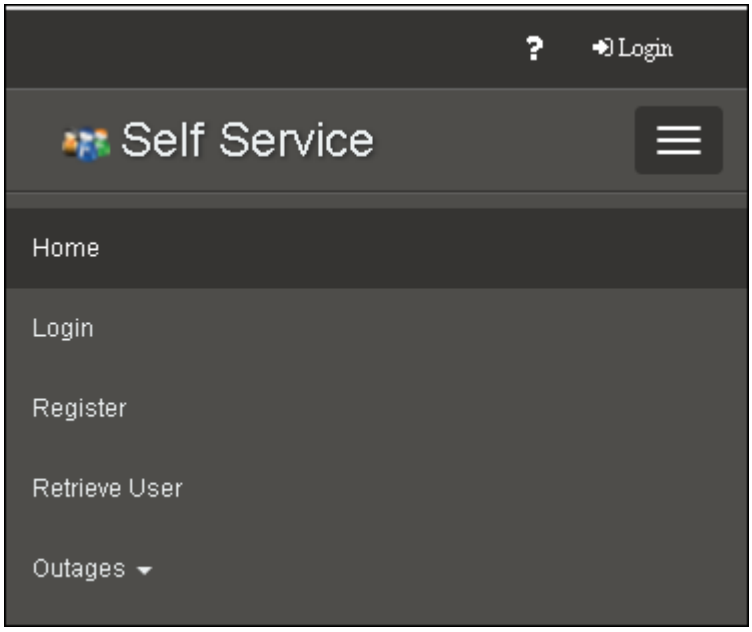
A CSR is a user who is a member of the WSSCSRGroup. CSRs have ability to impersonate any valid Portal user. CSR users can only the Customer Search page from OUSS Admin Portal to search for any user in the system and access their account details. Refer to Customer Search for more information.

Mobile Navigation

When the Portal is accessed using Mobile devices, OUCSS Portals are configured render links to only supported modules. The links in the OUCSS Portal use the EL #{DeviceAgent.currentScopeDeviceGroup.name eq 'AllMobiles'} to hide/show pages/links in navigation model on mobile devices.

Navigation Model for Public Users

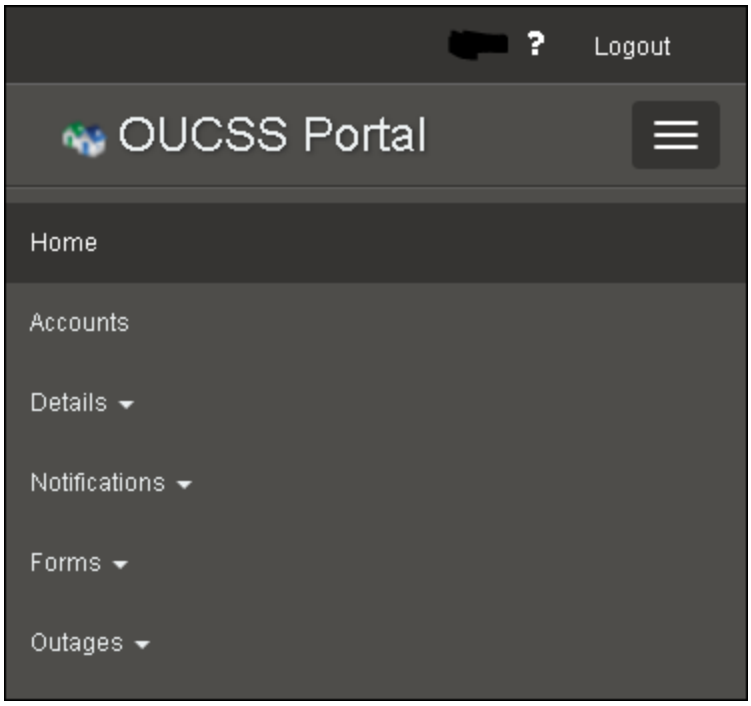
Public users can access the portal without logging in, and access Public pages such as Home or Registration. Portal is configured to use “OUCSS Mobile Navigation Model” for public users on Mobile. This navigation model contains links to Public Pages.



Navigation Model for Authenticated Users

Authenticated users are users who login to Portal with a valid user name and password.

On login, the user has access to all public and secured pages. If the user is enrolled to one or more accounts, the Accounts and Logout links are visible along with other public pages.



Skins

OUCSS Portal is configured to use separate Skin for each LOB/Context supported.

Skins for OUCSS Portal can be configured using OUCSS Admin -> Lookup Screen. See section Portal Resource Management later in this guide for more details. Following table summerizes the Skins that are configured for each LOB.

LOB	Skin Family	Description
Public/Default	OUCSSMosaic	Skin based on default Mosaic skin provided out-of-box by WebCenter Portal.
Residential	OUCSSMosaic	Skin based on default Mosaic skin provided out-of-box by WebCenter Portal.
Commercial	OUCSSAlta	Skin based on default Alta skin provided out-of-box by WebCenter Portal.

For Mobile devices, OUCSS Portal is configured to use the OUCSSMosaic Skin Family.

LOB	Skin Family
Public/Default	OUCSSMosaic
Residential	OUCSSMosaic
Commercial	OUCSSMosaic

Resource Catalog

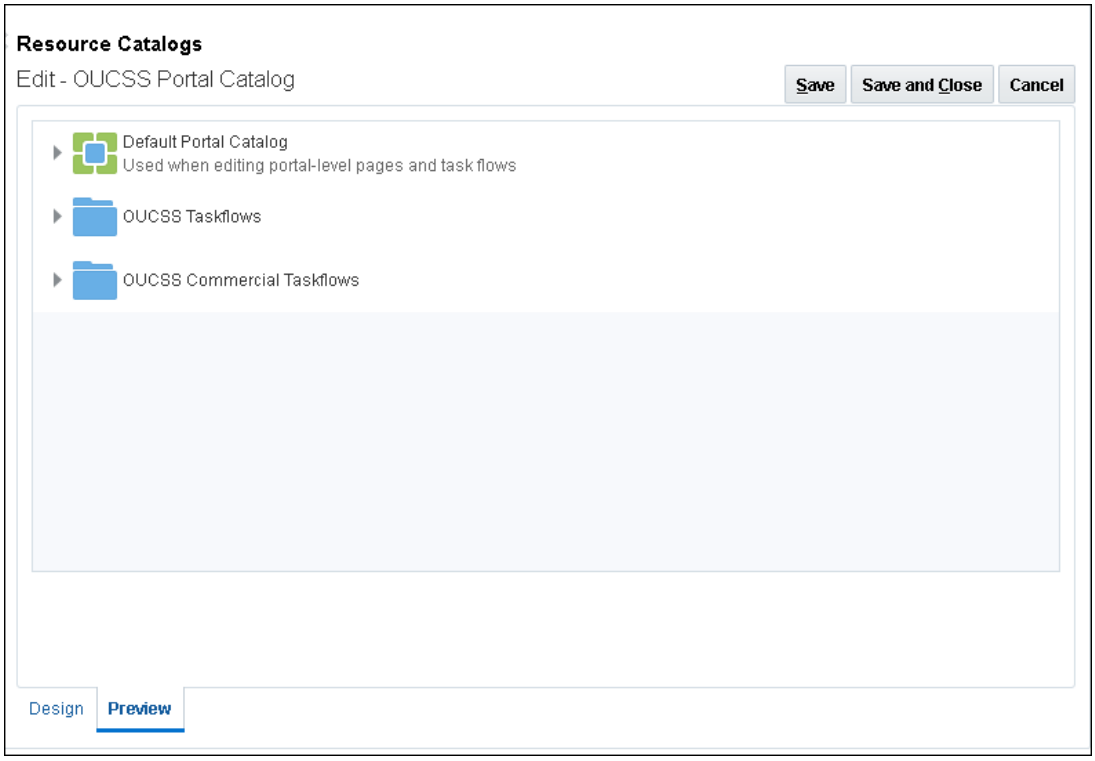
OUCSS Portal is configured can be configured use separate Resource Catalog for each type of LOB/Context and device types. OUCSS Admin Users can create or edit the page and drop required components/OUCSS taskflows using Resource Catalog.

Resource Catalog for OUCSS Portal can be configured using OUCSS Admin -> Lookup Screen. See section Portal Resource Management later in this guide for more details.

OUCSS Portal Catalog

OUCSS Portal is configured to use OUCSS Portal Catalog for Residential Context and Commercial Context. Default OUCSS Portal Catalog is configured with all OUCSS Core Taskflows.

OUCSS Taskflows are logically grouped with the modules name, as shown in the following preview of the OUCSS Portal Catalog.

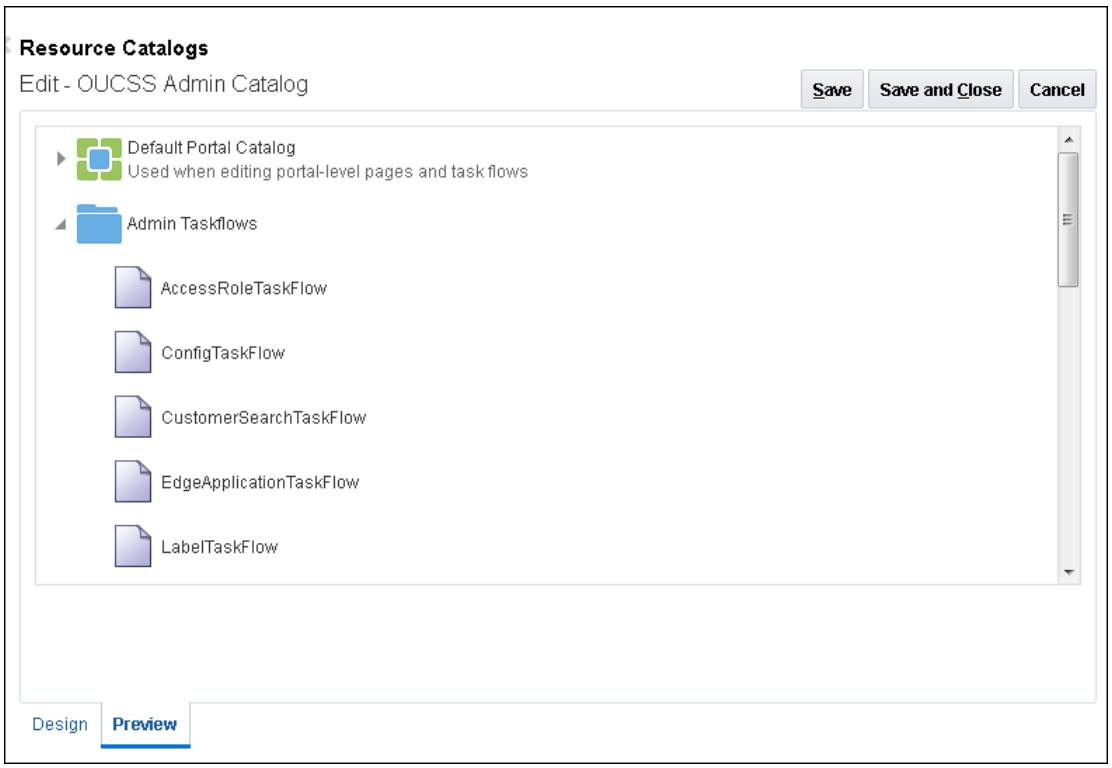


OUCSS Admin Catalog

OUCSS solution is configured to use the OUCSS Admin Catalog for Admin Portal/pages.

.

2 Preview of the OUCSS Admin Catalog



Reference Security Roles

The WebCenter Portal Application manages Tier-1 security. Most of the pages in the application are secured and are accessed only by specific enterprise groups/roles. Some pages are public and can be accessed by any user without logging in.

As part of the installation, two enterprise groups and two users are imported into LDAP. The enterprise groups are hierarchical.

Enterprise Groups

There are two enterprise groups provided in the OUCSS Portal solution.

Users who belong to the **WSSAdminGroup** enterprise group serve as administrators of the OUCSS application. Ideally system administrators will be members of this group.

Users who belong to the **WSSCSRGroup** enterprise group can perform CSR-related functions. Ideally, your CSRs who directly interact with consumers will be members of this group.

Pre-configured Users

WSSAdmin is the administrator of the OUCSS Portal. This user can manage all resources of the portal. **WSSAdmin** user is a member of the **WSSAdminGroup**.

WSSCSR is provided for certain group of users that need to perform CSR-related functions. This user is part of WSSCSRGroup. This user can carry out the same set of actions for any registered user who has access to a utility account.

OUCSS Application Configuration (Admin)

CSR View

To facilitate loading of any portal page for a given account id, a CSR or Admin can use the CSRView.jspx. The CSR View takes certain query parameters and navigates the CSR to desired target URL.

CSR or Admin can also use the Customer Search to search the Account Id and then navigate to Dashboard by clicking the View button. But this is multi-step process which is simplified by the CSR View page.

Note: If CSR/Admin is not logged in, then a login challenge will be displayed before transferring user to the target page. Only users from WSSCSRGroup are allowed to access the CSRView page. A error will be displayed if any other user tries to access the CSR View.

Implementation can use the following URL format to load any portal URL for a given account Id.

```
http://<<PortalHost>>:<<PortalPort>>/webcenter/portal/system/CSRView?
wssAccountId=<<accountId>>&personId=<<personId>>&wssTargetURL=<<relativeURL>>
```

The following query parameters are expected for the given type of request.

Request Parameters

Parameter Name	Description
wssAccountId	Account Id that needs to be set to the context.(*Required) If not provided then CSR Landing page will be loaded.
wssPersonId	Person Id that needs to be set to the account context. (Optional).
wssTargetURL	Relative URL to the target page to navigate. If not provided, then CSR will be navigated to CSR landing page.
wssPaymentAmount	Payment Amount to process onetime payment. If not provided the default payment Amount of 0.0 will be used.
wssCurrencySymbol	Currency Symbol to be used to process onetime payment. If not provided the default payment currency supported by locale will be used.
wssSaId	Sa Id to retrieve Scalar Meter Reads. (*Required) only for Add Scalar Read. If not provided, no data will be displayed.
wssBillId	Bill Id to be used for View Bill. If not provided, the default bill for the account will be loaded.

The following table provides sample URLs that can be used with the OTB OUCSS portal solution.

Note: Query parameters should be replaced with parameters appropriate to the environment.

Module	URL Pattern
Dashboard	http://host:port/webcenter/portal/system/CSRView?wssAccountId=1234567890&wssPersonId=1234567890&wssTargetURL=/portal/oucss/dashboard
View Bill	http://host:port/webcenter/portal/system/CSRView?wssAccountId=1234567890&wssTargetURL=/portal/oucss/viewbill
Make Payment	http://host:port/webcenter/portal/system/CSRView?wssAccountId=1234567890&wssTargetURL=/portal/oucss/makepayment&wssPaymentAmount=100.00

Scalar Read	http://host:port/webcenter/portal/system/CSRView?wssAccountId=1234567890&wssSaId=123456789&wssTargetURL=/portal/oracle/webcenter/page/scopedMD/sd1fe79b8_8f45_49bf_846b_059b446b5239/Paged0b93dd6_1cd8_4fbb_add1_cf0783c5f9cc.jspx

Customer Search

The Customer Search page is accessible only to OUCSS Administrators and CSRs. This search page is accessible from the **Customer Search** page in OUCSS Admin Portal.

Using the Customer Search page, a CSR and Admin can search for a registered user, or an Account in an edge application, or a User in an edge application.

OUCSS User Search

The user search allows the CSR to search for a user based on the following search criteria

- User Id
- First Name
- Last Name
- Email Address

The list of users will is rendered based on the search criteria in Search Results taskflow. The Administrator can now go and view all the user pages by clicking the **View User Pages** button for a particular user in the user list. This will set the current selected user from the user list in the context and the Administrator user will be able to see the user pages and carry out any action for that user.

Account Search by Account Id

CSR can search an Account using Account Id. The search results are queried against the Edge Application and all results matching the account id pattern are displayed in the Search Results taskflows. CSR then can click on View Pages against the Account Id from the Search result to view Details about the Account.

Name and Address Search

Accounts can also be searched by Name and Address associated with the account in Edge Application. This feature comes handy when the Customer is not aware of their Account Id. The name and address search allows the CSR to search for the account based on following search criteria

- Name
- Address
- City
- Postal

Search results matching the Name or Address are displayed in the Search Results taskflow. CSR then can click on View Pages against the Account Id from the Search result to view Details about the Account.

Access Roles

Open this Access page in OUCSS **Admin Portal**.

The **Access Role Code** uniquely identifies this access role.

Important! If you introduce new access roles, you must prefix the code with CM. If you do not do this, there is a possibility that a future release of the application could introduce a new message with the name you allocated.

Note: ACCOUNT_HOLDER and GUEST access roles are part of the seed data delivered with CSS. If access roles are managed in CCB, then add the CCB sourced access roles manually to the CSS database. For details, see the Release Notes.

Status indicates if an access role is **Active** or **Inactive**.

Description is the text of the message that appears on the various transactions in the system. Note, the access role's description can be overridden by specifying a **Description Override.**

Help and **Help Override** are reserved for future use.

Labels

Open the Labels page in **OUCSS Admin Portal**.

Some fields on this screen are protected as only the Oracle Utilities Product Development group may change them.

Label Code uniquely identifies this label.

Important! If you introduce new fields, you must prefix the code with **CM**. If you do not do this, there is a possibility that a future release of the application could introduce a new label with the name you allocated.

Description contains the text of the label. This is the text that appears on the various screens on which the label is displayed. Note, the label's description can be overridden by specifying a **Description Override**.

Help and **Help Override** are reserved for future use.

Translatable and **Translatable Context** are used if your OUCSS implementation supports multiple languages.

Labels in Oracle Utilities Customer Care and Billing

Enrolling Account process uses the verification questions defined in CCB. For prefetch enrollement process labeles for these verification fields are stored in CCB application.

To load these labels from CCB into OUCSS, go to **Labels** page from OUCSS Admin Portal. From the Actions menu dropdown select Reload Labels. After a successful reload, restart the server to load the newly added labels to resource bundle.

Note: Refer to the Self-Service Integration Master Configuration chapter for configuration information.

For more information on configuring self-service integration master configuration, see the embedded help provided in Oracle Utilities Customer Care and Billing.

Language

A locale exists for every language spoken by your users. The system uses this code to supply information to users in their respective language. Open **Language** page in OUCSS **Admin Portal**.

The system provides support for multiple languages in a single environment. Users can use the system in their preferred language, as long as a translation into that language has been provided. By default, users see the system in their default language, which is defined on their browser (e.g., Internet Explorer, Firefox) locale configuration. The browser locale may specify only a language code or both language and country code. The system verifies only the language code within the locale against the supported languages configured on this page. If a match is found, then that language is used to supply information to the user. Once the initial verification and language determination is completed, the system uses both the language code and country code (if available) to peform formatting of the content (e.g., Date). For example, the date format

can be different for different countries even though the langeage is same. The system will use the country code along with the language code to properly format the date.

Locale is a string that uniquely identifies the ISO language code in lower case.

Display Direction indicates if this language is written **Left to Right** or **Right to Left**.

Use the **Supported** checkbox to indicates whether or not the language is currently supported in the system.

Edge Application

Open Edge Application page in OUCSS Admin Portal.

Edge Application Code uniquely identifies this edge application.

Important! If you introduce new edge applications, you must prefix the code with CM. If you do not do this, there is a possibility that a future release of the application could introduce a new edge application with the name you allocated.

Server Reference Not used.

Description is the text of the message that appears on the various transactions in the system. Note, the edge application's description can be overridden by specifying a **Description Override.**

Help and **Help Override** are reserved for future use.

The grid contains the keys used to access this edge application. You can define up to 5 keys for each edge application.

Key Field Name uniquely identifies the key.

XML Field Name is the xml tag used to identify this key.

Line of Business

Open **Line of Business** page in **OUSS Admin Portal**.

Line of Business Code uniquely identifies this Line of Business.

Important! If you introduce new lines of business, you must prefix the code with CM. If you do not do this, there is a possibility that a future release of the application could introduce a new line of business with the name you allocated.

Description is the text of the message that appears on the various transactions in the system. Note, the line of business' description can be overridden by specifying a **Description Override.**

Help and **Help Override** are reserved for future use.

Edge Application is the owner of this line of business. All accounts associated with this line of business are maintained within this edge application.

Status indicates if a line of business is **Active** or **Inactive**.

Lookup

Open **Lookup** page in **OUCSS Admin Portal**.

Lookup Code is the unique name of the field whose lookup values are maintained in the grid.

Important! If you introduce new lookups, you must prefix the code with **CM**. If you do not do this, there is a possibility that a future release of the application could introduce a new lookup with the name you allocated.

Customizable check box indicates whether you are allowed to add valid values for a lookup field whose owner is not **Customer Modification**.

Description describes the lookup. Note, the lookup's description can be overridden by specifying a **Description Override**.

Help and **Help Override** are reserved for future use.

In this release, there are 3 sources of where lookups are coming from.

- Lookups defined in Oracle Utilities Customer Self-Service
- Lookups defined in Oracle Utilities Customer Care and Billing
- Lookups defined in Oracle Utilities Meter Data Management

The grid contains the values for a specific lookup.

Lookup Value is the unique identifier. If you add a new value, it must begin with a **CM** (in order to allow future upgrades to differentiate between your implementation-specific values and base-package values).

Status indicates if the value is **Active** or **Inactive**. The system does not allow **Inactive** values to be used (the reason we allow Inactive values is to support historical data that references a value that is no longer valid).

Attached Data is additional information associated with a lookup value.

Description is the name of the lookup value that appears on the various transactions in the system. Note, the lookup value's description can be overridden by specifying an **Override Description**.

Help and **Help Override** are reserved for future use.

Portal Resource Management (Lookups)

There are Portal Resource (Site Template, Navigation, Resource Catalog and Login Landing Page) configurations based on context/lob and device type. Implementation can update these entries and configure the behavior for existing or new LOB based on their requirement.

Site Template

Portal Site Templates are configured using the Lookup Code PORTAL_SITE_TEMPLATE (desktop) and MOB_B_ PORTAL_SITE_TEMPLATE (mobile). By default Site Template is defined for Public, Residential and Commercial LOBs. The relative URL to the site template is stored in Resources table.

Updating Default Site Template

To update the default Site Template for an existing LOB, update the "URL Override" column of the corresponding resource with new Site Template Internal Id using the Resource page in OUCSS Admin Portal.

Adding New Site Template

To add a new Site Template for a new LOB:

1. Go to Resource page in OUCSS Admin Portal.
2. Add a new Resource of WXURL type with the Site Template URL.
3. Go to Lookup page in OUCSS Admin Portal.
4. From the List of lookups, select the Lookup with Code PORTAL_SITE_TEMPLATE. For Mobile Templates, select MOB_B_PORTAL_SITE_TEMPLATE.
5. In the Lookup Value table, add an entry with Lookup value matching the LOB Code of the new LOB. Select the Resource added in Step 2.
6. Save the Changes and then select Flush Cache from Action menu.

Resource Catalog

Portal Resource Catalog is configured using the Lookup Code PORTAL_CATALOG and MOB_B_PORTAL_CATALOG. By default OUCSS Portal Catalog is defined for Residential and Commercial LOBs. The Internal Id of the Resource Catalog is stored as Resources.

Updating Default Catalog

To override the default catalog for an existing LOB, update the “URL Override” column of the corresponding resource with new Catalog’s Internal Id using Resources page in OUCSS Admin Portal.

Adding New Catalog

To add a catalog for a newly added LOB:

1. Go to Resource page in OUCSS Admin Portal.
2. Add a new Resource of WXURL type with the Catalog’s Internal Id. Save the changes.
3. Go to Lookup page in OUCSS Admin Portal.
4. From the List of lookups, select the Lookup with Code PORTAL_CATALOG or MOB_B_PORTAL_CATALOG for mobile.
5. In the Lookup Value table, add an entry with Lookup value matching the LOB Code of the new LOB. Select the Resource added in Step 2.
6. Save the Changes and then select Flush Cache from Action menu.

Skin

Skin for Portal are configured using the Lookup Code PORTAL_SKIN or MOB_B_PORTAL_SKIN. By default Skins are defined for Public users and Residential and Commercial LOBs. Skin is configured using the skinFamily name that is defined when creating a new skin.

Updating Default Site Template

To update the default Site Template for an existing LOB, update the “URL Override” column of the corresponding resource with new skinFamily value using Resources page in OUCSS Admin Portal.

Adding New Site Template

To add a new Site Template for a new LOB:

1. Go to Resource page in OUCSS Admin Portal.
2. Add a new Resource of WXURL type with the Skin Family name in URL column.
3. Go to Lookup page in OUCSS Admin Portal.
4. From the List of lookups, select the Lookup with Code PORTAL_SKIN or MOB_B_PORTAL_SKIN.
5. In the Lookup Value table, add an entry with Lookup value matching the LOB Code of the new LOB. Select the Resource added in Step 2.
6. Save the Changes and then select Flush Cache from Action menu.

Login Landing Page

Portal landing page is configured using the Lookup Code PORTAL_LANDING_PAGE or MOB_B_ PORTAL_LANDING_PAGE (mobile). By default landing pages are defined for Residential and Commercial LOBs. Landing pages can also be customized differently for customers enrolled to single Account and customers with multiple enrolled accounts per LOB. The relative URL of the landing page is stored as Resources.

Updating the Default Landing Page

To override the default catalog for an existing LOB, update the "URL Override" column of the corresponding resource with new Catalog URL using Resources page in OUCSS Admin Portal.

Adding a New Landing Page

To add a landing page for a newly-added LOB:

1. Go to the Resource page in OUCSS Admin Portal.
2. Add a new Resource of WXURL type with the Catalog URL. Add 2 Resources to handle both Single Account and Multi-Account scenario.

   For Single Account landing page resource suffix the RESOURCE_CD with "_SA". Save the changes.
3. Go to the Lookup page in OUCSS Admin Portal.
4. From the List of lookups, select the Lookup with Code PORTAL_LANDING_PAGE or MOB_B_PORTAL_LANDING_PAGE for Mobile.
5. In the Lookup Value table, add two entries with Lookup value matching the LOB Code of the new LOB. Select the Resource added in Step 2.
6. Save the Changes and then select Flush Cache from Action menu.

The CSR/Admin Landing Page

OUCSS CSR and OUCSS Admin will navigate to Customer Search Page after Login. Implementation can change the default landing page by adding a resource with code PORTAL_CSRVIEW_LANDING_PAGE.

Lookups in Oracle Utilities Customer Care and Billing

There are several configurations originating from CCB that are stored in OUCSS as lookups. They are:

- Valid Payment Types (e.g., Credit Card, Checking, Savings)
- Valid Card Types (e.g., Visa, American Express, Discover)
- Valid Bill Route Types (e.g., Postal, Email, Fax)
- Valid Phone Types (e.g., Mobile Phone, Home Phone, etc.)

To load these labels from CCB into OUCSS automatically, go to Admin Menu, Lookups. On the Actions menu dropdown, select Reload Lookups. After a successful reload, you will need to flush the cache by selecting Flush Cache from the Actions menu dropdown.

Notes:

- To configure the valid Payment Type and Card Type lookups, refer to the embedded helpprovided in the Self-Service Integration Master Configuration defined in the CCB system. For the Bill Route Type and Phone Type lookups, these will be part of your regular CCB configuration. Refer to the Oracle Utilities Customer Care and Billing Documentation.

- In order to add an additional Form Type to the application, a new lookup value need to be added to FORMTYPE lookup with Attached data matching CCB C1_SS_CATEGORY lookup value defined for the new form type in CCB.

Lookups in Oracle Utilities Meter Data Management

There is one configuration originating from MDM that is stored in OUCSS as a lookup. It is the valid Time Of Use codes and their associated color for graphical display (e.g., On peak data will be seen as red, Off peak data will be seen as blue, etc.).

You must define in OUCSS all the supported Time Of Use codes in MDM. For each Time Of Use codes, you must associate it with a unique color defined in hexadecimal format (e.g., #BB7D3E, #602040, etc.) in the **Attached Data** field.

A sample data set for TOU_CODES in SS_LOOKUP_VAL table:

LOOKUP_CD	LOOKUP_VALUE	ATTACHED_DATA
TOU_CODES	OFF	#C1DFAD
TOU_CODES	OFFSUM	#ABC9DD
TOU_CODES	OFFSW	#FF8080
TOU_CODES	OFFWIN	#993366
TOU_CODES	OFFWW	#CCFFCC
TOU_CODES	ON	#FF6600
TOU_CODES	ONSUM	#0066CC
TOU_CODES	ONSW	#339966
TOU_CODES	ONSWIN	#00FF00
TOU_CODES	ONSWW	#993300
TOU_CODES	PEAK	#660033
TOU_CODES	SH	#00CCFF

The Usage Aggregate and Usage Comparison modules in commercial context uses the lookups (USAGE_PERIOD, USAGE_OVERLAY, USAGE_NAVIGATION) defined in the lookup table to pass the corresponding MDM values for display modes, Overlay modes and navigation (Previous, Next).

Offer Set

Offers means both Rates and Promotions. There are three styles of formats available for displaying Offers:

- Banner format
- Standard format
- Column format

This provides the company with a choice about how (and where) to display this information to the customer.

Banner

This format will be used when a single promotion is to be highlighted using a wide image (a standard web format banner.

- This is restricted to a single entry.
- The most important information for this taskflow/portlet is the image. It is assumed that the image should encourage further investigation if it sounds appealing.
- Clicking the image/link will take the customer to a page for further information/action.

Standard

This format will be used to display offers in the standard format (rows and 3 columns). First column displays the image or the title to capture the offer. Second column displays a link for more information and the third column displays a detailed description about the offer.

- This format allow multiple entries to be displayed.
- The most important information for this taskflow/portlet is the promotion title or a picture. The text or picture should be a draw for the customer to request more information.
- Clicking on image or title will take the customer to a page for further information/action.

Column

This format will be used when a narrow column running down the page is desired. It is recommended that the column should normally be configured to the right of the screen.

- This format allow for multiple entries displayed in column format.
- The most important information for this taskflow/portlet is the name or image of the promotion. It is assumed that the name/image should encourage further investigation if it sounds appealing.
- Clicking on image or title will take the customer to a page for further information/action.

There are six taskflows/portlets implemented to cater to all three Offers formats. The taskflows are:

Promotions

- Banner Promotion
- Standard Promotion
- Column Promotion

Rates

- Banner Rates
- Standard Rates
- Column Rates

The input parameter to these offers is Offer Set Code and Locale. An Offer Set Code captures all the details like Type of Offer, Format and required values from the database. The locale is automatically picked up from the locale of the User. Customers can drop any of the offers taskflows/portlets on to a page and configure the Offer Set Code to a new or existing value to display the intended information.

OUCSS Inbound Services

The following applications are deployed as part of the OUCSS Inbound Services application:

- OTB Offers Services
- Account Enroll Service
- OUCSS REST Services

Offer Service (Web Service)

Offer Service is installed by default as part of installation of OUCSSInbound application on the same managed server as the taskflows/portlets. The service is implemented using ADF BC and exposed as WebService. This service uses the Offer Set Code and Locale to fetch the required data from the database (Offers tables in OUCSS schema). The taskflows uses this data to display the data in required format

Check the Offers taskflow/portlet in Admin module for more information on the format of the Offer database table and their relations in order to add/delete/modify data for the service. If you are using the out-of-box solution, you can use the Offers Admin taskflow/portlet to add/delete/modify entries for the offers you want to maintain.

To allow users to reuse the Offers taskflows, the taskflows are built using Web Service. Customers can build their own Web Service with complex logic to display offers. The new web service should match the WSDL of the out-of-the-box web service for the offers to render properly.

To Configure Offers with Your Own Web Service

The following procedure describes how to implement and use your own web service. Before starting, implement custom Offers Web Service and make sure the WSDLs are identical to the current service.

To update the Web Service used by Offers in OUCSS:

1. Login to Enterprise Manager.
2. Click on the deployed application (implementing OUCSS solution).
3. From the Application Deployment menu select ADF > Configure ADF Connections.
4. Select the Offers Service connection and click Edit.
5. Update the WSDL and Service Name of the connection to a new service.
6. Click OK, then open the Advanced Connection Configurations menu and select the port to update the End Point URL.
7. Select the Configurations tab.
8. Update the Endpoint Address in the General section, and then click Apply to commit the changes.
9. Click Apply again to commit the changes to the Offer Service connection (restart is not required).

Account Enrollment Web Service

This web service provides the following operations.

Operations

ManageUserService

If the user already exists in LDAP but not in CSS, they can be added to SS_USER table using this operation. This operation can be used in conjunction with external registration process to add the user in CSS.

XML Schema

```
<schema>
<input type=group>
    <action/>
    <key1/>
    <key2/>
    <key3/>
    <key4/>
    <key5/>
    <emailAddress/>
    <webUserId/>
    <ipAddress/>
    <cssUser/>
</input>
<mainData type="group">
    <customers type=”list”>
        <email/>
        <accessRoleCd/>
        <lobId/>
     </customers>
     <accounts type=”list”>
        <key1/>
        <key2/>
        <key3/>
        <key4/>
        <key5/>
     </accounts>
</mainData>
<results>
    <status/>
    <errors type=”list”>
         <email/>
         <account>
             <key1/>
             <key2/>
             <key3/>
             <key4/>
             <key5/>
         </account>
         <code/>
         <message/>
   </errors>
   </results>
</schema>
```

Input

- action – Action of the service. Currently valid value is ADD.
- userId – User Id of the user being added.
- emailAddress – Email of the CSS Registered

- firstName – First name of the user.
- lastName – Last Name of the user.
- sendEmail – Flag to send an email after successful addition.
- ipAddress – For tracking purposes. Optional.
- locale – Locale to localize the statusMessage.
- user_tz – Time zone of the user. Optional and currently not used.

Output

- Status – SUCCESS / ERROR / WARNING
- genUserId – Generated user Id upon successful addition.
- statusMessage – Message related to returned status.

Processing

- The service is secured and can be invoked only by users from WSSAdminGroup. If the user invoking this service does not belong to the group, then SOAP Fault is thrown.
- Validate that either userId is provided in input is valid for registration in OUCSS.
- Verify if the email address is registered to a user or pending email address. If email already used, return status as ERROR with statusMessage populated with the error message.
- Add the user using OUCSSUserRegistrationService.addUser method.
- If success, return the status and genUserId.
- If error occurs, return status as ERROR with statusMessage populated with the error message.
- Send the email if the sendEmail attribute is set to true.
- If sending email fails, then return status as WARNING with statusMessage populated with send email error message.

AccountEnrollService

Enroll multiple users to a set of accounts. Users may or may not be registered in the Self-Service application.

If the user is already registered in the Self-Service application, the user will be given access to the accounts and an email will be sent out to the user. If the user is not registered in the Self-Service application, an email will be sent to the user. The user will have to register with the Self-Service application in order to view the accounts. The process will be similar to the Invite process for unregistered users.

XML Schema

```
<schema>
<input type=group>
    <action/>
    <key1/>
    <key2/>
    <key3/>
    <key4/>
    <key5/>
    <emailAddress/>
    <webUserId/>
    <ipAddress/>
```

```
    <cssUser/>
</input>
<mainData type="group">
    <customers type=”list”>
        <email/>
        <accessRoleCd/>
        <lobId/>
     </customers>
     <accounts type=”list”>
        <key1/>
        <key2/>
        <key3/>
        <key4/>
        <key5/>
     </accounts>
</mainData>
<results>
    <status/>
    <errors type=”list”>
         <email/>
         <account>
             <key1/>
             <key2/>
             <key3/>
             <key4/>
             <key5/>
         </account>
         <code/>
         <message/>
   </errors>
   </results>
</schema>
```

Input

- Action – Not Used
- Key 1 –Key5 – Not Used
- Email Address – Email of the CSS Registered user running the service/inviting/enrolling other users
- webUserId – CSS Login of the user running the service/inviting/enrolling other users
- ipAddress – Not Used
- cssUser – Not Used

MainData

- customers - List of users to be enrolled/ invited
- customers/email – Email of the user to be enrolled / invited
- customers/ accessRoleCd - Access Role given to the user and all the accounts in the request
- customers/lobId - Line of Business Id for which the access is given to the user and account
- accounts - List of accounts to be enrolled for each user in customers list
- accounts/key1 to key5 - Identifiers for the account

Output

- Status – SUCCESS / ERROR / WARNING
- errors - this is a list of the errors
- email – email of the user corresponding to the error.
- Account/key1 to key5 – Identifiers for the account corresponding to the error
- code – Error code / Message Code defined in OUCSS
- message – Error message / Detailed Message defined in OUCSS

Processing

- Validate that either emailAddress (or) webUserId is provided in input section and it corresponds to a valid registered user in OUCSS.
- Validate that at least one customer and account are provided.
- Validate that accessRoleCd and lobId are valid.
- Verify if the email address is registered to a user. If registered, enroll the user to all the accounts in the request. Add entries to the User Enrollment Table for each entry in the accounts list.
  - If the user is already enrolled to the account in the request, add the error to the errors list and continue processing.
  - If the email address is not registered, add the email address to the User table and entries for each account to User Enrollment Table
- Send an email to the indicated address(es) informing the customer(s) of the enrollment / invite.
- If any error occurs, add it to the errors list in the output and continue processing.

WS Security

The Account Enroll service is secured using OWSM server policy oracle/multi_token_rest_service_policy. This policy enforces one of the following authentication policies, based on the token sent by the client:

- HTTP Basic - Extracts username and password credentials from the HTTP header.
- SAML 2.0 Bearer token in the HTTP header - Extracts SAML 2.0 Bearer assertion in the HTTP header.
- HTTP OAM security - Verifies that the OAM agent has authenticated user and establishes identity.
- SPNEGO over HTTP security - Extracts Simple and Protected GSSAPI Negotiation Mechanism (SPNEGO) Kerberos token from the HTTP header.

How to Invoke the Web Service

- Account Enrollment web service will be deployed during installation as part of a seperate application called ‘OUCSSInboundServices’.
- The URL will be http://<<server>>:<<port>>/ <<context>>/AccountEnrollService?wsdl where server and port are as per the information provided in the deployTarget section for oucssInbound in InstallProperties.xml
- Provide the security credentials.
- For example, use the HTTP Authorization header with base 64 encoded username/password.
- Select the operation (ManageUserService or AccountEnrollService) and populate the request.

- Any errors will be returned in the output.

Security

Taskflow/Portlet Security Overview

The Tier 2 security controls access to the links and buttons on the taskflows/portlets. The access rights for a logged in user are loaded from the database based on the configuration.

Taskflow/Portlet security restricts access to its transactions as follows:

- Each taskflow/portlet must be defined in Portlets table with list of actions allowed for this portlet
- Available actions should be defined for each Line Of Business and Access Role. Every user has each Line Of Business and Access Role
- Specific user interface components (buttons, links) can be hidden or visible based on the access role.

When you grant an Access Role access to a portal, you must also define the permitted action.

For example, you may indicate a Line Of Business/Access Role has inquire-only access to a taskflow/portlet , whereas another role has also change privilege to the same taskflow/portlet.

How to Configure Security Settings

In order to add or change security settings, the user must login to the system as administrator.

Changes in security for a specific user or group of users will be visible in the system only after the user logs out and logs in again.

User

The link between Line of Business/Access Role and User is done during registration process.

A new link between User and Line of Business/Access Role is created if a user is invited as guest. If the guess access is revoked, this link is removed.

Security

Go to the Security page in OUCSS Admin Portal.

For each combination of Line of Business and Access Role, specify portals that a user can access and list of actions a user can perform.

Field Level Security

Specific user interface components (buttons, links) can be hidden or visible based on the access role.

The Base Bean contains Java methods to check for required permission. The Java methods are isReadPermission, isUpdatePermission and isAddPermission to check for Read/View, Update and Add permissions repectively.

For example, to show or hide the “Update” button on the View Mailing address taskflow/portlet the following code needs to be added for the Update button. The rendered property of the button is set using the isUpdatePermission method (in EL this corresponds to updatePermission).

```
<af:button text="#{ssBundle.ACCOUNT_UPDATE_LBL}"
           partialSubmit="true" id="amupclnk"
           inlineStyle="white-space:nowrap"
           disabled="#{pageFlowScope.accountAddressManagedBean.updatePageURL eq null}"
          rendered="#{pageFlowScope.accountAddressManagedBean.updatePermission}"/>
```

Portlets

Open the **Portlets** page in **OUCSS Admin Portal**.

Portlet Code uniquely identifies this portlet.

Important! If you introduce new portlets, you must prefix the code with CM. If you do not do this, there is a possibility that a future release of the application could introduce a new portlet with the name you allocated.

Description is the text of the message that appears on the various transactions in the system. Note, the portlet’s description can be overridden by specifying a **Description Override.**

Help and **Help Override** are reserved for future use.

The grid contains the values for the valid actions allowed on a specific portlet.

Action Value is a dropdown of the allowed actions. The values for **Action Value** are defined in the base lookup **ACTION_FLG**.

Message

Open the **Message** page in **OUSS Admin Portal**.

Message Code uniquely identifies this message.

Important! If you introduce new messages, you must prefix the code with CM. If you do not do this, there is a possibility that a future release of the application could introduce a new message with the name you allocated.

Message Category identifies if the message is any of the following categories:

- Error
- Informational
- Warning
- Fatal
- Email

Description is the text of the message that appears on the various transactions in the system. Note, the message’s description can be overridden by specifying a **Description Override.**

Help and **Help Override** are reserved for future use.

Trains

Open the **Trains** page using **OUSS Admin Portal.**

- **Train Code** – A unique name of the train.It is a required field that is enabled on create.

- **Description** – Description of the train.
- **Description Override** – optional field that overrides the description field if populated.
- **Help and Help Override**- are reserved for future use
- **Owner Flag** – indicates the owner of the train.
- **Number of Screens** – indicates the number of active screens in the train.

Train Portlet Maintenance Grid

This displays all the portlets/trains stops available for the a specific train.

- **Portlet Code** – a required field . On create, the user can select a portlet code from the dropdown list.The field is read only on save. Once the user selects a portlet code, the Load Actions button loads the applicable action values of the portlet.
- **Action Value** – This field displays all applicable actions for the portlet code.
- **Status** – Indicates if the portlet is Active or Inactive.If Active, portlet will be rendered in the train UI, if inactive, it will not be rendered.
- **Sequence Number** – This dictates the sequence of the train stops.
- **Sequence Number Override** – This overrides the Sequence Number value.

Configuration Properties (System)

Configuration properties can be adjusted using **System** page in **OUCSS Admin Portal**. After changes have been made through user interface the system must be restarted for some changes to take effect. Alternatively, configuration properties can be configured directly in the table SS_CONFIGURATION. System restart is also necessary in this case.

The SS_CONFIGURATION table contains properties which serve as system defaults. It also contains flags to enable email-sending-related properties.

The following configuration properties must be configured prior to use of the system:

Property	Description	Default Value
current.oucss.version	This property tracks the current OUCSS version. Please do not change this property.	2.2.0.0
installation.owner.flag	This is the current installation owner flag. When the product will be shipped this property should be CM	CM
oucss.default.locale	Default Locale of OUCSS Application	en
edgeapplication.ccb.datasource	This is the edge application code (e.g., C1, M1, etc.) from where the data is pulled.	C1
lookups.webservice.status	Status of the Lookup Value inserted using this service. Only allowed values are ACTIVE, INACTIVE.	ACTIVE
validate.regex.email	Regular expression to validate email address in OUCSS UI.	^(?:[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}|)$

validate.regex.username	Regular expression to validate a valid user name that can be used when registering.	[a-zA-Z0-9_]*
validate.regex.password	Regular expression to validate the password pattern that can be used when registering or changing user password.	[a-zA-Z0-9_]*
validate.length.password.min	Integer value for minimum length of the password that should accept when registering or changing password.	6
validate.length.password.max	Integer value to allow maximum length of the password when registering or changing password.	12
mail.session.jndi.name	JNDI of Mail Session to send emails.	mail/OUCSS
webcenter.register.url	This will be used in the registration email which is sent to the user with the registration key and a URL. User shall click on this link in the email to complete the OUCSS registration.	http://<PortalHost>:<PortalPort>/webcenter
webcenter.login.url	This property will be used in email messages to send the user the login URL.	http://<PortalHost>:<PortalPort>/webcenter/portal/public/login
outage.map.color.theme.buckets	Match the number to number of colors configured in outage.map.color.theme.colors property	4
outage.table.page.size	Number of records that can be displayed at a time on the Outage Table screen	10
outage.map.base.map	This property is used to setup BASE MAP configured in the MapViewer. This property will be used to display the Outage Summary Map.	Value configured in InstallProperties.xml
outage.map.color.theme	This property is used to color code Outages in Outage Summary.	Value configured in InstallProperties.xml
outage.map.color.theme.loc	This property is the Area Column from the Color Theme configured above.	Value configured in InstallProperties.xml
outage.map.srid	This property is SRID of the Coordinate system used by Base Map and Theme configured above.	Value configured in InstallProperties.xml /8307
outage.map.color.theme.colors	This property is configured to set the number of buckets to aggregate the Outages as well as the respective color of each bucket.	#00FF00;#EEEE00;#FF7F00; #FF0000 (Green, Yellow, Orange and Red)
outage.area.nms.config	This property represents the Area configured to aggregate Outages in NMS. This will affect the color theme and/or theme location column configured above.	ZIP (other valid values are CITY and COUNTY).

rollback.on.failed.email	This property if set to 'Y' will roll back Enroll or Invite to an Account if the sending the email fails. Set it to 'N' if mail session is not configured or you want to commit Enroll and Invite even if sending email fails.	Y
account.list.max.rows	This property specifies the number of associated accounts that will be displayed without enabling search functionality. Once the number of accounts exceeds this limit, search will be enabled.	10
account.list.page.size	This property controls the page size or the number of accounts per page presented in the Account List. If the number of associated accounts exceeds the value of this parameter, paging will be enabled.	10
max.rows.premise.search	Set this property to the max number of rows that can result in a Premise search. The system will show error if the premise search in Start Service and New Customer service exceeds the value set here.	10
default.customerclass.parm	Property to set default Customer Class configured in CCB for Start/Stop/Transfer Services.	R
default.personbusiness.parm	This property sets the default Person Business configured in CCB for Start/Stop and Transfer service.	P
default.newcust.requestmode	This property sets the default Request Mode configured in CCB for New Customer Service.	C1ST
oucss.reset.password.pattern	Set of characters to be used to generate a random password using the Forgot Password functionality.	0123456789abcdefghijklmnopqrstuvwxyz$#*_ABCDEFGHIJKLMNOPQRSTUVWXYZ
default.enroll.role	Access role to be used when a user enrolls to an account. If the Access role configured in this property is not found, ACCOUNT_HOLDER will be used as default.	ACCOUNT_HOLDER
csr.account.access	Access role to be used for accounts when CSR views the account after searching it using Customer Search screen.	ACCOUNT_HOLDER
csr.search.results.fetchsize	Number of accounts to be fetched when searching using CSR search screen.	300
include.topTag.usageDownload	Property to control if the top tag should be included when downloading Usage Data in XML format. If set to false, the top tag will be omitted in the file.	true
outage.map.base.startingX	Property to set the default latitude location to center the Outage Map.	-81.70

outage.map.base.startingY	Property to set the default longitude location to center the Outage Map.	40.69
outage.map.base.zoom	Property to set default Map Zoom level when no data is found.	4
attachment.list.max.row	Property to set the maximum rows to be allowed for attachment.	5
forms.list.page.size	Property to set the maximum number of issues to be displayed in forms list page.	10
oucss.debug.enable	Property to enable showing of debug message along with error messages on screen.	false
def.day.mode.range.in.month	Property used in Usage Detail to set the default range for Day view mode. Default value is 3 months.	3
def.hour.mode.range.in.days	Property used in Usage Detail to set the default range for Hour view mode. Default value is 7 months.	
def.month.mode.range.in.year	Property used in Usage Detail to set the default range for Month view mode. Default value is 1 year.	1
max.day.mode.range.in.years	Property used in Usage Detail to set the maximum range for Day view mode. Default value is 1 year	1
max.hour.mode.range.in.days	Property used in Usage Detail to set the maximum range for Hour view mode. Default value is 30 days.	30
max.month.mode.range.in.years	Property used in Usage Detail to set the maximum range for Month view mode. Default value is 4 years.	4
scalar.usage.graph.color	Property to control the default color of Scalar Usage Detail graph	#660033
enable.email.validation	Property to control if the Email Validatation is enabled. Email Validation is enabled by default. Please visit the section to learn more about this functionality.	Y
enable.html.email	Property to control is Emails are sent in HTML formats using templates. This is enabled by default.	Y
oucss.attachment.max.size	Property to control the max memory (in bytes) allowed when uploading a file.	5243000
oucss.link.ccb.person	Property to control whether CCB-CSS User Linking is enabled	Y

address.validation.enabled	Property that indicates if the address validation for Account Address Information will take effect or not. Possible values are True or False.	False
enable.email.userId	Property to enable using email address as user id. Possible values are Y and N	N
oucss.context.selection	Property to enable Consolidated Account. If the value is set to A the Portal has no context. If set to S, then user get to choose a context at the time of login.	A
oucss.deviceagent.enable	Flag to enable Mobile Responsive layout on mobile phones. Set it to false to disable this feature.	true
Commercial Properties		
oucss.comm.context.limit	Property to set the number of accounts that can be selected in Business context to view Multi account taskflows	10
oucss.comm.context.pagesize	Property to set the number of accounts to be displayed in Business context without scroll bar. Accounts more than the set property will be displayed with a scroll bar.	10
set.account.page.size	Property to control the page size of the number of accounts per page presented in the Set. If the number of associated accounts exceeds the value of this parameter, paging will be enabled.	10
Notification Properties		
notification.list.page.size	Property to control the page size of notification. If the number of notifications associated with accounts exceeds the value of this parameter, paging will be enabled.	15
notif.list.chars.preview	Property to set the length of the notification before a preview mode is enabled.	140
notif.validate.regex.phone	Property to set the regular expression to validate the phone number for Notification preferences.	^(?:\+?1[-.]?)?\(?(\d{3})\)?[-.]?(\d{3})[-.]?(\d{4})$
oucss.notification.installed	Property to indicate if Notification Center is installed. Note: This property has been deprecated in release 2.2.0.0 and is no longer used. Use oucss.notification.owner property instead.	false

oucss.notification.owner	Property to indicate if notifications and notification channels is owned by Notification center or CCB. Possible values are OUNC : when Notification Center owns the notification preferences and delivery channels, CCB : When CCB owns the notification preference and delivery channels CSS : "Remove Account" functionality on Account List screen will invoke UnEnroll Service, to remove notification preference for original bill notification types Blank(no value) : "Remove Account" functionality on Account List screen will invoke WXBillNotifyPreferenceService to remove notification preference for original bill notification types (for backward compatibility)	OUNC

Resources

Open the **Resources** page using **OUCSS Admin Portal**.

Some fields on this screen are protected as only the Oracle Utilities Product Development group may change them.

Resource Code uniquely identifies this resource.

Important! If you introduce new resources, you must prefix the code with **CM**. If you do not do this, there is a possibility that a future release of the application could introduce a new resource with the name you allocated.

Resource Type defines what kind of resource is thisCurrently, the system uses resources of type WXIMG – images and of type WXURL – links.

Locator Type allows to better annotate what kind of URL is used to describe the resource. URL for resources can be defined as relative (WXREL) or absolute (WXABS).

Description contains the text of the label. This is the text that appears on the various screens on which the resource is displayed. Note, the resources's description can be overridden by specifying a **Description Override**.

Help and **Help Override** are reserved for future use.

Portal Resources

Section Portal Resource Management in this guide details management of Portal Resources per LOB. Site-wide resources related to OUCSS Portal can be configured using Resources page in OUCSS Admin Portal. By default, these resources are not created as part of product release. Implementation can choose to customize the Page Not Found, Un Authorized and Server Error pages using the System pages screen in WebCenter Administrator Console.

RESOURCE_CD	Purpose
PORTAL_PAGE_NOT_FOUND_PAGE	Not used in 2.2.0.0 with WebCenter Portal.
PORTAL_UNAUTHORIZED_PAGE	Not used in 2.2.0.0 with WebCenter Portal.
PORTAL_SERVER_ERROR_PAGE	Not used in 2.2.0.0 with WebCenter Portal.
PORTAL_LOGOUT_PAGE	Use this code to configure the Logout Page.

PORTAL_LOGIN_PAGE	Use this code to configure the Login Page.
PORTAL_CSRVIEW_LANDING_PAGE	Use this code to configure the Landing page for CSR Users.

Resources in Oracle Utilities Customer Care and Billing

In some cases CCB must return information about a particular resource to use, such as a link or an image for an alert. This is accomplished by configuring named resource in OUCSS in the **Resources** screen and providing the value of the Resource Code to the CCB team. 'PAY_ARRANGEMENT' and 'PLANNED_OUTAGE_URL' are examples of such configurations.

User

User registration and profile update requires validation through email if the configuration property 'enable.email.validation' is set to 'Y'. Additionally, user can receive validation email in HTML format when the configuration property 'enable.html.email' is also set to 'Y'.

Registration

User registration may require user to validate their email id based on the configuration property 'enable.email.validation'. Section Configuration Properties (System) describes steps to change the configuration property value. User needs to follow the procedure described below to register when email validation is set to 'Y':

- Click on **Register** from the top navigation.
- Enter User Name, First Name, Last Name, Email Address and Confirm Email Address.
- User name needs to be unique and email address should not be in use. Please correct any error reported by system while trying to register user.
- An email will be sent to the email address used during registration with the validation link. User needs to follow the link sent in the email to validate email address.
- In case user wants to modify email address before validation, he is permited to do so by re-registering using the same user id as used before and modifying the email address.
- User can register with same information multiple times until the email address is validated. This will send new email to user's mail box with new validation link. Other validation links sent in older emails will be invalidated. This feature helps user to register even when he loses original validation email.
- Registration validation page will prompt for User Name, Email Address, Password and Confirm Password. User name and email address should match the information entered during registration.

After successful registration followed by email validation user will be registered into the system. User may now login into the system using user name and password he used during registration process. System administrator may use the field EMAIL_CONFIRM_SENT_ON in the table SS_USER to determine how long the user registration is in pending state, and may choose to write database script to remove users pending registration for longer than certain period.

Update Profile

Profile update lets user change profile information once user is registered in the system. User may be required to validate their email based on the configuration property 'enable.email.validation'. Section Configuration Properties (System) describes steps to change the configuration property value. User needs to follow the procedure described below to update profile when email validation is set to 'Y':

- Login into the system and navigate to User Profile > Change Personal Information.

- User information will be pre-populated into the fields First Name, Last Name and Email Address. One or more of the fields could be updated to modify user profile.
- Any update to First name and Last name will be effective immediately. Any change to Email Address, however, will require validation. When email address is updated, an email will be sent to the new email address with the validation link. User needs to follow the link sent in the email to validate email address.
- User may click on 'Resend Confirm Email' to resend validation email to user's mail box. This will send new email to user's mail box with new validation link. Other validation links sent in older emails will be invalidated. This feature helps user to update email address even when he looses original validation email.
- Please note that the button 'Resend Confirm Email' will only appear when user has already updated email address which is pending validation.
- In case user wants to modify email address before validation, he is permited to do so by login into system with the same user id as used before and updating profile with new email address.
- Update profile validation page will prompt for User Name and Email Address. These fields should match the information entered during profile update.

Email Template (HTML)

Emails are being sent from several modules of OUCSS to inform users of certain events or notify them about certain actions pending on them. This functionality enables customer to send mail in plain text as well as HTML format. As most of the email clients support HTML, this feature enables customer to send mail which better represents customer branding and improves email look & feel.

Using HTML Email Template

Email will be sent in HTML format if the configuration property 'enable.html.email' is set to 'Y'. Section Configuration Properties (System) describes steps to change the configuration property value. Customer needs to follow the procedure described below to send mail with customized HTML mail templates:

- Customer may start with sample HTML templates shipped with OUCSS or may create their own templates from scratch. Sample HTML templates for email will be stored inside 'Shared' application in the package - Shared/Utilities/public_html/com/oracle/ss/shared/view/html
- The basic form of an HTML template would look like the following, where {0} represents a placeholder. Please follow this Section for a complete list of placeholders used in default HTML templates.

```
<html>
<body>
{0}
</body>
</html>
```

- To include custom HTML templates inside the 'extend.oucss.portal' and deploy, see the section "Deploying the extend.oucss.portal as a Shared Library" in the document, *Whitepaper: Customizing And Extending the OUCSS Custom Portal*.

Example: A user might create and choose the public_html\oracle\ugbu\ss\custom\pages\public\template package inside extend.oucss.portal to store custom HTML templates. (Note: Don't create the package inside WEB-INF or it won't be accessible from other applications.) The user would then navigate to the Resources page in OUCSS Admin Portal and update the Resource entry for the template with the relative path to the template in the URL Override field. For a complete list of Resources defined for HTML email templates, see Email Hyperlink Resources.

Email Template Resources

A Customer may choose to override default template path for the resources listed in the following table.

RESOURCE_CD	Purpose
REGISTER_EMAIL_TEMPLATE	Html template used in confirm email sent during Registration
REGISTER_EMAIL_NO_VALID_TEMPLATE	Html template used in confirm email sent during Registration without validation
UPDATE_PROFILE_EMAIL_TEMPLATE	Html template used in confirm email sent during Update
PASS_RESET_EMAIL_TEMPLATE	Html template used in to send new password after password reset
INVITE_EMAIL_TEMPLATE	Html template used in to construct email sent during Invite
SERVICE_INVITE_EMAIL_TEMPLATE	Html template used in to construct email sent during Service Invite
SET_INVITE_EMAIL_TEMPLATE	Html template used in to construct email sent during Set Invite

Sample Email Templates

A customer may find sample email templates in the paths listed in the following table.

Template	Path
Registration Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/RegisterHtmlEmailTemplate.html
Registration Template (No validation)	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/RegisterHtmlEmailNoValidTemplate.html
Profile Update Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/UpdateProfileHtmlEmailTemplate.html
Password Reset Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/PassResetHtmlEmailTemplate.html
Invite Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/InviteHtmlEmailTemplate.html
Service Invite Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/ServiceInviteHtmlEmailTemplate.html
Set Invite Template	Shared/Utilities/public_html/com/oracle/ss/shared/view/html/SetInviteHtmlEmailTemplate.html

Email Template Tokens

A customer should use the placeholders mentioned here when creating custom email templates.

Template	Token	Value
Registration Template	{0}	Email validation link for registration
Registration Template (No validation)	{0}	Email login link for registration
Profile Update Template	{0}	Email validation link for profile update
Password Reset Template	{0}	New Password
Invite Template	{0}	Account ID
	{1}	Link to Login URL

Service Invite Template	{0}	Account ID
	{1}	Link to Login URL
Set Invite Template	{0}	Account ID
	{1}	Link to Login URL

Chapter 3

Customer Care and Billing Configuration

Oracle Utilities Customer Care and Billing Configuration

To set up Oracle Utilities Customer Care and Billing for self-service implementation, the following must be configured:

- Self-Service Integration Master Configuration
- Self-Service Task Types
- Campaigns and Packages

Note: For more information on configuring and working with Oracle Utilities Customer Care and Billing, see the Oracle Utilities Customer Care and Billing user documentation.

Self Service Integration Master Configuration

Create a master configuration for self-service integration in Oracle Utilities Customer Care and Billing.

Navigation	Guideline
Admin Menu > Master Configuration	Create a master configuration to be used for self-service integration.

Note: For more information on configuring self-service integration master configuration, see the embedded help provided in Oracle Utilities Customer Care and Billing.

Self-Service Task Types

Create self-service task types for each self-service task in Oracle Utilities Customer Care and Billing.

Navigation	Guideline
Self-Service > Self-Service Task Type	Create self-service task types for each self-service task business object supported by your implementation.

Note: For more information on configuring self-service integration master configuration, see the embedded help provided in Oracle Utilities Customer Care and Billing.

Campaigns and Packages

If your implementation supports start/stop service requests via the self-service application and the base product start service criteria script is used, campaigns must be configured to handle the request. Setup a campaign for each customer class that may submit start/stop service requests via the self-service application.

In Forms processing, campaigns are used to prompt the self-service user for more information related to a form. If necessary, set up a campaign for each form that requires the gathering of additional information.

Navigation	Guideline
Sales & Marketing > Campaign	Configure your campaign(s) to support start service requests. This campaign should include the questions your implementation wishes to present to the end customer that will assist in package/SA selection.
Sales & Marketing > Package	Configure your packages for each campaign to support start service requests. These packages define the service agreement(s) to start for the customer

Note: For more information on configuring campaigns and packages, see the *Oracle Utilities Customer Care and Billing User Guide*. Also refer to the CCB demo environment for a sample self-service campaign.

Admin Data Setup

This section describes unique setup steps specifically related to configuring your system for the integration. The following topics are discussed:

- Self-Service Master Configuration
- IWS Configuration

For more information about configuring Oracle Utilities Customer Care and Billing, see the *Oracle Utilities Customer Care and Billing User Guide.*

Self-Service Master Configuration

Your implementation must configure an instance of the self-service master configuration. The master configuration contains pertinent information needed for a CCB and self-service integration.

See the demo environment for a sample self-service master configuration. Embedded help is available on the master configuration to guide you through the setup process.

Displaying Self-Service Alerts

The base product provides the following sample alert scripts:

Alert Type	Script	Description
Active Pay Arrangement / Pay Plan	C1-PAPPAlert	This script checks if the account has an active pay arrangement SA or pay plan.
Budget Enrollment	WX-NtfyBudg	This script checks if the account is currently on a budget or not.
Highlight Open Service Tasks	C1-OpenTaskA	This script checks if the account has a non-final service task for a task type that requires an alert. The alert header and description are defined on the service task type. Such an alert might be useful if the account has a pending start/stop or pay arrangement request.
Highlight Pending Start/Stop	C1-PendStSpA	This script checks if the account has a pending start or pending stop SA.
Outstanding Balance	C1-AcctCurrB	This script retrieves the account's outstanding balance to display. Note that all new charges are excluded from the outstanding balance returned.
Prepay Biller Information	C1-PPBAlert	For accounts with an active Prepay Biller Task, this script checks if the available prepaid balance is below a certain threshold.

The following steps should be followed if your implementation requires additional alerts:

- Create a script that contains the alert construction logic
- Define an alert type for the new alert by adding a value for the lookup field **WX_CCB_ALERT_TYPE_FLG**
- Alert type, priority and the corresponding script must be defined on the self-service master configuration for CCB to include this type of alert when building the alert list returned to the self-service application.

Enrolling Accounts

A self-service user is prompted for certain information when requesting access to an account. The required information or verification fields must be defined on the self-service master configuration for each line of business you've classified in the self-service application. Verification fields can be defined for one of the following field types:

- **Match Identifier**. The identifier type to use is required. The identifier type description is displayed as the verification field label in the self-service application. For example, assume that the identifier type specified here is social security number. The self-service user will be prompted to enter the social security number along with the account number that they want to enroll. A service call will then be made to CCB to verify that the SSN that the customer entered in the self-service application actually matches the SSN captured in CCB for the account. Note that the identifier value must be entered using the CCB identifier format. For example, social security number must be entered in the format 999-99-9999.
- **Match Phone Type**. The phone type to use is required. The phone type description is displayed as the verification field label in the self-service application. Note that the phone number must be entered using the CCB phone format.
- **Match Any Phone.** A field name is required. The field description is displayed as the verification field label in the self-service application. A service call will then be made to CCB to verify that the phone number that the customer entered in the self-service application actually matches one of the phone numbers captured in CCB for the account.
- **Customer Class.** The account's customer class is looked-up from this list to identify the verification fields if line of business is not supplied.

Additional field types can be introduced by adding new values for the lookup field **WX_FIELD_TYPE_FLG**. If additional field types are introduced, your implementation must ensure that the appropriate logic to validate the new field values is added to the lifecycle of the transaction BO linked to the Account Verification service task type defined on the self-service master configuration.

Note: If a customer adds new fields for enrollment to CCB Master Configuration, it would require reloading labels in OUCSS and restarting OUCSS.

Account List Filtering

A customer may have multiple accounts enrolled or linked in self-service. Self-service allows the customer to select a subset of their accounts based on some filter criteria. The **Default Account List Filter** to display in self-service must be configured on the self-service master configuration. The filtering of accounts works as follows:

- When a self-service user logs in, their account list is sent to CCB to retrieve the account information to display. Inbound service WXViewAccountList accepts the list of accounts and calls the appropriate processing script defined on the self-service master configuration to retrieve account information. The base product includes a sample information script that returns the accounts service address, and an indication if the account covers service at multiple premises.
- In addition to the account list, CSS may also send an account filter request. In this case, the inbound service will first call the **Account List Filter** script defined on the self-service master configuration to select the accounts that match the filter criteria prior to calling the account information script. The base product includes a sample filter script that allows filtering by the service address field constituents (address line1, city, state and/or postal code)

Account Documents

If your business wishes to make certain account documents available for viewing by the customer, you must set up a **File Location Value Characteristic Type** for each document. These characteristic types are then defined in master configuration. If any of these characteristics are present on the account, the inbound service returns the description and URL of the document for display in CSS.

Payment Processing

Payment types that your business accepts via self-service must be defined on the self-service master configuration along with a corresponding tender type. The base product includes values for Checking, Savings and Credit Card Withdrawal. Additional payment types can be introduced by adding new values for the lookup field **EXT_TYPE_FLG.** The tender type associated with each payment type will be used when creating a payment tender for the online payment made by the self-service user. The tender type, along with an external source id (or bank routing number), is also used to retrieve an auto pay source when setting a self-service user enrolls in an automatic payment plan with either checking or savings withdrawal.

In addition, credit card types that your business accepts via self-service must be defined on the self-service master configuration along with a corresponding autopay source. New credit card types may be defined by adding values for the lookup field **CARD_TYPE_FLG**. The credit card types defined are used to build a dropdown list of valid credit cards when a user posts a payment in the self-service application. It's also used to retrieve an auto pay source when a self-service user enrolls in an automatic payment plan.

Self-Service Processing Scripts

The base product provides processing scripts that contain the logic used to build information strings displayed on taskflows/portlets in the self-service application. Your implementation should ensure that these processing scripts are defined on the self-service master configuration.

The product includes the followig processing scripts:

Processing Type	Script	Description
Account List Filter	C1-AccLstFlt	This script contains the logic used to filter a customer's account list. Customers with a large number of linked accounts may select a subset of these accounts by providing some filter criteria. The base script allows a customer to filter their account list by the service address field constituents (address line1, city, state and/or postal code).
Budget Eligibility	C1-BudgetElg	This script determines if a customer is eligible to be placed on a budget plan. For eligible customers, it returns to self-service the list of SAs that are budge-eligible along with the recommended budget amount for each one; as well as the list of budget-eligible SAs that were made ineligible through a plug-in algorithm on the SA Type. For ineligible customers, the script returns the reason why the customer is ineligible. A customer is ineligible if (a) none of its SAs are budget-eligible, (b) he is already on a budget plan, (c) a budget request service task is in progress, and (d) the CIS division associated with the account has no budget plan specified for it in CCB.
Context Information	C1-CtxtInfo	This script returns a list of properties and values about an account to help CSS tailor their displays for the customer. These are captured in the extendible lookup C1-CSSContext. An example of a property, which this script determines, is 'PPB' (used to indicate whether or not an account has active prepaid service) with corresponding values of either 'Y' or 'N'.
Maintain Communication Preference	C1-MtCmPf	This script returns a list of active communication preference as well as notification types eligible for communication preferences for the person and account in context. This allows CSS users to maintain notification communication preferences stored in CCB. This is not used when OUNC owns customer notification preferences.
One Time Payment Creation	C1-AddOTPay	This script adds a One Time Payment service task and maintains self-service payment options.
One Time Payment Cancellation	WX-CanSchPay	This script discards One Time Payment service tasks when users cancel future payments.
Pay Arrangement Eligibility	C1-PAEligibl	This script determines if a customer is eligible for a pay arrangement. For eligible customers, it returns to self-service the list of pay arrangement task types to select from, as well as the duration and installment amount for each type. For ineligible

		customers, this script returns the reason why the customer is not eligible for a pay arrangement. The base product eligibility script does the following: •Check if an active pay arrangement already exists for the customer. •Check if a non-final pay arrangement service task already exists for the customer. •Check if the customer had a broken pay arrangement in the past 6 months.
Projected Charges	C1-CalcPrChg	This script builds the account's service charges to date and the projected charges for the end of billing period. It may be overridden with custom service script.
Prepaid Billing Detailed Information	WX-PerInfo	This script returns the last daily calculated cost as well as the 30 day average cost related to prepaid billing charges.
Prepaid Billing Main Information	WX-PerInfo	This script returns a customer's prepaid credit balance (expressed as a positive figure) as well as the last seven bill segments for each of his prepaid service agreements.
Start Service Criteria	C1-StrSvcCri	When a customer requests new service, the Start Service Criteria script is executed to determine what information should be requested from the customer. The base product script does the following: •Using the Campaign defined on the self-service master configuration, it gets the questions & miscellaneous fields on the campaign to present to the self-service customer. The customer's response to these questions determines the eligible SA types and start options for new service. •Retrieves the list of New Customer Identifiers defined on the self-service master configuration. •Retrieves the list of phone types and descriptions configured in CCB. •Retrieves current contact information for existing customers, e.g., name, mailing address and contact numbers.
UnEnroll Account	C1-UnEnlAcct	This script cleans up notifications when accounts are deactivated. This is used when notifications are owed by OUNC or the original billing notifications are being used.
Account Information	WX-AcctInfo	This script builds the account information string displayed on self-service portlets. Note that account information scripts must use the data area WX-AccountInfo.
Person Information	WX-PerInfo	This script builds the person information string displayed on self-service portlets. Note that person information scripts must use the data area WX-PersonInfo.
Register Information	WX-RegInfo	This script builds the register information string displayed on self-service portlets. Note that register information scripts must use the data area WX-RegisterInfo.
Service Agreement Information	WX-GetSAInfo	This script builds the SA information string displayed on self-service portlets. Note that SA information scripts must use the data area WX-SAInfo.
Service Point/Meter Information	WX-SPMtrInfo	This script builds the SP/meter information string displayed on self-service portlets. Note that SP/meter information scripts must use the data area WX-SPMeterInfo.

Self-Service Tasks

Ensure that the tasks supported by your implementation are defined on the self-service master configuration. The corresponding service task type must be specified for each service task identifier. This is the service task type that the system will use when an inbound request is received for a specific task. Refer to the business object's detailed description in CCB for more information on the service task.

The product includes the following:

Service Task Identifier	Business Object	Description	Transaction BO
Account Verification	WX-AccountVerifyTaskType	This BO defines the expected behavior when verifying that a self-service user is the owner of an account.	WX-AccountVerifyTask
Auto Pay Setup	WX-AutoPayTaskType	This BO defines the expected behavior when a self-service user enrolls (or unenrolls) their accounts for automatic payments.	WX-AutoPayTask
Bill Due Notification*	WX-NotifyBillDueTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding upcoming bill due dates.	WX-NotifyBillDueTask
Bill Ready Notification*	WX-NotifyBillReadyTaskType	This BO defines the expected behavior when a self-service user requests notifications when their new bill is ready to view online.	WX-NotifyBillReadyTask
Late Payment Notification*	WX-NotifyLatePayTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding late payments.	WX-NotifyLatePayTask
Meter Read Creation	WX-SelfServiceMRTaskType	This BO defines the expected behavior when a self-service user enters a meter read.	WX-SelfServiceMRTask
One Time Payment	WX-OneTimePayTaskType	This BO defines the expected behavior when a self-service user makes a payment.	WX-OneTimePayTask
Payment Received Notification*	WX-NotifyPayReceivedTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding payments received.	WX-NotifyPayReceivedTask
Start/Stop Service Request	C1-StartStopTaskType	This BO defines the expected behavior when a self-service user requests to start, stop or transfer service. To configure your system for self-service requests using this business object: • Setup a service task type that references this business object • Ensure that the service task type is defined on the self-service master configuration for the Start/Stop Service Request task In addition, the system caters for displaying an alert in the self-service application if the customer has a non-final task of a particular type. To use this functionality: • Ensure that customer alert required is set to true on your servivce task type. Specify the message category and numbers to use when constructing the alert. The base product includes message category 11115, message numbers 11816 and 11840 for the alert header and description text. • Also ensure that the Highlight Open Service Tasks script (C1-OpenTaskA) is specified on the self-service master configuration. • Define valid discard reasons for self-service tasks. Navigate to the lookup page and define valid values for the lookup field C1_CUSTREQ_DISCARD_RSN_FLG	C1-StartStopRequestTask

* These Self-Service Task Types are used to support billing notifications that are initiated in OUCSS and result in storing a self-service task in CCB. There are two newer notificaiton preference models. These self-service task types are only used to support the original billing notifications. OUCSS contains an configuration that indicates the model that is used.

The following guidelines should be followed if your implementation requires additional service task identifiers:

- Setup an inbound web service to create the task instance (see the IWS configuration section below)
- Add the service task type and service task business objects
- Configure the corresponding service task type
- Add a lookup value to the lookup field **WX_BO_SSTASK_TYPE_FLG** for the new service task identifier
- Update this self-service master configuration to include the new task identifier and corresponding task business object
- Create the corresponding user interface or taskflow/portlet in OUCSS using Oracle ADF (Application Development Framework) that will be responsible for invoking the inbound service

Notification Service Tasks

OUCSS supports notification preferencs ownded by Oracle Utilities Notificaton Center (OUNC) or notification preferencs owned by CCB. This section is only used when OUNC owns the notifications.

Ensure that the notification tasks supported by your implementation are defined on the self-service master configuration. The corresponding service task type must be specified for each notification type identifier. This is the service task type that the system will use when an inbound request is received for a specific task. Refer to the business object's detailed description in CCB for more information on the service task.

The product includes the following:

Notification Type Identifier	Business Object	Description	Transaction BO
Bill Due Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding upcoming bill due dates.	WX-NotifyBillDueTask
Bill Ready Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications when their new bill is ready to view online.	WX-NotifyBillReadyTask
Issues Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding updates to issues they have opened.	C1-NotifyPreferenceTask
Late Payment Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding late payments.	WX-NotifyLatePayTask
Marketing Communication Preference	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding marketing initiatives that might be available.	C1-NotifyPreferenceTask
Payment Received Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests notifications regarding payments received.	WX-NotifyPayReceivedTask
Prepaid Billing New Charge Notification	C1-NotifyTaskType	This BO defines the expected behavior when a self-service user requests a notification each time a new charge is generated for his prepaid account.	C1-NotifyPreferenceTask

Prepaid Billing Payment Request Notification	C1-PPBPaymentNotifyTaskType	This BO defines the expected behavior when a self-service user requests a notification each time a payment request is made to top up his prepaid account.	C1-NotifyPreferenceTask

Configuring Notification Tasks

For the customer notification tasks defined, the following configuration is required:

- An outbound message type and an external system must be setup and defined in master configuration under the External Communication section. A Request Type of Notifications exist to capture this information, which will be used when notifying the self-service user.
- When a self-service user signs up for notification, the inbound service WXSetNotificationPreferences creates a task for the service task type defined on the self-service master configuration. This task remains active until the self-service user cancels his registration for the notification. This applies to the model where OUNC owns the notification preference. When CCB owns the notification preference, a contact preferences is created. A service task may be created or updated by the contact preference if configured to do so. When this maintained via OUCSS, WXMaintainCommPreferences miantains the contact preferecens in CCB.
- For the bill due, bill ready, payment received and late payment notification types
  - The various CCB processes that may trigger notifications (like billing and payment processing), will set an indicator on the task to mark it as due for notification. The following algorithms must be plugged in to enable this:
    - For bill due, bill ready, and late payment notifications, bill completion algorithm WX-SSBNOTIFY must be defined on the applicable customer classes.
    - For payment received notifications, payment freeze algorithm WX-SSPNOTIFY must be defined on the applicable customer classes.
  - An enter algorithm on the notify state of the transaction BO is responsible for sending the notification via email. It retrieves the outbound message type and external system specified on the service task type.
  - An exit algorithm on the notify state of the transaction BO is responsible for resetting the notification indicator once the message is sent.
  - The self-service notification monitor WX-NOTIF must be scheduled to process these notifications.
- For the issues, marketing communication, prepaid billing new charge and prepaid billing payment request notification types, the management of when the notifications are sent are handled by their associated base objects (forms, leads, and prepay biller service task, respectively). The notification task that is created simply captures the communication preference that the self-service user has specified. When CCB owns the notification preference, a contact preferences is created in lieu of a notification taks. Some notificaton types utilize notification tasks. •For more information on configuring CCB owned notifcaiton types and associated self-service task type, refer to the Oracle Utilities Customer Care and Billing user documentation.

Forms

Ensure that the form tasks supported by your implementation are defined on the self-service master configuration. One or more service task types must be specified for each type of form. This is the service task type that the system will use when an inbound request is received for a specific task. Refer to the business object's detailed description in CCB for more information on the service task.

The product includes the following:

Form Type	Business Object	Description	Transaction BO
Forms - Issues	C1-FormTaskType	This BO defines the expected behavior when a self-service user creates a form of type Issue.	C1-FormTask

C1-FormTaskTypeForNotification	This is used when OUNC owns notification preferences, This BO is the same as the above, but is used when CCB owns notification preferences.

Configuring Form Service Tasks

The following configuration is required for configuring a form of type **Issues**:

- Configure the service task type your implementation would like to use for each issue
  - The base product includes business object C1-BudgetTaskType that defines the expected behavior when a self-service user creates a specific form. This business object allows for the definition of a Campaign that contains the prompts needed to collect additional information from the customer about the issue he is creating. The customer's responses will be recorded on the service task that ultimately gets created. This service task BO is defined as the Related Transactional BO on the service task type.
  - This business object caters for the definition of notification information. Two sets of notification information are available: one for use when an issue is updated by the CSR and one for use when it transitions to the Closed state. The notification configuration here is treated as an override to the Issues Notification configuration discussed in the previous section. If these notifications are desired, the algorithm that creates the notification (C1-BLDFRNOTF) must be plugged in on the Notifying Customer and Closed states. Retry configuration is also catered for by this business object, as well as a script parameter for determining whether the form is updateable by a customer (the default is yes). When a customer requests an issue to be closed, they must provide a close reason. The reasons available for selection are also defined in this business object.
  - This business object caters for displaying an alert in the self-service application if the customer has a non final task of a particular type. A customer alert required indicator on the service task type instance controls this behavior. The message category and numbers to use when constructing the alert must be specified on the task type. You must define message category and message numbers for the alert header and description text.
  - If your implementation wishes to display such an alert to the self-service user, the Highlight Open Service Tasks alert script (C1-OpenTaskA) must be specified on the self-service master configuration
- Ensure that the service task types are defined on the self-service master configuration
- Your implementation must define valid close reasons for the customer to choose from when requesting a form be closed. Navigate to the Admin -> S -> Status Reason and define them for the C1-FormTask BO.

When a customer clicks on a link in the Oracle Utilities Customer Self Service application to create an issue, the following takes place:

- An initial request is sent to CCB to retrieve the service task types associated with the form type (in this case « Forms – Issues »), i.e., the WXFormsManagement inbound service is called with a read action. The script does the following:
  - In the case of an initial call (customer has not selected a service task type) the script returns the descriptions of all service task types as defined in master configuration for the form type.
  - Once a service task type is selected by the customer, a second Read invocation is performed. The service returns the prompts associated with the Campaign that may have been defined on the service task type.
  - A third Read mode is available for customers updating an existing form. In this mode, the form type, service task type and service task ID are provided. The information related to the service task is then returned to self-service for display to the customer. Self-service displays a list of existing forms to the customer via the WXFormList service.
- Once the customer has entered all the information needed for form creation, a second request is sent to CCB to process the form; e.g., the WXFormsManagement inbound service is called with an update action
- This results in the creation of a service task for the type selected. The service task BO to use is defined as the transactional BO on the task type. The base product includes transactional BO C1-FormTask for this purpose.

- When a Campaign is defined, its prompts may be related to a column reference. The system event Service Task Order Processing on the column reference is available for execution. The algorithm (C1-EXSTSKCRF) is delivered to execute this system event and may be used by your business to initiate any other processing based on a customer's response to the campaign prompt. The algorithm needs to be plugged in on the C1-FormTask BO if this feature is desired. Likely candidates for plugging this into the BO life cycle are the Closed state and the In Progress state.

Configuring Additional Form Types

The following configuration is required for configuring additional form types:

- Navigate to the lookup page and define the new form type value for C1_SS_CATEGORY.
- The new form type value must be defined on all form service task types that fall under it.
- The new form type value must also be defined in Oracle Utilities Customer Self Service as it needs to pass this value when invoking the inbound service.

Additional Customer Requests

In addition to the requests defined on the self-service master configuration, the base product also supports the following:

- Requesting to be placed on a budget plan
- Requesting payment arrangements to payoff outstanding debt.
- Starting, stopping or transferring service for new or existing customers.

These requests result in the creation of a service task that performs the processing required to complete the request. A parent business object **(C1-SelfServiceCustomerReqTask**) captures the common schema and lifecycle for these service tasks. A child business object exists for the budget request (**C1-BudgetTask**), payment arrangement request (**C1-PayArrangementTask**) and start/stop request (**C1-StartStopRequestTask**). These product business objects support the following:

- Customer requests received result in the creation of a pending service task and the task id is communicated to the customer
- Implementations can use the validation state to plug-in any basic validation algorithms they wish to perform. Note that these validation errors are returned to the self-service customer, so the message text should be customer friendly. A to do entry is not created for validation errors since the assumption is that the customer will attempt to submit the request again after correcting the error
- If no validation errors are encountered, the task is transitioned through to completion. If any internal errors are encountered, the task transitions to the Processing Error state where a to do is created to notify a CSR.

Budget Requests

The base product supports self-service customers requesting to be placed on a budget plan:

- Configure the service task type your implementation would like to use for self-service.
  - The base product includes business object C1-BudgetTaskType that defines the expected behavior when a self-service user requests to be placed on a budget plan. This business object caters for displaying an alert in the self-service application if the customer has a non final task of a particular type. A customer alert required indicator on the service task type instance controls this behavior. The message category and numbers to use when constructing the alert must be specified on the task type. The base product includes message category 11115 and message numbers 11873 and 11874 for the alert header and description text.
  - If your implementation wishes to display such an alert to the self-service user, the Highlight Open Service Tasks alert script (C1-OpenTaskA) must be specified on the self-service master configuration

- Ensure that the service task type as well as a budget eligibility script are defined on the self-service master configuration
- Your implementation must define valid discard reasons for self-service tasks. Navigate to the lookup page and define valid values for the lookup field C1_CUSTREQ_DISCARD_RSN_FLG

When a customer clicks on a link in the self-service application to request a budget plan, the following takes place:

- An initial request is sent to CCB to check if the customer is eligible, i.e., the WXProcessBudgetRequest inbound service is called with a read action. The budget eligibility script defined on the self-service master configuration is executed. The sample eligibility script included in the base product (C1-BudgetElg) does the following:
  - Checks if the account is ineligible (none of its SAs are budget-eligible; the account is already on budget; a budget request task is already in progress; no budget plan is defined in CCB for the account's CIS division) and formats the ineligible message
  - If the account is otherwise eligible, the script returns the list of budget-eligible SAs along with their service type description and recommended budget amount. It also returns the list of budget-eligible SAs that were made ineligible through a plug-in algorithm on the SA Type. The script also returns the total budget amount and the number of months in between true ups for that particular budget plan.
- If the end customer is agreeable to the recommended budget amount, a second request is sent to CCB to process the application; e.g., the WXProcessBudgetRequest inbound service is called with an update action
- This results in the creation of a service task for the type selected. The service task BO to use is defined as the transactional BO on the task type. The base product includes transactional BO C1-BudgetTask for this purpose.

Payment Arrangement Requests

The base product supports self-service customers requesting payment arrangements to payoff their outstanding debt. The following steps should be followed to enable this:

- Configure the payment arrangement service task types your implementation would like to use for self-service.
  - The base product includes business object C1-PATaskType that defines the expected behavior when a self-service user requests a payment arrangement. This business object captures the number of installments the end customer would pay, as well as the CIS division and SA type to use when setting up the pay arrangement SA
  - In addition, this BO caters for displaying an alert in the self-service application if the customer has a non final task of a particular type. A customer alert required indicator on the service task type instance controls this behavior. The message category and numbers to use when constructing the alert must be specified on the task type. The base product includes message category 11115 and message numbers 11816 and 11842 for the alert header and description text.
  - If your implementation wishes to display such an alert to the self-service user, the Highlight Open Service Tasks alert script (C1-OpenTaskA) must be specified on the self-service master configuration
- Ensure that the service task types as well as a payment arrangement eligibility script are defined on the self-service master configuration
- Your implementation must define valid discard reasons for self-service tasks. Navigate to the lookup page and define valid values for the lookup field C1_CUSTREQ_DISCARD_RSN_FLG

When a customer clicks on a link in the self-service application to request a payment arrangement, the following takes place:

- An initial request is sent to CCB to check if the customer is eligible, e.g., the **WXProcessPayArrangementRequest** inbound service is called with a read action. The pay arrangement eligibility script defined on the self-service master configuration is executed. The sample eligibility script included in the base product (C1-PAEligibl) does the following:
  - Checks if an active pay arrangement service agreement already exists for the customer
  - Checks if a non-final pay arrangement service task already exists for the customer
  - Checks if the customer had a broken pay arrangement in the past 6 months

  - If one of these checks is true, the customer is ineligible and the script returns the reason to the self-service application
  - If the customer is eligible for a pay arrangement, the script returns the list of pay arrangement task types to select from, as well as the duration and installment amount for each type
- If the end customer selects one of these pay arrangement types, a second request is sent to CCB to process the application; e.g., the **WXProcessPayArrangementRequest** inbound service is called with an update action
- This results in the creation of a service task for the type selected. The service task BO to use is defined as the transactional BO on the task type. The base product includes transactional BO C1-PayArrangementTask for this purpose.

Start/Stop Service Requests

When a customer clicks on a link in the self-service application to start, stop or transfer service, the following takes place:

- An initial request is sent to CCB to gather information to request from the customer, e.g., the **WXProcessStartStopRequest** inbound service is called with a read action. The start service criteria script defined on the self-service master configuration is executed. The sample criteria script included in the base product (C1-StrSvcCri) does the following:
  - Using the campaign defined on the self-service master configuration, it gets the questions & miscellaneous fields on the campaign to present to the self-service customer. The customer's response to these questions determines the eligible SA types and start options for new service
  - Retrieves the list of new customer identifiers defined on the self-service master configuration
  - Retrieves the list of phone types and descriptions configured in CCB
  - Retrieves current contact information for existing customers, e.g., name, mailing address and contact numbers
- If your implementation wishes to use the sample criteria script included in the base product, you must setup a campaign and package(s) for each customer class supported.
- After entering required information and submitting the self-service request, a second message is sent to CCB to process the service request; e.g., the **WXProcessStartStopRequest** inbound service is called with an update action
- This results in the creation of a service task for the service task type defined on the self-service master configuration. The service task BO to use is defined as the transactional BO on the task type.

External Communications

The base product provides the following request types that are used to request data from an external party in a self-service integration:

- **Usage Overview.**. This is used to retrieve an x-day usage overview for a self-service user's account, where the number of days is provided as input. The system will attempt to retrieve usage information from MDM for each of the account's service agreements that require bill determinants. The base product includes a business object for the MDM request called **WX-UsageOverviewRequest** that should be defined on the outbound message type.
- **Usage Details.** This is used to retrieve usage details for a self-service user's account for some period (e.g., year, month or day). The system will attempt to retrieve usage information from MDM for each of the account's service agreements that require bill determinants. This service may also return temperature information. The base product includes a business object for the MDM request called **WX-UsageOverviewRequest** that should be defined on the outbound message type.
- **Usage Request.** This is used to request bill determinants from MDM when needed for either rate analysis calculations, or when attempting to calculate unbilled charges to date. The base product includes a business object for the MDM request called **C1-UsageRequestOutMsg** that should be defined on the outbound message type

These request types along with the corresponding external system and outbound message type should be specified on the self-service master configuration.

In addition, an email address must also be specified on the self-service master configuration. This email address is used when sending email notifications to a self-service user; e.g., if the self-service user has signed up for bill ready notifications. The outbound message type and external system used when creating the email outbound message are defined on the appropriate service task types. The **Email Address** is used to populate the 'reply-to' address on the email.

The 'from' address on the notification email is configured in Middleware. Refer to Installation document -Installing Oracle Utilities Notification Center Flows – Sender Address Configuration on how ‘from’ email address of notification is configured in Middleware.

Auditing Self-Service Calls

The base product supports feature to track self-service invocations that update CCB. Inbound service scripts required for auditing are configured in master configuration. In addition, the audit script - C1-CSSAudit that creates the audit entries must be supplied. When self-service calls any of these service scripts with update action, an audit entry is created in the audit table. These audit entries may be viewed by navigating to Admin, General Audit.

For custom inbound services that require auditing, implementation may include these inbound service scripts in the extendable look-up - C1-CSSInboundServiceScripts. The custom inbound service script must be changed to call the common audit service script - C1-CSSAuProc. Refer to any of the delivered base inbound service scripts as sample.

Service Charges to Date

Days Since Last Bill is used to evaluate the latest bill date relative to the business date.

If populated, the value is subtracted from the business date. If the latest bill's Bill Date is between this derived date and the business date, CCB returns the latest bill amount as service charges to date, to OUCSS, instead of calling MDM to obtain usage and calculate the charges.

This cover the scenario when the user logs in to OUCSS just after the bill was sent. In this case, latest bill’s amount will be displayed in OUCSS.

Link Self-Service User to Person/Account

OUCSS can be configured to link the user with a person. When this functionality is enabled in OUCSS, the **Account Person Maintenance Script** is required. The base product provides a sample script **C1-CSSUsrAcM** that does the following.

- Updates or creates a person record. The following are needed when a person record is created.
    - If a person ID is required on installation options, define a **Person ID Type** and **Person ID Value** to be used when creating a person. OUCSS does not capture this information and it may be required to create a person. Is expected that the type and value are similar to the defaults used when persons are created and this information is not available. For example the type may be "NOID" and the value "111111".
    - The **Format Person Name Script** is optional. If not provided, name is formatted "LastName,FirstName". Format Person Name Scripts must use the data are **C1-FormatPersonNameData**. The **Person Name Separator Character** is passed into the script's data area.
- Updates or creates a person contact with the CSS User Name if **Person Contact Type for CSS User** is defined. The person contact **Contact Nickname** will contain the Web User Id.
- Creates a person contact record for email if the **Person Contact Type for Email** is defined and an email person contact record does not exist with the same email address. The email address is the one provided and retained in OUCSS. Some additional behavior is provided based on **Always Add Email**.

  - When checked, the system creates an email person contact as long as an existing contact for the same email does not exist.
  - When unchecked, a new email person contact is only created if the person does not have an email person contacts as identified by the contact routing.

- Updates or creates an account/person record. The account/person indicates the OUCSS role, which is used to indicate that the record exists in OUCSS for the account and user combination. When a new record is created, it is done using the **Account Relationship Type**.
- The **Self-service URL** and **Self-service Target URL** are optional and used to provide a hyperlink to OUCSS from the Person page of Account. This allows a system user to access self-service as the CSS user (Person) and Account.

Log Self-Service Access

The inbound service Get Self Service Context Information is called when an OUCSS user accesses an account. This script can be used to log OUCSS access in CCB.

Define a **Log Self-service Access Script** to enable this functionality. The base product includes sample script **C1-CSSCreCC** that creates a customer contact. This sample script requires that a **Customer Contact Class** and **Customer Contact Type** are defined.

Self-Service To Do Types

Exception Info To Do Type defines the To Do Type uses to capture errors encountered while processing requests from OUCSS when these errors should not be sent back to OUCSS. The system is delivered with to do type **Self-Service Errors** (C1-SSTD) or this purpose.

Exception Info To Do Role is the To Do Role for the above To Do Type.

IWS Configuration

Message Sender

To create a real-time Message Sender configured to communicate with the integration layer:

Message Sender for Usage Request Integration Service

To create a new Message Sender which points to the Usage Request EBF endpoint URL for Usage Request integration service.

1. Navigate to Admin Menu, Message Sender.
2. Enter a unique Message Sender and Description.
3. Populate values as follows:
   - Invocation Type = Real-time
   - XAI Class = RTHTTPSNDR. (Real Time Sender to route messages via HTTP)
   - Select the Active check box.
   - MSG Encoding = UTF-8 message encoding
4. Select the Context tab and set values for the following Context Types:
   - HTTP Login User – User ID for the URL to be accessed

- HTTP Login Password – Password for the URL to be accessed
- HTTP Header – SOAPAction: "process"
- HTTP Method (POST/GET) – POST
- HTTP Proxy Host – Set the proxy server name if applicable
- HTTP Proxy Port – Port for the proxy server if applicable
- HTTP Transport Method – SendReceive
- HTTP Timeout: 60 (put timeout in seconds)
- HTTP URL 1 – Set the URL to be accessed. If the URL value does not fit, use the additional HTTP URL types to set the complete URL. This should point to the Usage Request EBF endpoint URL.

For example:

```
http://demoenv:8015/soa-infra/services/CCB2-
MDM2/OUCCB2OUMDM2SSUsageReqEBF/ouccb2oumdm2ssusagereqebf_client_ep
```

Note: The endpoint URL for the process can be obtained from the SOA Enterprise Manager. From the SOA folder in the Navigator, under soa-infra -> CCB2-MDM2, select the OUCCB2OUMDM2SSUsageReqEBF composite application. At the top of the page, click Test tab. In the test tab, you will see the composite application's endpoint URL.

Outbound Message Type

To create an Outbound Message Type for each process in this the integration:

Usage Request Integration Service

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique Outbound Message Type and Description.
3. Populate values as follows:
   - Business Object =*C1-UsageRequestOutMsg* (Usage Request Outbound Message)
   - Priority = (choose from the selection)

External System

To create a new External System for this integration:

1. Navigate to Admin Menu, External System.
2. Enter a unique External System and Description.
3. Set Our Name in Their System to *CCB*
4. Define the Outbound Message Types associated with the integration.

   For Usage Request Outbound Message Types, populate the values as follows:
   - Outbound Message Type = (Outbound Message Type for Usage Request)
   - Processing Method = *Real-time*
   - Message Sender = (Message Sender for Usage Request Integration Service)
   - Message XSL = CDxAddEnvelope-SOAP1-2.xsl
   - Response XSL = C1-CCBRemoveEnvEnvelopeAndNamespace.xsl

For Usage Overview Outbound Message Types, populate the values as follows:

- Outbound Message Type = (Outbound Message Type for Usage Overview)
- Processing Method = *Real-time*
- Message Sender = (Message Sender for Usage Overview Integration Service)
- Message XSL = CDxAddEnvelope-SOAP1-2.xsl
- Response XSL = C1-CCBRemoveEnvEnvelopeAndNamespace.xsl

For Usage Detail Outbound Message Types, populate the values as follows:

- Outbound Message Type = (Outbound Message Type for Usage Detail)
- Processing Method = *Real-time*
- Message Sender = (Message Sender for Usage Detail Integration Services)
- Message XSL = CDxAddEnvelope-SOAP1-2.xsl
- Response XSL = C1-CCBRemoveEnvEnvelopeAndNamespace.xsl

Chapter 4

Meter Data Management Configuration

Oracle Utilities Meter Data Management Configuration

To set up Oracle Utilities Meter Data Management for self-service implementation, the following admin data setup must be configured:

- Self-Service Integration Master Configuration
- Self-Service Task Types

Self-Service Integration Master Configuration

Create a master configuration for self-service integration in Oracle Utilities Meter Data Management.

Navigation	Guideline
Admin Menu > Master Configuration	Create a master configuration to be used for self-service integration.

Self-Service Task Types

Create self-service task types for each self-service task in Oracle Utilities Meter Data Management.

Navigation	Guideline
Admin Menu > Self-Service Task Type	Create self-service task types for each self-service task business objects.

Inbound Web Services (IWS)

The following are the base Inbound Web Services invoked by Oracle Utilities Customer Self Service.

Name	Description
WX-CreateSelfServiceMeterRead	This inbound service is used by the self-service application. It is responsible for retrieving and adding manual or scalar meter reads. When adding a new meter read, the service creates an instance of the IMD business object.
WX-GetScalarConsumptionSummary	This inbound service retrieves consumption information to display in the self-service application. It retrieves consumption for service agreements that do not require MDM bill determinants.
WX-GetUsageOverview	This inbound service retrieves an x-day usage overview for a self-service user's account. The number of days is provided as input to this service. The system will attempt to retrieve usage information from MDM for each of the account's service agreements.
WX-RETWSSTOUMappingService	This inbound service retrieves usage details for a self-service user's account for some period (e.g., year, month or day). The system will attempt to retrieve usage information from MDM for each of the account's service agreements. This service may also return temperature information.
WX-MultipleAccountTOUUsagesByServiceType	This inbound service invokes the Get Usage Details service to retrieve the input list of account's aggregated usages. Each account's usage is summarized by service type, UOM and SQI.
WX-MultipleAccountUsagesByServiceType	This inbound service invokes the Get Usage Details service to retrieve the input list of account's usages. Each account's usage is summarized by account, service type, UOM and SQI.
WX-MultipleAccountUsagesDownload	This inbound service invokes the Get Usage Overview service, to retrieve the input list of account's usages by usage subscription.

Admin Data Setup

This section describes unique setup issues specifically related to configuring your system for the integration.

Self Service Master Configuration

Your implementation must configure an instance of the self-service master configuration. The master configuration contains pertinent information needed for a MDM and self-service integration, including the following:

Main

Temperature Details

To show temperature details, temperature source factor and mode must be set in the master configuration. In addition, Temperature Source Factor, Temperature Zone Characteristic Type, Temperature Source Measuring Component and the Service Point BO must also be configured.

Offset Duration

Applies to retrieval of frequent scalar usage details. Specify the number of minutes added to end date time buckets used as search limit when searching for measurements. This is to handle end readings that are off by few minutes from the expected read date time.

Usage Calculation Skip Option

This applies to Multiple Accounts Aggregation and Comparison service. Indicates how processing proceeds if one of the account's usage subscriptions is skipped. Specify 'All or Nothing' to skip the account if one of its usage subscription is skipped. Specify 'Allow Partial Usage' to continue with the summarization if a usage subscription is skipped.

Calculation Function

This applies to Multiple Accounts Aggregation and Comparison service. Indicates how the usage returned by the Get Usage Details service is summarized for the account. If not supplied, the usage is summarized based on the UOM's measures peak flag - 'Sum' if UOM does not measure peak or 'Max' is UOM measures peak.

Service Charges to Date Settings

Daily Scalar Read Lag Window

Specify the number of hours subtracted from scalar end date time to form a window period for searching frequent scalar measurement used in calculating service charges to date.

Processing Scripts

The base product provides processing scripts that contain the logic used to build information strings displayed on taskflows/portlets in the self-service application. These include scripts that build information strings such as SP / Device Configuration, Measurement Component and Usage Subscription. Your implementation should ensure that these processing scripts are defined on the self-service master configuration.

Service Tasks

Ensure that the tasks supported by your implementation are defined on the self-service master configurationThe corresponding business object must be specified for each service task identifier. This is the business object that the system will use when an inbound request is received for a specific task. Refer to the business object's detailed description in MDM for more information on the service task.

The following guidelines should be followed if your implementation requires additional service task identifiers:

- Setup an inbound service to create the task instance (see the IWS configuration section below)
- Add the service task type and service task business objects
- Configure the corresponding service task type
- Add a lookup value to the lookup field **WX_BO_SSTASK_TYPE_FLG** for the new service task identifier

- Update this self-service master configuration to include the new task identifier and corresponding task business object
- Create the corresponding user interface or taskflow/portlet in OUCSS using Oracle ADF (Application Development Framework) that will be responsible for invoking the inbound service

Rate Compare Configuration

Used to configure how usage adjustments are applied when performing self-service rate comparisons. If your organization wishes to allow self-service customers to see the effect of usage adjustments on their consumption, configure the relevant usage adjustments possible for each rate defined in the CCB Rate Schedule extendable lookup. Types of usage adjustments (electric car, solar panels, etc.) are defined using the Usage Adjustment Type extendable lookup, and profile factors should be configured with measurement data representing the effect of each adjustment type. Customers can then choose one or more usage adjustments they would like to apply to their existing usage.

Supported Scalar Usage Groups

In order for MDM to return scalar consumption for a given usage subscription, the usage subscription's usage group must be defined in this list. UOM/TOU/SQI for the usage group is also specified in this list, to identify the specific scalar consumption that will be retrieved from usage transaction and to identify the unit of measure used when this consumption is displayed in OUCSS application.

Supported Interval Usage Groups

In order for MDM to return interval usage-to-date information for a given usage subscription or to identify applicable usage groups eligible for rate comparison, the usage subscription's usage group must be defined in this list.

Create Scalar Read Service Task Type

The Self Service Meter Read Task Type must be created to enable OUCSS direct call to MDM for self-service initiated creation of scalar meter read.

The following values must be supplied:

- Service Provider
- Data Source
- Measurement Condition

Service Provider

Service Provider for external application – OUCSS, must be created to specify the provider of the scalar meter readings

Processing method for the role – Initial Measurement Creation, must configured with the IMD BO to use in the creation of self-service scalar readings.

Temperature Zone Characteristic Type

This will hold the available temperature zones. See the demo environment for a sample temperature zone characteristic type.

Temperature Source Measuring Component

This will hold the temperature values. See the demo environment for a sample temperature source measuring component.

Temperature Source Factor

This is used to retrieve the measuring component from which the temperature values are retrieved. See the demo environment for a sample temperature source factor.

Service Point Business Object (Update)

Add temperature zone to the business object schema. This should be a flattened characteristic that references the temperature zone characteristic type. See the demo environment for a sample service point business object.

Note: Each service point instance must then reference a temperature zone.

Measuring Component Type

Specify a VEE group used in calculating projected usages.

Measuring Component

Specify a VEE group to override the measuring component type VEE group used in calculating projected usages.

Chapter 5

Network Management System Configuration

Network Management System (NMS) must be configured to support reporting an outage for a specific customer (account), viewing a customer's outage information, and viewing system-wide outage information by geographic area and outage locations from the Self Service Application.

Note: If CCB is the CIS System used, see the Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Network Management System Release 12.1.0 for customer synchronization processes between the two systems.

NMS Configuration

This section describes how to configure the Oracle Utilities Network Management System to meet the requirements for the integration.

The following components of the Oracle Utilities Network Management System are involved in the CSS Direct BPEL integration:

- **Geographic Outages, Geographic Outage Areas, and Geographic Outage Status Materialized Views** - The CSS application reads these Materialized Views to display Oracle Utilities Network Management System outage data.

  These materialized views are created in the Network Management System project configuration as defined in the Oracle Utilities Network Management System Configuration Guide Chapter eight, Building the System Data Model, section on Preparing the NMS Model for CSS.

Refer to the *Oracle Utilities Network Management System Installation Guide*, "Optional Spatial Outage Summary Installation" section, for setup details to refresh these materialized views.

- **PL/SQL package PK_CCB** - Provides access to the functions of the Oracle Utilities Network Management System required by the integration. It is part of Oracle Utilities Outage Management Base license.
- The following stored procedure are used by the integration :

Stored Procedured Name	Description
pk_ccb.trouble_call_config	Stored procedure to retrieve list of trouble codes configured in the Oracle Utilities Network Management System.
pk_ccb.submit_call	Stored procedure used to submit trouble calls to the Oracle Utilities Network Management System.
pk_ccb.job_history	Stored procedure to retrieve list of jobs matching the passed in search condition.
pk_ccb.switching_history	Stored procedure to retrieve a list of current, future, and (optionally) past switching plans affecting a given customer

For more information on the stored procedures, see the Generic IVR Adapter chapter in the *Oracle Utilities Network Management System Adapters Guide*.

- **Generic IVR Adapter** - Processes trouble calls received from Oracle Utilities Customer Self Service. It is part of Oracle Utilities Outage Management Base license.

The Generic IVR Adapter has to run with the '-troublecall' command-line option to enable trouble call data flow. The command-line option '-docustquery' should not be used because correct customer information is expected to be received from the Oracle Utilities Customer Self Service system.

For more information on configuring Generic IVR Adapter, see the Generic IVR Adapter chapter in the Oracle Utilities Network Management System Adapters Guide.

- **Service Alert** – Generates notification based on configuration received from Oracle Utilities Customer Self Service.

The Service Alert service process needs to be running in order for notifications to be generated, and can be used only in support of the Oracle Utilities Customer Self Service integration to Oracle Utilities Network Management System. Full use of Service Alert for configuring its own notifications requires the Oracle Utilities Outage Management Paging license

External ID Prefix

All valid external ID prefix values must be specified using the configuration rule 'callIdPrefix'. If this is not configured, retrieving call and job history by the External ID of a call may not work properly.

External ID prefix is the first few characters of the external ID and is used to identify the system where the trouble call originated (for example, if external ID is '2389583093' then '2' can be prefix indicating that this call came from Oracle Utilities Customer Self Service). It is also used to guarantee that each call has unique external ID value.

For more information on configuring and working with Oracle Utilities Network Management System, see the Oracle Utilities Network Management System User Guide and Configuration Guide. The chapter called Building the System Data Model in the Configuration Guide contains information about connecting customer data to Oracle Utilities Network Management System electrical model.

Chapter 6

CCB-MDM Integrated Flows

This section provides general information about the functionality and processing of the Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management for Self Service Release. This is an AIA Direct Integration using SOA Suite and does not require the AIA Foundation Pack to be installed.

About the Products

Oracle Utilities Customer Care and Billing

Oracle Utilities Customer Care and Billing (CCB) is a customer and billing system that manages all aspects of customer service needed by most utilities to operate their business.

Oracle Utilities Meter Data Management

Oracle Utilities Meter Data Management (MDM) supports the loading, validation, editing, and estimation (VEE) of meter data - from meter configuration, to meter read and usage validation, to bill determinant calculations.

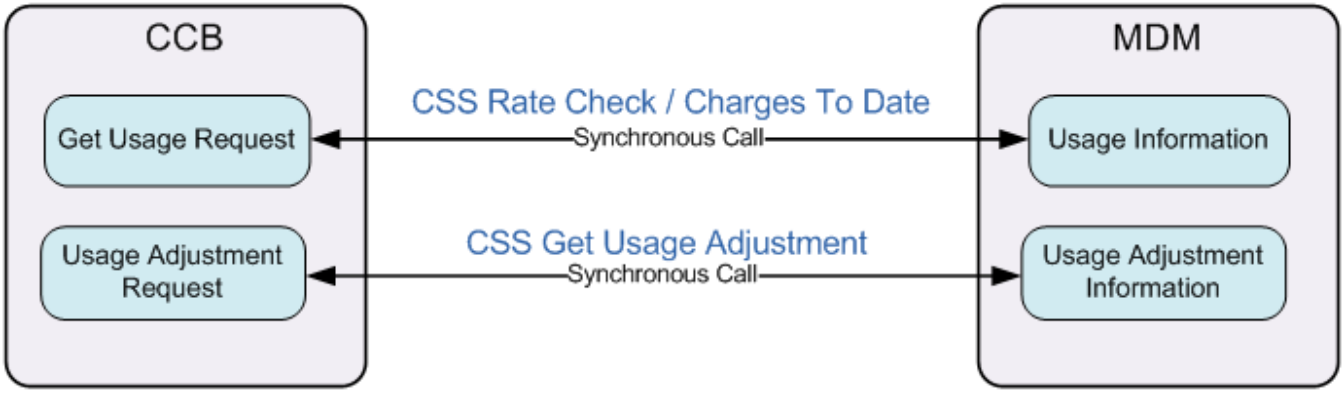
Supported Business Processes

This integration between Customer Care and Billing and Meter Data Management is used to support the business use cases described below for the web self-service solution for Oracle Utilities.

The Business use cases are as follows.

Business Process	Description
Compare Rate Plan and Analysis	Provide a tool where the customer can compare and see the difference to their bill if they should choose to transfer to a different rate plan. Customers also have the option to maintain their current rate plan but change their consumption behavior or change both their current rate plan and consumption behavior to assess the impact on their energy bill.
Current Bill-To and Estimate Graph	A customer will often use a Self Service application to pay their bill. While paying their bill, it would be opportune to present them any unbilled charges to-date. In addition, the customer should be able to see the bill estimate (projection) for the account for the end of the billing period. This estimate should be based on current unbilled activity and extrapolated future usage

This integration will provide a real time interface in CCB to request usage and usage adjustments from MDM.



Understanding the CCB-MDM Integrated Flows use for OUCSS

This section outlines the overall Technical overview, business process and specific integration points handled by this integration.

Technical Overview

This is a direct integration between Oracle Utilities Customer Care and Billing (CCB) and Oracle Utilities Meter Data Management (MDM) to get usage and usage adjustment information.

All the end-to-end integration flows are synchronous.

Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release12.1* (CCB-MDM Integration) for more details. Under Part 1: Understanding the Integration – Chapter 2 Chapter 2 Understanding the Integration Processes – Technical Overview – Synchronous Flows section.

Integration Points

Get Usage Request Integration Flow

Refer to the Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 (CCB-MDM Integration) for more details. Under Part 1: Understanding the Integration – Chapter2 Understanding the Integration Processes – Integration Points – Billing Related Flows – Get Usage Request Integration Flow section.

Usage Adjustment Request Integration Flow

Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for more details. Under Part 1: Understanding the Integration – Chapter 2 Understanding the Integration Processes – Integration Points – Customer Self Service Related Processes – Usage Adjustment Request Integration Flow section.

Notes

- Usage Overview Integration Flow (OUCCB2OUMDM2SSUsageOverviewReqEBF) was deprecated in OUCSS Release 2.1.0. Use the CSS-MDM Direct Usage Overview Integration Flow instead.
- Usage Detail Integration Flow (OUCCB2OUMDM2SSUsageDetailReqEBF) was deprecated in OUCSS Release 2.1.0. Use the CSS-MDM Get Usage Detail Integration Flow instead.

Implementing the CCB-MDM Integrated Flows

Prerequisites

Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 Media Pack must be installed and configured.

Configuring the Integration

This section provides details about the configuration settings required for the integration, and also discusses details related to:

Setting up Oracle Utilities Customer Care and Billing

Setting up Oracle Utilities Meter Data Management

Data Synchronization between CCB and MDM

Oracle Utilities Customer Care and Billing Configuration

Refer to the Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 (CCB-MDM Integration) for more details. Under Part 2: Implementing the Integration Product – Chapter 3 Configuring the Integration – Setting Up Oracle Utilities Customer Care and Billing section.

Oracle Utilities Meter Data Management Configuration

Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for more details. Under Part 2: Implementing the Integration Product – Chapter 3Configuring the Integration – Setting Up Oracle Utilities Meter Data Management section.

Data Synchronization

Oracle Utilities Meter Data Management serves as the database of record for meter device connections and usage while Oracle Utilities Customer Care and Billing manages customers (persons), accounts (service agreements), and service points. The person, SP, SA, meter, meter configuration, and SP-meter history sync integration points add relevant SP/SA and meter data from Oracle Utilities Customer Care and Billing in Oracle Utilities Meter Data Management.

The data synchronization for rates is not completed by the integration product.

See the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release12.2* for data synchronization processes between two systems.

CCB-MDM for Self Service Integration Product Configuration

The following sections describe the configuration needed in the integration pack to meet the requirements for this integration. Configuration steps include setting the following:

Task	Remarks
Setting Configuration Properties	Update the ConfigurationProperties.xml file.
Setting System Properties	Set the Module Configurations properties that are shared by multiple integration flows and Service Configurations properties that are used by a specific BPEL process.
Domain Value Maps	Set the Domain value maps (DVMs) to map codes and other static values across applications.
Error Handling	Set up error notifications.

- Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for more details. Under Part 2: Implementing the Integration Product – Chapter 3 Configuring the Integration – Setting Up the Process Integration section.

Setting Configuration Properties

The ConfigurationProperties.XML file contains properties which can be defaulted in the integration. Also, it contains flags to enable extension points within the integration.

ConfigurationProperties.XML is located in MDS under the directory apps/CCB-MDM/AIAMetaData/config.

Note. Whenever the ConfigurationProperties.XML file is updated, it must be reloaded to MDS for updates to be reflected in the applications or services that use the updated properties. You can perform the reload by rebooting the SOA server.

Setting System Properties

There are two sets of configuration properties described in this section:

- Module Configurations are the properties that are shared by multiple integration flows within this Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 Media Pack. No new properties were introduced for this integration and this integration is not referring to any of the properties in the module configuration.
- Service Configurations are the properties that are used by a specific BPEL process. Shown below are the service configuration for the CCB-MDM Integrated flows use for OUCSS only, refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for the complete service configurations.

Property Name	Default / Shipped Value	Description
Service Name : OUCCB2OUMDM2SSUsageReqEBF		
Default.SystemID	OU_CCB2_01	Initiating system ID.
Extension.PreXformCCB2toMDM2	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCCB2toMDM2	false	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCCB2	false	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCCB2	false	If set to true, the post transformation extension service for the response message is invoked.
MDM2.D2CalculateUsageMultipleRequest.Endpoint .URL		This value is the MDM Service D2CalculateUsageMultipleRequestt Endpoint URL. - Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@ EdgeApplications.OUMDM.ManagedServer.hostname:@E dgeApplications.OUMDM.ManagedServer.portnumber/@E dgeApplications.OUMDM.ManagedServer.context/Webser vices/D2-CalculateUsageMultipleRequests During install, the MDM edge application information will be tokenized to point to the correct MDM server being used
Service Name : OUCCB2OUMDM2SSUsageAdjustmentReqEBF		
Default.SystemID	OU_CCB2_01	Initiating system ID.
Extension.PreXformCCB2toMDM2	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCCB2toMDM2	false	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCCB2	false	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCCB2	false	If set to true, the post transformation extension service for the response message is invoked.

MDM2.WX-UsageAdjustmentRetrieval.Endpoint.URL	This value is the MDM Service WX-UsageAdjustmentRetrieval Endpoint URL. - Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-UsageAdjustmentRetrieval During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.

Domain Value Maps

Domain value maps (DVMs) are a standard feature of the Oracle SOA Suite which maps codes and other static values across applications. For example: “US” and “USA”

DVMs are static in nature, though Administrators can add additional maps as needed. Transactional business processes never update DVMs - they only read from them. They are stored in XML files and cached in memory at runtime.

To maintain information within the domain value maps:

1. Open a browser and access the SOA Composer application.
2. On the SOA Composer, click the “Open” dropdown and select “Open DVM”. This displays a list of all DVM files in MDS.
3. Select the relevant DVM you wish to maintain.
4. Edit the selected DVM. The Edit button in the top navigation bar enables editing the DVM.
5. Once the DVM has been edited, click Save in the navigation bar. This saves the DVM data for that session.
6. Click Commit after updating each DVM. This saves the DVM data in MDS.

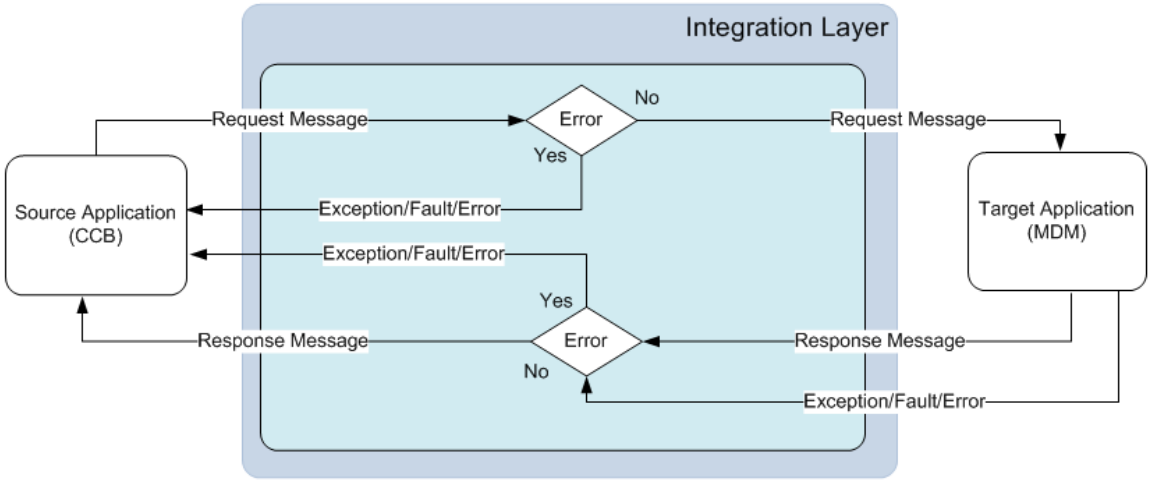
The DVMs used for the CCB-MDM integrated flows for OUCSS are the existing DVMs for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 Media Pack. No new DVMs were introduced for these integration flows.

The lists of DVMs reused for the CCB-MDM integrated flows for OUCSS are as follows:

DVM	Integration Points	Description
OUCCB2_OUMDM2_AllowEstimate	Get Usage Request	Transform CCB Allow Estimate to MDM Allow Estimate and vice versa.
OUCCB2_OUMDM2_BillCondition	Get Usage Request	Transform CCB Bill Condition to MDM Bill Condition and vice versa.
OUCCB2_OUMDM2_UsageRequestMode	Get Usage Request	Transform CCB Usage Calculation Call Type flag to MDM Usage Request Mode flag.
OUMDM2_OUCCB2_IsEstimate	Get Usage Request	Transform MDM Is Estimate flag to CCB Is Estimate flag and vice versa.
OUMDM2_OUCCB2_UOM	Get Usage Request	Transform CCB UOM to MDM UOM and vice versa.

OUMDM2_OUCCB2_TOU	Get Usage Request	Transform CCB TOU to MDM TOU and vice versa.
OUMDM2_OUCCB2_SQI	Get Usage Request	Transform CCB SQI to MDM SQI and vice versa.
OUMDM2_OUCCB2_UsageType	Get Usage Request	Transform the MDM usage type to CCB usage type. For example: Interval, Scalar, etc.
OUMDM2_OUCCB2_SPHowToUse	Get Usage Request	Transform MDM SP How to Use flag to CCB SP How to Use Flag and vice versa
OUMDM2_OUCCB2_MeasuresPeakQuantity	Get Usage Request	Transform MDM Measures Peak Quantity flag to CCB Measures Peak Quantity flag and vice versa

Error Handling



Any exception or error thrown by the integration service is sent back to CCB as a SOAP Fault or exception which will change the outbound message status to be in 'Error'.

Integration service will also send back the exception or SOAP fault received from MDM to CCB. This will also change the outbound message status to be in 'Error'

No email notifications for Business and Technical errors will be sent out from the integration service.

Customization and Extension Methodology

The integration process allows extensibility of transaction messages using the following methods:

- Pretransformation Extension Point
- Posttransformation Extension Point
- Custom Transformations

Pretransformation Extension Point

The pretransformation extension point is invoked before the main transformation is executed. This transformation aids in transforming the source XML coming as an input to the integration process.

The integration layer defines an external call from the pretransformation extension point. This extension point accepts source XML as input and gives the source XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL This can assist the implementation in invoking any external Web service and transform the input XML.

Post Transformation Extension Point

The post transformation extension point is invoked after the main transformation is executed. This transformation aids in transforming the target XML going as an input to the target queue.

The integration layer defines an external call from the post transformation extension point. This extension point accepts the target XML as input and gives the target XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL. This can assist the implementation in invoking any external Web service and transform the output XML.

Custom Transformations

The custom transformations are used to add data to custom elements in the incoming and outgoing messages. The incoming and outgoing messages have custom elements defined in the message. These custom elements refer to a custom XML schema. The main transformation invokes custom transformation.

Empty custom transformation and custom schemas are shipped with the product. The implementation team can add additional fields in the custom schema and map them using the custom transformations.

Using custom transformations allows the implementation to define and pass additional data from the source system to the target system.

Steps to Implement Extension Points

1. Each process in the integration has a pre- and post-transformation extension point which can be used to invoke Web services and transform the payload.
2. The desired extension point can be triggered from the process by enabling it using the ConfigurationProperties.xml pre- and post-transformation extension flags as described in section Setting Configuration Properties.
3. Each process has its own concrete wsdl which is used to read the endpoint location for the extension service.

   These concrete wsdl files are located in MDS under the following directories:

   `/apps/CCB2-MDM2/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCCB2`

   Update the concrete wsdl file to define the binding and service details for the extension service to be called and move the concrete wsdl file to MDS. See the *Sample* below.
4. To move the updated concrete wsdl to MDS, update the appropriate wsdl in the product install home.

   - The directories to put the concrete wsdl in product install home are the following:

```
$PRODUCT_HOME/MDS-Artifacts/CCB2-
MDM2/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCCB2
```

   Then deploy the concrete wsdls to MDS by running the ant deploy command for Deploying MDS folder.

Note: For more information about the command to use to deploying to MDS, see the *Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 Media Pack Installation Guide*, under Deploying MDS Folder section.

5 After deploying the files to MDS, restart the SOA server.

6 After restarting the SOA server, the extension point invokes the Web service in the concrete WSDL.

Sample wsdl File with Binding and Service Details

To enable the extension points for OUCCB2OUMDM2SSUsageReqEBF, add the binding and service elements to the OUCCB2OUMDM2SSUsageOverviewEBF ExtensionConcrete.wsdl, as shown in this sample:

```
<binding name=" OUCCB2OUMDM2SSUsageReqV1ExtensionServiceSOAP11Binding"
         type="ccbext: OUCCB2OUMDM2SSUsageReqV1ExtensionService">
     <soap:binding style="document"
                   transport="http://schemas.xmlsoap.org/soap/http"/>
     <operation name="PreXformCCB2toMDM2">
            <soap:operation style="document"
soapAction="http://xmlns.oracle.com/OUCCB2OUMDM2SSUsageReqEBF/OUCCB2OUMDM2SSUsageReqExten
sion/V1/PreXformCCB2toMDM2"/>
            <input>
                <soap:body use="literal" parts="PreXform_CCB2ToMDM2"/>
            </input>
            <output>
                <soap:body use="literal" parts="PreXform_CCB2ToMDM2"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
     </operation>
     <operation name="PostXformCCB2toMDM2">
            <soap:operation style="document"
soapAction="http://xmlns.oracle.com/OUCCB2OUMDM2SSUsageReqEBF/OUCCB2OUMDM2SSUsageReqExten
sion/V1/PostXformCCB2toMDM2"/>
            <input>
                <soap:body use="literal" parts="PostXform_CCB2ToMDM2"/>
            </input>
            <output>
                <soap:body use="literal" parts="PostXform_CCB2ToMDM2"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
     </operation>
     <operation name="PreXformMDM2toCCB2">
            <soap:operation style="document"
soapAction="http://xmlns.oracle.com/OUCCB2OUMDM2SSUsageReqEBF/OUCCB2OUMDM2SSUsageReqExten
sion/V1/PreXformMDM2toCCB2"/>
            <input>
                <soap:body use="literal" parts="PreXform_MDM2ToCCB2"/>
            </input>
            <output>
                <soap:body use="literal" parts="PreXform_MDM2ToCCB2"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
     </operation>
     <operation name="PostXformMDM2toCCB2">
```

```
            <soap:operation style="document"
soapAction="http://xmlns.oracle.com/OUCCB2OUMDM2SSUsageReqEBF/OUCCB2OUMDM2SSUsageReqExten
sion/V1/PostXformMDM2toCCB2"/>
            <input>
                <soap:body use="literal" parts="PostXform_MDM2ToCCB2"/>
            </input>
            <output>
                <soap:body use="literal" parts="PostXform_MDM2ToCCB2"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
     </operation>
</binding>
<service name=" OUCCB2OUMDM2SSUsageReqV1ExtensionService">
      <port name=" OUCCB2OUMDM2SSUsageReqV1ExtensionService"
            binding="ccbext: OUCCB2OUMDM2SSUsageReqV1ExtensionServiceSOAP11Binding">
            <soap12:address location="http://ouaf.oracle.com:8072/soa-
infra/services/default/CCBMDM2SSExtService/Service1"/>
        </port>
</service>
```

Note: The binding and service can be added easily using the Oracle JDeveloper 12c.

Steps to Implement Custom Transformations

To implement custom transformations:

- Each process in the integration has its own xsd file. The messages have custom elements which can be used to pass additional data from one application to another or vice versa. Refer to the message mappings to see the location of customElements in each message.
- Each process uses two XSD files, one for the Oracle Utilities Customer Care and Billing message and one for the Oracle Utilities Meter Data Management message.
- Each XSD file has a corresponding CustomType XSD file in which the complexType elements for each customElements tag are defined.

Example:

MDM schema file (XSD) for Usage Overview is: OUMDM2SSGetUsageOverview.xsd.

Corresponding custom type schema file (Custom XSD) is: OUMDM2SSGetUsageOverviewCustomType.xsd.

The custom XSD files are located in your product install home under the following directories:

`$PRODUCT_HOME/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCCB2/V1/schemas`

`$PRODUCT_HOME/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/schemas`

- To pass additional elements in the customElements tag, the corresponding complexType in the customType xsd needs to be modified. Add the additional elements required in the complexType elements (see example below).
- Each process has a main transformation which invokes custom templates. Each main transformation file has a corresponding Custom xsl and the custom templates are defined in the Custom xsl.

  Example:

  The transformation file (XSL) for Usage Overview request is:
  `Transformation_CCBtoMDM_UsageOverviewRequest.xsl`

The corresponding custom transformation file (custom XSL) is:
`Transformation_CCBtoMDM_UsageOverviewRequest_Custom.xsl`

The custom XSL files are located in product install home under the following directories:

`$PRODUCT_HOME/ services/industry/Utilities/EBF/<Process Name>/xsl`

- These custom templates are invoked at the location where each customElements tag is present. The Custom xsl can be modified to add transformation for the newly added elements in the custom xsd files. See example below.
- After updating the XSD and XSL files in the product install home, update MDS using the ant deploy command for Deploying MDS folder.

For more information about the command to use to deploying to MDS, refer to the *Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1 Media Pack Installation Guide*, "Deploying MDS Folder" section.

- After deploying the files to MDS, restart the SOA server. When the SOA server is restarted, the changes to the custom xsd and xsl will be reflected in the integration.

Example

In the Usage Request process, to pass requestDetails > request>customElements > userId element from Oracle Utilities Customer Care and Billing to D2-CalculateUsageMultipleRequests > requests >requestsList> customElements > userId element in MDM, the following changes must be implemented:

A In OUCCB2SSUsageRequestCustomType.xsd, add the userId element to the schema. This custom xsd file is located in the `$PRODUCT_HOME/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCCB2/V1/schemas` folder.

```
  <xsd:complexType name=" requestCustomType">
       <xsd:sequence>
            <xsd:element name="userId" type="xsd:string"/>
       </xsd:sequence>
  <xsd:complexType>
```

B In OUMDM2SSCalculateUsageMultipleRequestsCustomType.xsd, add the userId element in the schema. This xsd file is located in the `CCB2-MDM2/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/schemas` folder.

```
  <xsd:complexType name=" requestsListCustomType">
       <xsd:sequence>
            <xsd:element name="userId" type="xsd:string"/>
       </xsd:sequence>
  <xsd:complexType>
```

C Transformation: Transformation_CCBtoMDM_UsageOverviewRequest_Custom.xsl

```
 <xsl:template name=" requestsList-customElements">
   <userId>
     <xsl:value-of
select="/ns0:requestDetails/ns0:request/ns0:customElements/ns0:userId"/>
   </userId>
</xsl:template>
```

Data Mapping

Get Usage Request Mapping

Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for more details. Under Appendix A: Data Mapping – Get Usage Request section.

Usage Adjustment Request Mapping

Refer to the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* (CCB-MDM Integration) for more details. Under Appendix A: Data Mapping – Usage Adjustment Request section.

Chapter 7

CSS Direct BPEL Flows

This section provides general information about the functionality and processing of the Oracle Utilities Self-Service application invoking integration BPEL web services to access MDM or NMS applications. This is an AIA Direct Integration using SOA Suite and does not require the AIA Foundation Pack to be installed.

About the Products

Oracle Utilities Customer Care and Billing

Oracle Utilities Customer Care and Billing (CCB) is a customer and billing system that manages all aspects of customer service needed by most utilities to operate their business

Oracle Utilities Meter Data Management

Oracle Utilities Meter Data Management (MDM) supports the loading, validation, editing, and estimation (VEE) of meter data - from meter configuration, to meter read and usage validation, to bill determinant calculations.

Oracle Utilities Network Management System

Oracle Utilities Network Management System (NMS) processes trouble calls from customers and analyzes them to determine probable outage locationsIt can generate estimated restoration times (ERTs) that can then be provided back to customers. It also keeps a history of all of the customer calls that were entered in the system, as well as a history of all events that were known to affect a customer even if the customer did not call in.

In addition to responding to unplanned outages and non-outage problems, Oracle Utilities Network Management System can help a utility plan maintenance work or new construction that may impact existing customersWhen the detailed switching

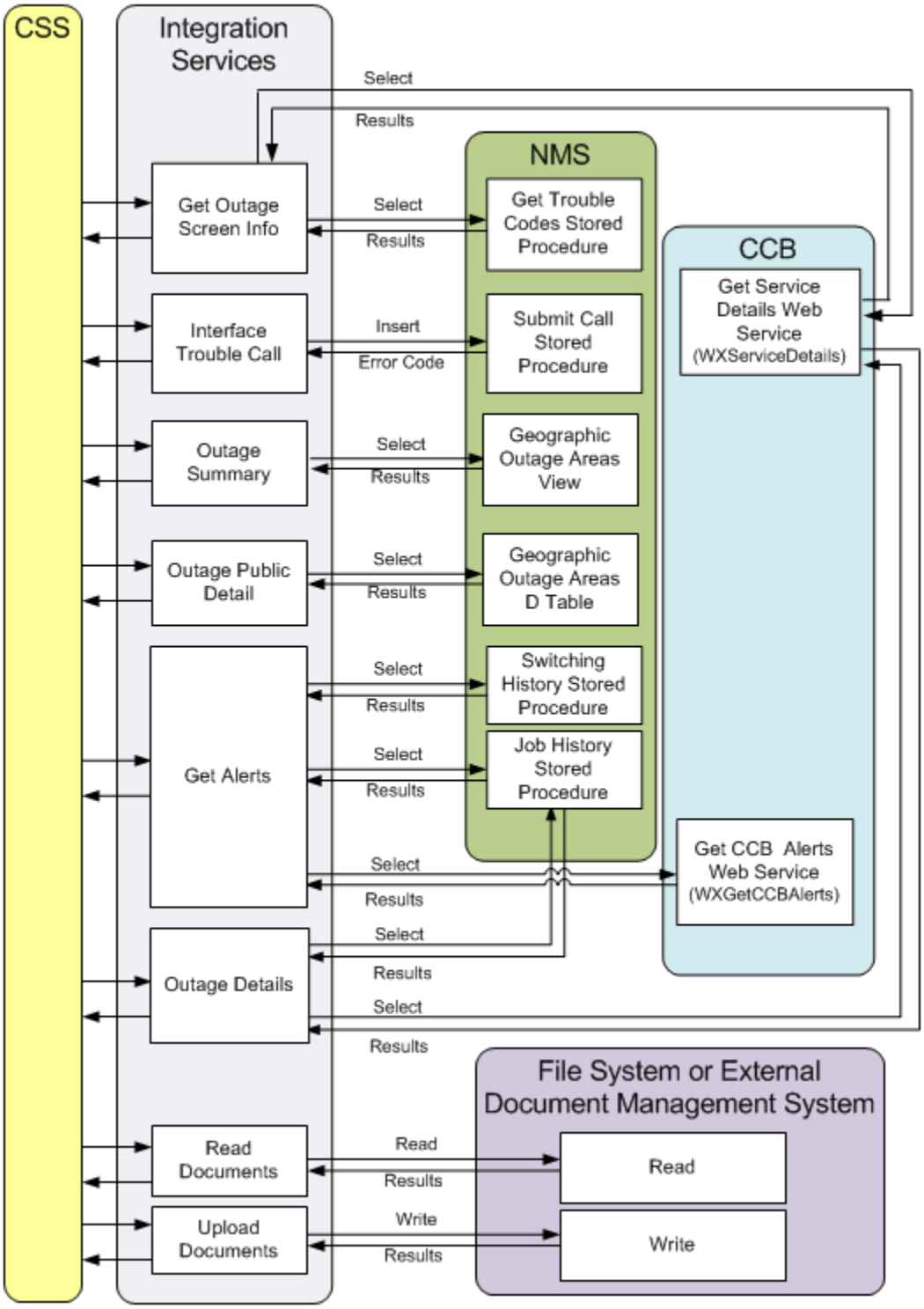
plans are generated in Oracle Utilities Network Management System, information can be provided to customers about planned outages that will impact them.

Supported Business Processes

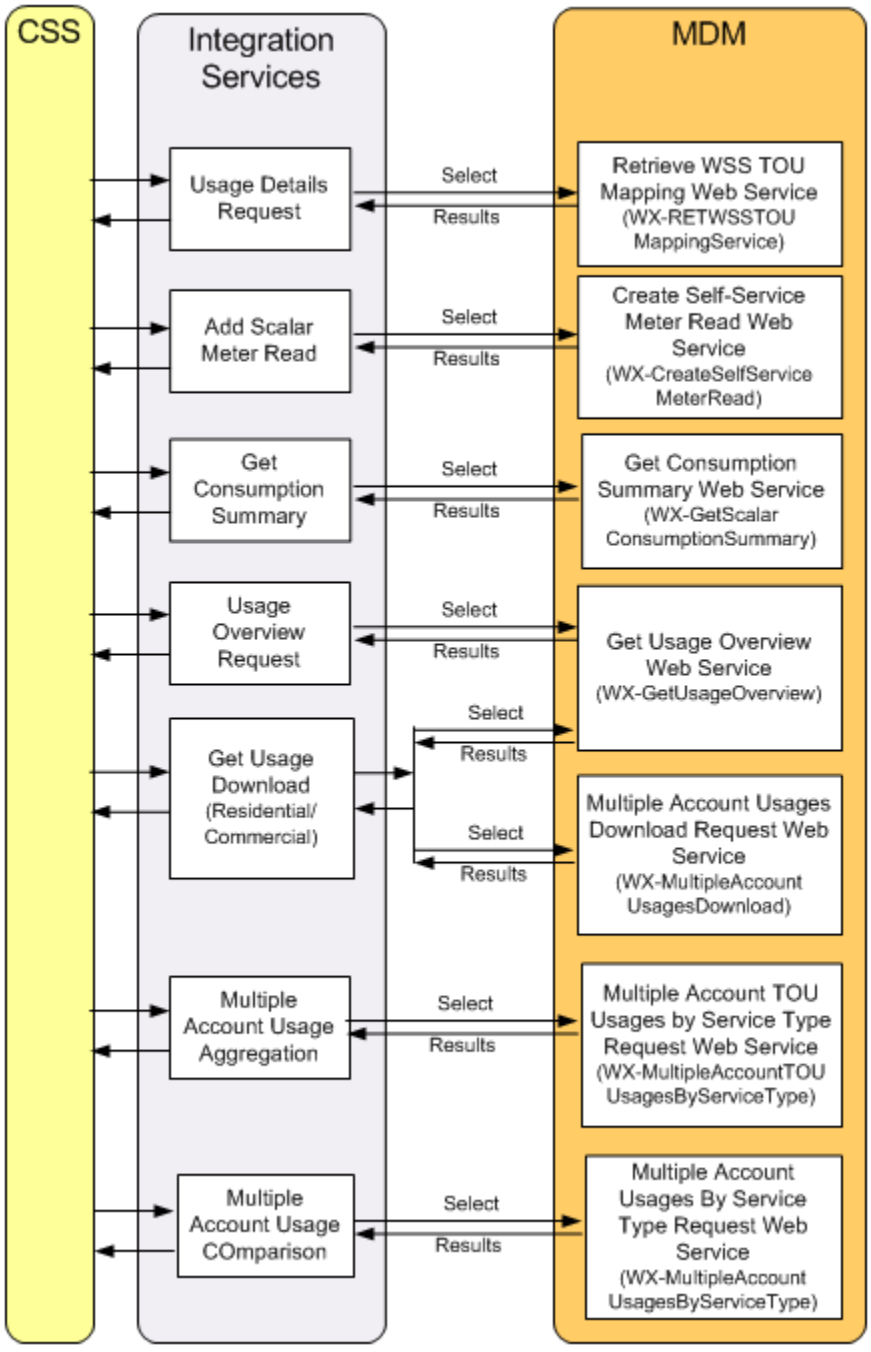
This integration between Self-Service and Meter Data Management or Network Management System is used to support the business use cases described below for the web self-service solution for Oracle Utilities.

The Business use cases are as follows

Business Process	Description
Public Outage Maps and Tables	Provide Web Self-Service users to see general outage information for the utility. The user will have access to a map with details provided as highlights. If the customer is logged in, the outage map should default to the customer's area (county or zip/postal code). The outage information should come from the Utility's network management system (e.g., NMS).
Customer Specific Outage Information	Customers who are logged in to OUCSS should be able to retrieve outage information for their premise. The outage information should come from the Utility's Network Management System (e.g., NMS).
Report an Outage	Customers should be able report an outage for their location using OUCSS. If the customer is logged in, the location should default to the customer's account location. If the customer does not have a self-service account, they should still be able to report the outage by providing the outage location and type of outage. The outage information should be logged in the Utility's network management system (e.g., NMS).
Get Alerts	Customers who are logged in to OUCSS should be able to see alerts relevant to his account. Alerts are retrieved from CCB and NMS.
Usage Overview	Customers who are logged in to OUCSS should be able to see their daily usage for the last *x* days.
Usage Detail	Customers who are logged in to OUCSS should be able to view their usage details. The usage details for a self-service user's account for a certain period is retrieved from MDM.
Usage Download	Ability for residential and commercial customer to download their interval usage data into a standardized format.
Add Scalar Meter Reads to MDM System	Provide out-of-the-box support for scalar meter reads stored in MDM. Customers should be able to enter their scalar meter read using OUCSS and the meter read information should be stored in MDM.
Consumption Summary	Provide scalar meter consumption summary stored in MDM.
Multiple Account Usage Aggregation	To be able to display aggregated usage information for commercial customers in the CSS Portal. It should have the ability to aggregate up to "N" accounts into a single display, where a configuration parameter defines the value of "N".
Multiple Account Usage Comparison	To give commercial customers a method by which they can compare usage for their properties in a single view.
Read Documents	Customers who are logged in to OUCSS should be able to read any electronic documents previously stored and render it in CSS
Upload Documents	Customers who are logged in to OUCSS should be able to upload electronic documents when reporting an issue.



Direct CSS Integrations (CSS to CCB, NMS or File System)



Direct CSS Integrations (CSS to MDM)

Understanding the CSS Direct BPEL Flows

This section outlines the overall Technical overview and specific integration points handled by this integration.

Technical Overview

- This is a direct integration between Oracle Utilities Customer Self-Service (OUCSS) and the following applications:
  - ***Oracle Utilities Meter Data Management (MDM)*** to get usage overview information, usage details, usage download data, consumption summary and to add a scalar meter read to MDM.
  - ***Oracle Utilities Network Management (NMS)*** to get the outage summary, submit an outage call, get the account's outage details and get the outage alerts.
  - ***Oracle Utilities Customer Care and Billing (CCB)*** to get the account's service details and alerts coming from CCB.
  - ***File System or External Document Management System*** to store and retrieve electronic documents stored when reporting an issue.
- All the end-to-end integration flows are synchronous from CSS to the integration layer.
- One BPEL process manages each integration point and the BPEL flow is exposed as a web service. The BPEL Process handles the following:
  - Transform the request message coming from the source (CSS) application to the target application's (e.g., NMS/MDM) format and invoke the MDM web service or NMS Stored Procedure synchronously.
  - Create DB Adapters to interact with the NMS Database to invoke the NMS stored procedure to select or insert data in NMS or to query an NMS table/view.
  - Invoke CCB web service to get additional data, when needed.
  - Receives the response message coming from the target application (e.g., NMS, MDM) and transforms the message to the source application's (CSS) format.
  - Error handling and optional error notification, when applicable.
  - Handles message extensions through custom xsl or by using extension points.

Note: No DVMs are used in the request or response transformations.

Integration Points

CSS-NMS Flows

Outage Summary Integration Flow

Business Details

This process is used to get the outage summary information for the Outage Summary information screen in CSS from the Utility's network management system (e.g., NMS).

CSS will send a request to the integration BPEL service to get outage information needed to for the Outage Summary Screen. The BPEL service returns all the outages in the NMS system aggregated by area type (City, Zip etc.) to Self-Service and outages in the NMS system aggregated by area type and selected area to CSS

Technical Details

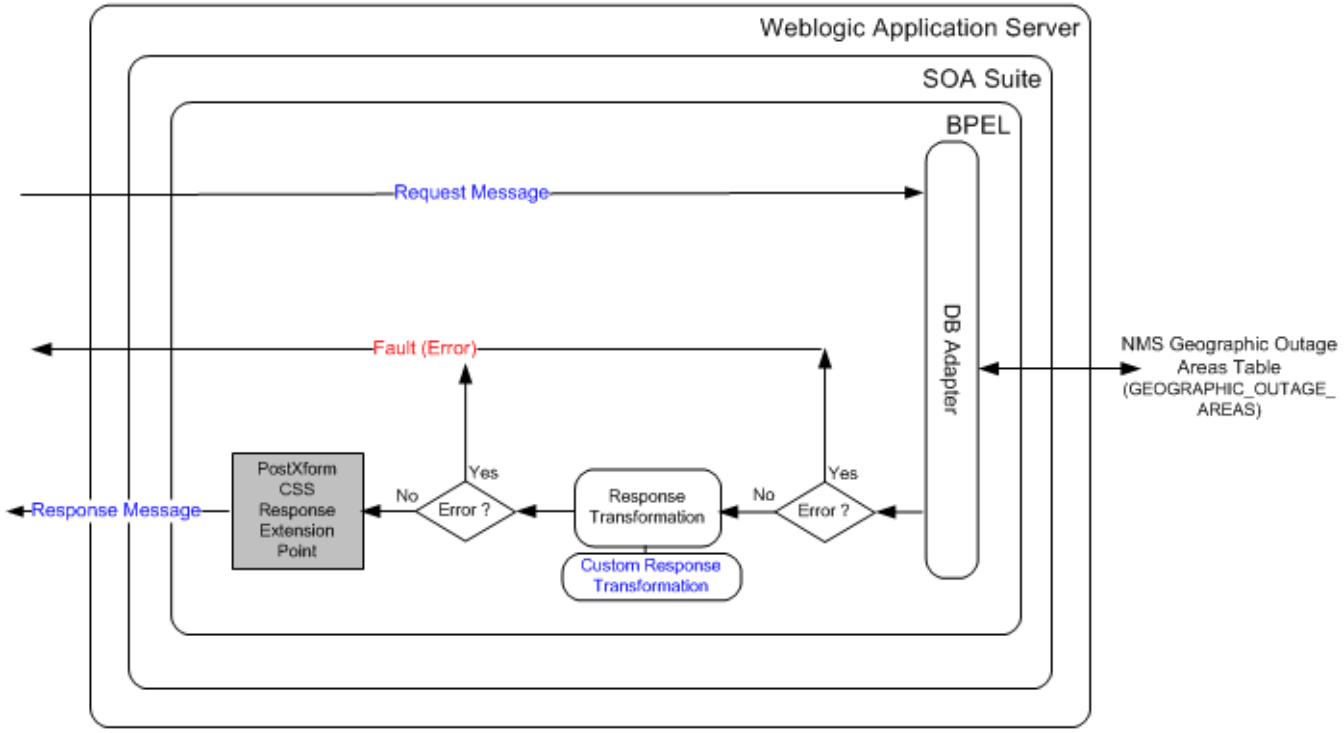
When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation

  - CSS request message does not pass anything to NMS so no request transformation is needed.
- Get Outage Summary from NMS
  - The Integration BPEL process queries the NMS Geographic Outage Areas Views to get the outage summary.
- Response Message Transformation
  - The Integration BPEL process transforms and passes back the NMS outage summary back to the CSS response format.
- Error Handling
  - When a business or technical error is encountered in NMS, or in the Integration, a fault will be returned to CSS with a specific message code. The message codes are obtained from the Configuration properties file.
- Customization
  - If the extension point flag (***Extension.PostXformOUCSStoOUNMS)*** is enabled, it will invoke the PostXform CSS Response Custom Extension Service. The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUNMSOutageSummaryEBF	Self Service Outage Summary BPEL Process Synchronous BPEL process that transform incoming CSS request message to NMS format and retrieve outage summary information from NMS. Transform the response coming from NMS back to CSS format.

External Service Call

DB Adapter Service

Name	Description	NMS Table/View
OUNMSOutageSummaryService	This adapter service invokes the NMS Geographic Outage Areas view to get the outage summary in NMS.	GEOGRAPHIC_OUTAGE_AREAS

Outage Public Detail Integration Flow

Business Details

This process is used to get the outage detail information for the selected Area chosen from the Outage Summary information screen in CSS. The outage detail information for the selected area is obtained from the Utility's network management system (e.g., NMS).

CSS will send a request with selected area to the integration BPEL service to get outage detail information to display on the Outage Summary Screen. The BPEL service returns all the outages in the NMS system for the selected area type (City, Zip, etc.) and selected area to Self-Service.

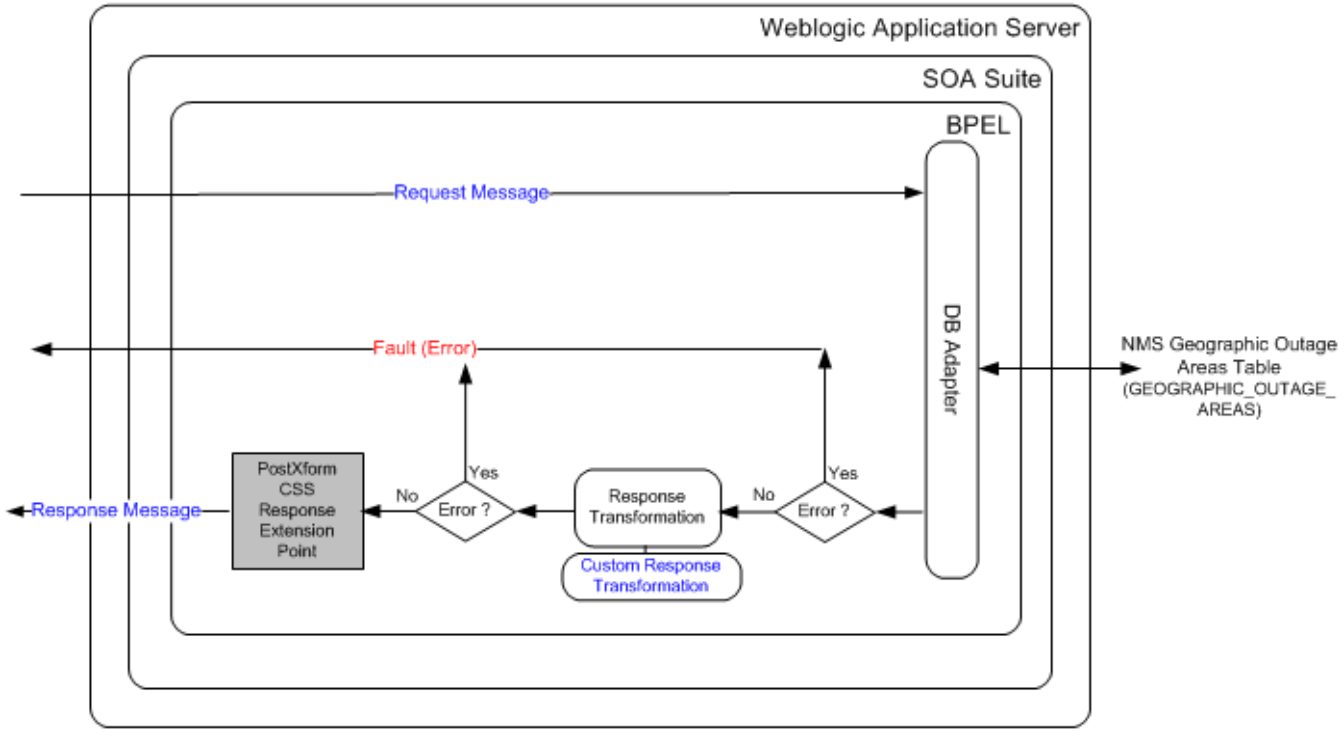
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent NMS input fields to select the outages for a specific area type and area.
- Get Outage Summary from NMS
  - The Integration BPEL process queries the NMS Geographic_Outage_Areas_D materialized View to get the outage details for a specific area.
- Response Message Transformation
  - The Integration BPEL process transforms and passes back the NMS outage summary back to the CSS response format.
- Error Handling
  - When a business or technical error is encountered in NMS, or in the Integration, a fault will be returned to CSS with a specific message code. The message codes are obtained from the Configuration properties file.
- Customization
  - If the extension point flag (***Extension.PostXformOUCSStoOUNMS)*** is enabled, it will invoke the PostXform CSS Response Custom Extension Service. The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUNMSOutagePublicDetailEBF	Self Service Outage Public Detail BPEL Process Synchronous BPEL process that accepts the CSS request message to get the outage details from NMS for Selected Area Type and Area. The response from NMS will be sent back to CSS after appropriate transformations.

External Service Call

DB Adapter Service

Name	Description	NMS Table
OUNMSOutagePublicDetailDBService	This adapter service selects from GEOGRAPHIC_OUTAGE_AREAS_D table from NMS to get the outage details for a given zip or county.	GEOGRAPHIC_OUTAGE_AREAS_D

Get Outage Screen Info Integration Flow

Business Details

This process is used to get the outage screen information when a user wants to Report an Outage for a premise or Report a Public Outage.

CSS will send a request to the integration BPEL service to get outage screen information needed to report an outage. The BPEL service returns the trouble codes defined in NMS back to Self-Service.

If a customer is reporting an outage for his location or for a known premise, CSS also sends the account id as part of the request message and the BPEL service will also return the account information needed for the report an outage screen back to Self-Service.

Notes:

Trouble Codes definition in NMS needs to be the same in CSS to ensure that the trouble code sent from CSS when reporting an outage is interpreted similarly when the trouble code is received by NMS.

Note: The Account Information from CCB is only retrieved if CSS passes an account id to the process.

Technical Details

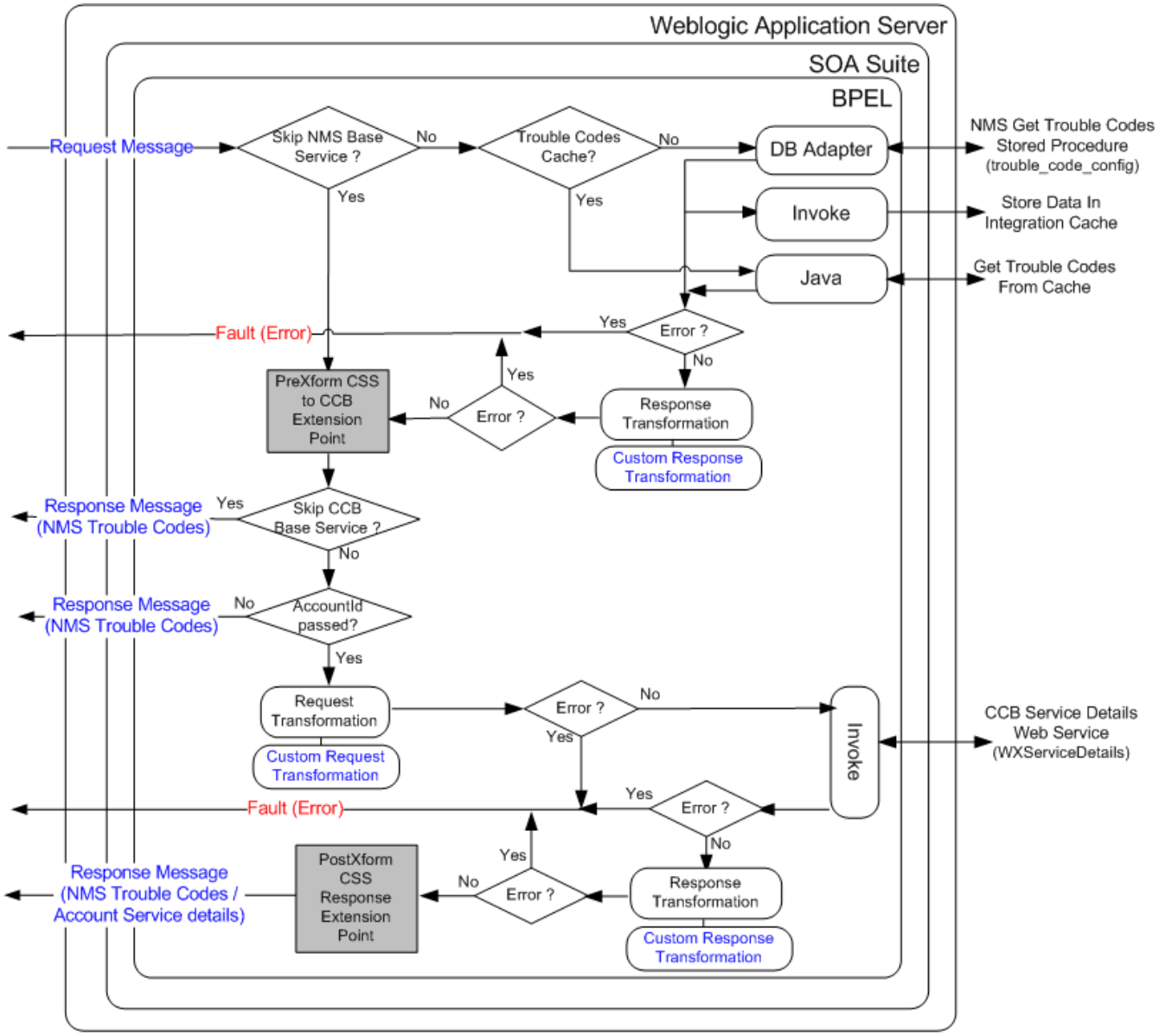
When the integration BPEL service receives a request from CSS, it will do the following:

- Check Skip NMS Base Service Flag
  - The Skip NMS Base Service Flag (***NMS.SkipBaseServiceFlag***) is defined in the ConfigurationProperties xml file. The value is defaulted to false so the NMS Get Trouble Codes stored procedure can be invoke to get the trouble code values.
  - If the customer does not want to invoke the NMS base stored procedure, this flag must be changed to true so the BPEL process will not call the NMS stored procedure and no trouble codes will be returned from NMS.
- Check Skip CCB Base Service Flag
  - The Skip CCB Base Service Flag (***CCB.SkipBaseServiceFlag***) is defined in the ConfigurationProperties xml file. The value is defaulted to false so the CCB Get Service Details web service can be invoke to get the account and service information of the account provided.
  - If the customer does not want to invoke the CCB web service, this flag must be changed to true so the BPEL process will not call the CCB web service and no account information will be returned from CCB.
- Request Message Transformation
  - Transforming the CSS request message to the equivalent CCB request message format. NMS does not have any inputs so no request transformation is needed.
- Get Trouble Codes
  - The Integration BPEL process will call the NMS Get Trouble Code Stored Procedure to get the trouble codes if the NMS trouble codes are not cache in the integration layer. After retrieving the codes from NMS, the BPEL process will store the trouble codes in the integration cache and send the response back to CSS.
  - If NMS trouble codes are stored in the integration cache, then BPEL process will just get the trouble codes from the integration cache. It does not need to call the NMS stored procedure. When the integration server is bounce, the trouble code values stored in the integration cache will be deleted.
  - If the values of the NMS trouble codes changed in NMS, the integration server has to be bounced to clear the integration cache so the BPEL process will invoke the NMS stored procedure to get the latest values.
- Get Account and Service Information
  - If the CSS request message passed an account id, the BPEL process will invoke CCB to get the account id's account and service information.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to CCB and CCB returns the language-related account and service elements in the appropriate language passed.
  - If no record were found for the account, CCB will return an empty response. Otherwise, it will return the account and service information back to the integration.

- Response Message Transformation
  - Transforming the NMS response and CCB response back to the CSS response message format and send the response back to CSS.
- Error Handling
  - When a business or technical error is encountered in NMS, CCB or in the Integration, a soap fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.
- Customization
  - If the extension point flag (***Extension.PreXformOUCSStoOUCCB)*** is enabled, it will invoke the PreXform CSS to CCB Custom Extension Service.
  - If the extension point flag (***Extension.PostXformOUCSStoOUCCB)*** is enabled, it will invoke the PostXform CSS to CCB Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSGetOutageScreenInfoEBF	Self Service Get Outage Screen Information BPEL Process Synchronous BPEL process that accepts the CSS request message to get the outage screen info from NMS and CCB. BPEL calls NMS stored procedure to get the trouble codes and sends it back to CSS. Optionally, if the account id is provided, BPEL calls the CCB Get Service Details web service to get the account information and pass it back to CSS

External Service Call

DB Adapter Service

Name	Description	NMS Stored Procedure
OUNMSGetTroubleCodes	This adapter service invokes the NMS Get Trouble Code Stored Procedure to get the trouble codes configuration defined in NMS.	trouble_code_config (in NMS Package PK_CCB)

Web Services

Application	IWS Service Name	Description
CCB	WXServiceDetails	Retrieve Service Details This inbound service is used by the self-service application. It's responsible for retrieving service details for outage reporting. The following is currently returned for the input account: •Customer information (name and phone number list) •Premise list which includes the following: oPremise ID, address field constituents and premise info SP list which includes SP ID, SP type and SP type description

Trouble Calls Interface Integration Flow

Business Details

This process is used to submit an outages or trouble calls reported in CSS to NMS.

NMS is the owner of the trouble calls data and all outages submitted in CSS are stored in NMS. No outage or trouble call data are stored in the CSS system.

From CSS, the user is allowed to report the following outages:

- If the customer has a self-service account, he can report an outage at his location or premise. When CSS sends a request to the integration, the Service Point Id (SP Id) must be provided..
- If the customer does not have a self-service account, he can still report a public outage or an outage at some other location (including non-premise outages such as street lights). When CSS sends a request to the integration, the SP Id will be blank and this outage will be considered a fuzzy trouble call in NMS.

Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

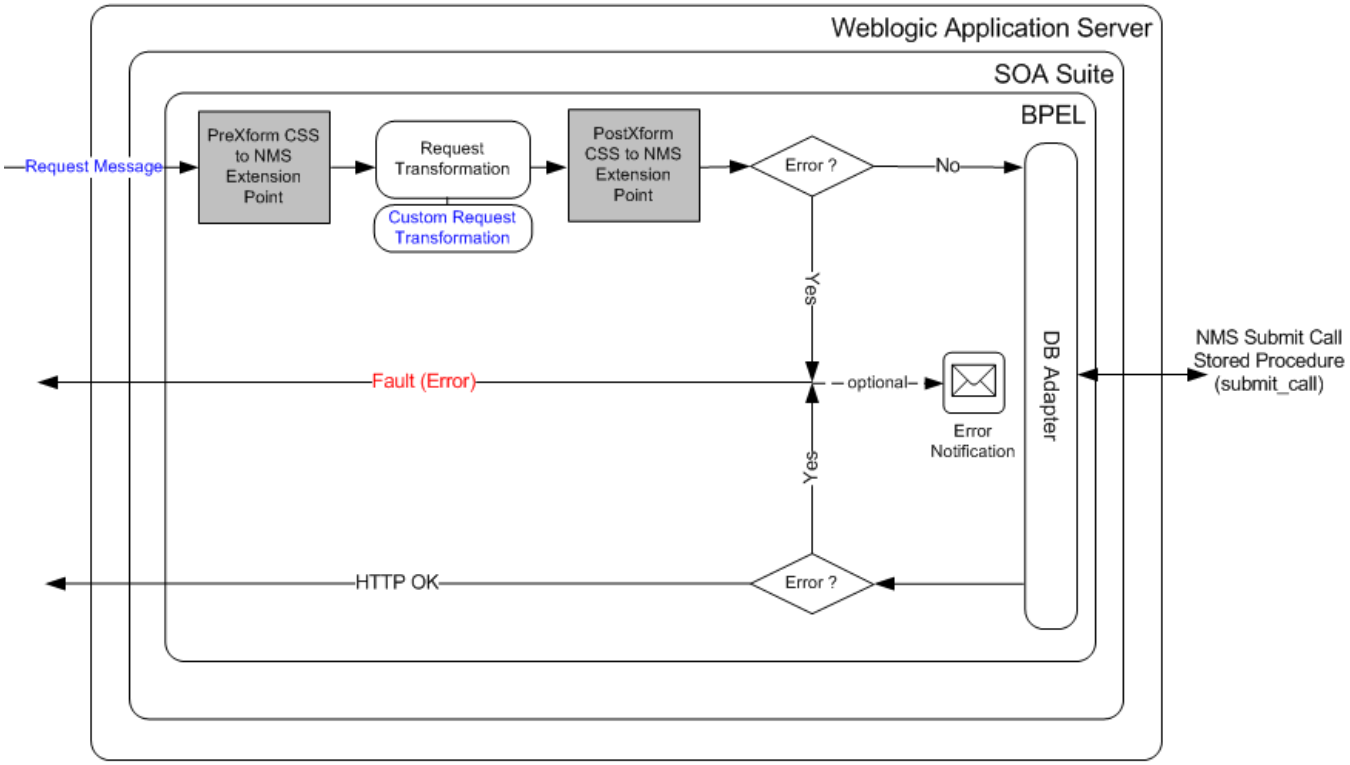
- Request Message Transformation
  - Transforming the CSS input message to the equivalent NMS input fields that will be used by the trouble calls stored procedure
- Insert Trouble Calls (Outage)
  - The Integration BPEL process will use the DB Adapter to interact with the NMS Database to invoke the NMS trouble calls stored procedure that inserts the trouble call record to the Trouble Calls table
  - When the insert to the NMS Trouble Call table is successful the integration layer will synchronously send a positive acknowledgement back to CSS.

  - If the values of the NMS trouble codes changed in NMS, the integration server has to be bounced to clear the integration cache so the BPEL process will invoke the NMS stored procedure to get the latest values.

- Error Handling
  - When a business or technical error is encountered in NMS or in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.
- Email Notification (optional)
  - As an option, email notification can also be setup to send the errors out in an email to a designated user. By default, no email notification will be setup for this integration.
  - The Business Error Notification Flag (BusinessError.NotificationFlag) and Technical Error Notification Flag (TechnicalError.NotificationFlag) are defined in the ConfigurationProperties xml file. The values are defaulted to false so email notification when errors are encountered is disabled.
- Customization
  - If the extension point flag (Extension. PreXformCSStoNMS) is enabled, it will invoke the PreXform CSS to NMS Custom Extension Service.
  - If the extension point flag (Extension. PostXformCSStoNMS) is enabled, it will invoke the PostXform CSS to NMS Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUNMSTroubleCallInterfaceEBF	CSS-NMS Trouble Call Interface BPEL Process Synchronous BPEL process that transforms incoming CSS trouble call message to NMS format and insert the trouble call record in NMS. The BPEL process will include transformations, extensions and error notifications.

External Service Call

DB Adapter Service

Name	Description	NMS Stored Procedure
OUNMSSubmitCallStoredProcedure	This adapter service invokes the NMS Submit Call Stored Procedure to create outage in NMS.	submit_call (in NMS Package PK_CCB)

Get Alerts Integration Flow

Business Details

This process is used to get the alerts available for an account from CCB and also check in NMS if the customer has a current or pending outage(s) in any of his locations and send back the outage alerts to CSS.

When a customer logs into CSS, CSS will send a request to the integration BPEL service to get alerts related to the user's account. Integration will call CCB and NMS to get the alerts related to the account and respond back to CSS with a list of alerts for his account.

Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Check Skip CCB Base Service Flag
  - The Skip CCB Base Service Flag (***CCB.SkipBaseServiceFlag***) is defined in the Configuration Properties xml file. The value is defaulted to false so the CCB Get Alerts web service can be invoke to get the alert information of the account provided.
  - If the customer does not want to invoke the CCB web service, this flag must be changed to true so the BPEL process will not call the CCB web service and no alert information will be returned from CCB.
- Check Skip NMS Planned Outage Base Service Flag
  - The Skip NMS Planned Outage Base Service Flag (***NMS.SkipPlanOutBaseServiceFlag***) is defined in the Configuration Properties xml file. The value is defaulted to false so the NMS Switching History stored procedure can be invoke to get the planned outage(s) related to the account provided.
  - If the customer does not want to invoke the NMS base stored procedure, this flag must be changed to true so the BPEL process will not call the NMS stored procedure and no planned outage will be returned from NMS.
- Check Skip NMS Current Outage Base Service Flag
  - The Skip NMS Current Outage Base Service Flag (***NMS.SkipPlanOutBaseServiceFlag***) is defined in the Configuration Properties xml file. The value is defaulted to false so the NMS Job History stored procedure can be invoke to get the current outage(s) related to the account provided.

  - If the customer does not want to invoke the NMS base stored procedure, this flag must be changed to true so the BPEL process will not call the NMS stored procedure and no current outage will be returned from NMS.

- Request Message Transformation
  - Transforming the CSS request message to the equivalent CCB request message format and NMS input fields to be used by the stored procedure.
- Get CCB Alerts
  - The BPEL process invokes CCB and gets the alerts related to the account id provided.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to CCB and CCB returns the alerts in the appropriate language passed.
  - If no alerts were found for the account, CCB will return an empty list. Otherwise, it will return a list of CCB alerts back to the integration.
- Get NMS Current Outage
  - The BPEL process uses the DB Adapter to interact with the NMS Database to invoke the NMS Job History stored procedure to get the current outage for the account.
  - NMS Job History stored procedure accepts the account id and number of days of history parameters. If the number of days of history parameter is null, NMS will only return the current active outages back to the integration. Since CSS only needs the current outages, integration only needs to pass the account id and leave the number of days of history to null in the request message.
  - CSS Language Code is not passed to NMS since NMS does not support Multilanguage.
  - If no current outages were found, NMS will return an empty list. Otherwise it will return a list of current outages for the account requested back to the integration.
  - When NMS returns a current outage back to the integration, integration builds the current outage alert by retrieving the values of the following properties from the Configuration Properties xml file and pass it back to CSS:
    - NMS Current Outage Alert Header Text (NMS.CurrentOutageAlertHeader)
    - NMS Current Outage Alert Text (NMS.CurrentOutageAlertText)
    - NMS Current Outage Alert Icon Code (NMS.CurrentOutageAlertIconCode)
    - NMS Current Outage Alert Link Code (NMS.CurrentOutageAlertLinkCode)
    - NMS Current Outage Alert Type (NMS.CurrentOutageAlertType)

Note: Only one alert is created for the Current Outage, regardless whether NMS returns more than one current outage.

- Get NMS Planned Outage(s)
  - The BPEL process uses the DB Adapter to interact with the NMS Database to invoke the NMS Switching History stored procedure to get the planned outage for the account.
  - NMS Switching History stored procedure accepts the account id and number of days of history parameters. If the number of days of history parameter is null, NMS will only return the current and future planned outages back to the integration. Since CSS only needs the current and future planned outages, integration only needs to pass the account id and leave the number of days of history to null in the request message.
  - CSS Language Code is not passed to NMS since NMS does not support Multilanguage.
  - If no planned outages were found, NMS will return an empty list. Otherwise it will return a list of planned outages for the account requested back to the integration.
  - When NMS returns the planned outages back to the integration, integration builds the current outage alert by retrieving the values of the following properties from the Configuration Properties xml file and pass it back to CSS:
    - NMS Pending Outage Alert Header Text (NMS.PendingOutageAlertHeader)

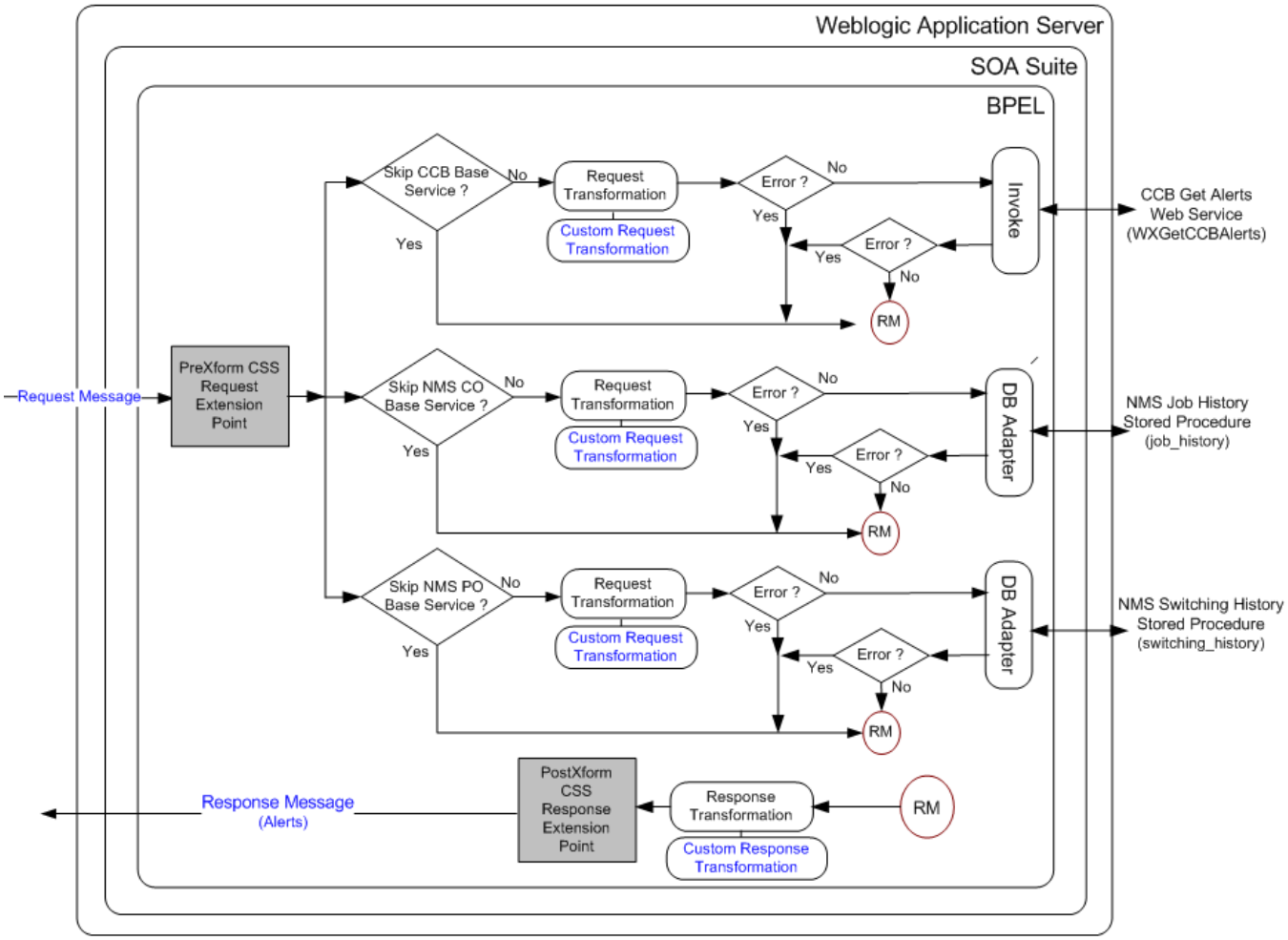
- NMS Pending Outage Alert Text (NMS.PendingOutageAlertText)
- NMS Pending Outage Alert Icon Code (NMS.PendingOutageAlertIconCode)
- NMS Pending Outage Alert Link Code (NMS.PendingOutageAlertLinkCode)
- NMS Pending Outage Alert Type (NMS.PendingOutageAlertType)

Note: The Alert Text sent back to CSS is a concatenation of NMS.PendingOutageAlertText and the start date of the planned outage coming from NMS. (Example: Your service will have a planned outage on ***date/time***).

- Response Message Transformation
  - The BPEL process will transform and pass back the CCB and NMS alerts back to the CSS response format.
- Error Handling
  - Any error encountered in the integration or any errors received from CCB or NMS, integration will not send a fault or log it as an error in the response message. It will just ignore the error.
- Customization
  - If the extension point flag (***Extension.PreXformCSS)*** is enabled, it will invoke the PreXform CSS Request Custom Extension Service.
  - If the extension point flag (***Extension.PostXformCSS)*** is enabled, it will invoke the PostXform CSS Response Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSGetAlertsEBF	Self Service Get Alerts BPEL Process Synchronous BPEL process that orchestrate getting the CSS Alerts from CCB and NMS. This BPEL process will receive the CSS request messages and invoke CCB Web Service and NMS stored procedures to get the account related alerts for the CSS users. The response from CCB and NMS will be sent back to CSS after appropriate transformations.

External Service Call

DB Adapter Service

Name	Description	NMS Stored Procedure
OUNMSCurrentOutageAdapterService	This adapter service invokes the Job History Stored Procedure to check if the account has an active outage in NMS.	job_history (in NMS Package PK_CCB)

OUNMSPendingOutageAdapterService	This adapter service invokes the Switching History Stored Procedure to check if the account has an active or pending planned outage in NMS.	switching_history (in NMS Package PK_CCB)

Web Services

Application	IWS Service Name	Description
CCB	WXGetCCBAlerts	This inbound service retrieves a list of alerts to display in the self-service application. The list of alert types and corresponding scripts are defined on the Self-Service Integration master configuration.

Outage Detail Integration Flow

Business Details

This process is used to get the outage details for an account from NMS and the account's service details from CCB and send back the information to CSS.

Technical Details

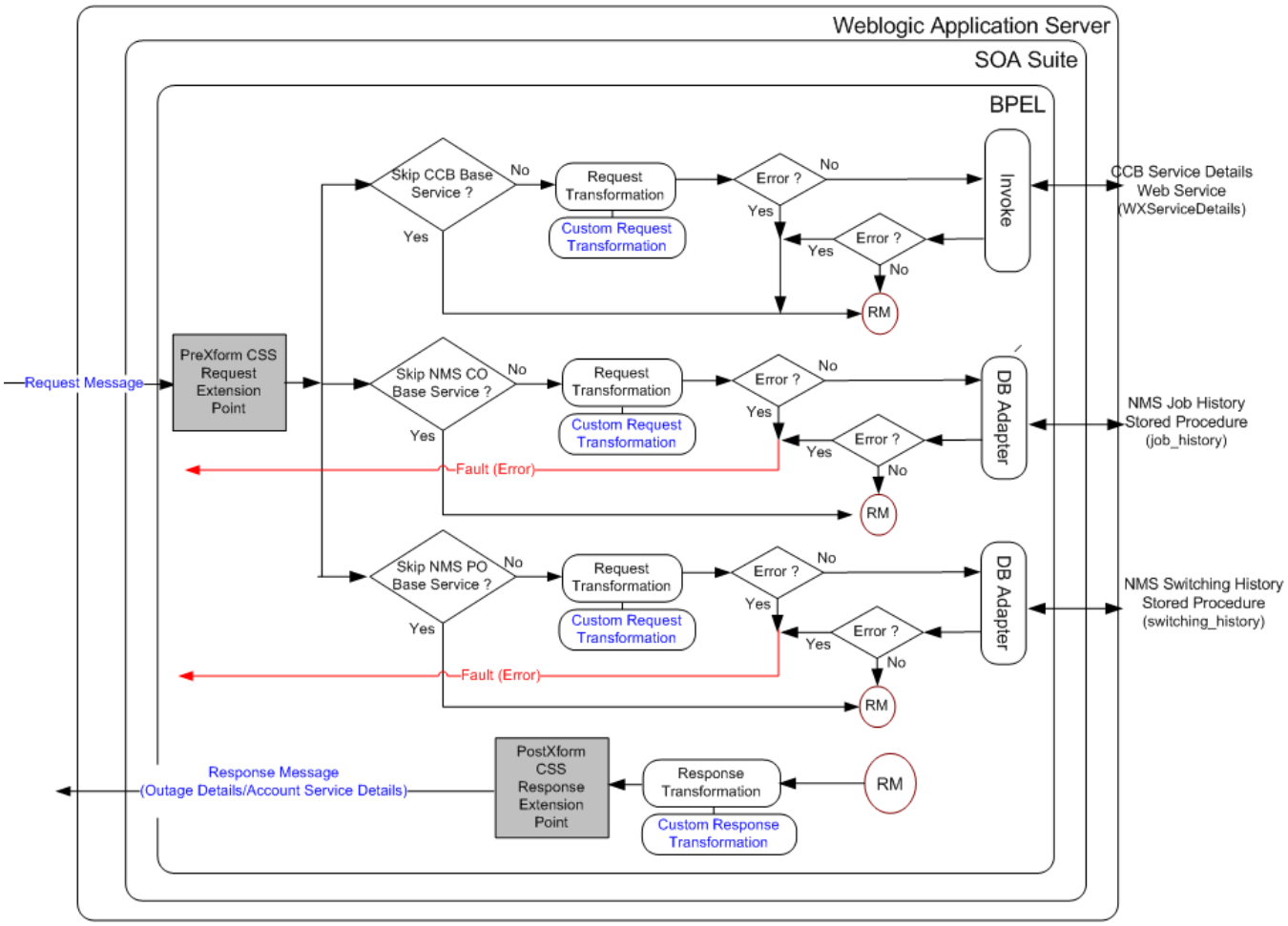
When the integration BPEL service receives a request from CSS, it will do the following:

- Check Skip CCB Base Service Flag
  - The Skip CCB Base Service Flag (***CCB.SkipBaseServiceFlag***) is defined in the ConfigurationProperties xml file. The value is defaulted to false so the CCB Get Service Details web service can be invoke to get the account and service information of the account provided.
  - If the customer does not want to invoke the CCB web service, this flag must be changed to true so the BPEL process will not call the CCB web service and no account information will be returned from CCB.
- Check Skip NMS Planned Outage Base Service Flag
  - The Skip NMS Planned Outage Base Service Flag (***NMS.SkipPlanOutBaseServiceFlag***) is defined in the Configuration Properties xml file. The value is defaulted to false so the NMS Switching History stored procedure can be invoke to get the planned outage(s) related to the account provided.
  - If the customer does not want to invoke the NMS base stored procedure, this flag must be changed to true so the BPEL process will not call the NMS stored procedure and no planned outage will be returned from NMS.
- Check Skip NMS Current Outage Base Service Flag
  - The Skip NMS Current Outage Base Service Flag (***NMS.SkipPlanOutBaseServiceFlag***) is defined in the Configuration Properties xml file. The value is defaulted to false so the NMS Job History stored procedure can be invoke to get the current outage(s) related to the account provided.
  - If the customer does not want to invoke the NMS base stored procedure, this flag must be changed to true so the BPEL process will not call the NMS stored procedure and no current outage will be returned from NMS.
- Request Message Transformation
  - Transforming the CSS request message to the equivalent CCB request message format and NMS input fields to be used by the stored procedure.
- Get Account and Service Information

  - If the CSS request message passed an account id, the BPEL process will invoke CCB to get the account id's account and service information.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to CCB and CCB returns the language-related account and service elements in the appropriate language passed.
  - If no record were found for the account, CCB will return an empty response. Otherwise, it will return the account and service information back to the integration.
- Get NMS Current Outage
  - The BPEL process uses the DB Adapter to interact with the NMS Database to invoke the NMS Job History stored procedure to get the current outage for the account.
  - NMS Job History stored procedure accepts the account id and number of days of history parameters. If the number of days of history parameter is null, NMS will only return the current active outages back to the integration. Since CSS only needs the current outages, integration only needs to pass the account id and leave the number of days of history to null in the request message.
  - CSS Language Code is not passed to NMS since NMS does not support Multilanguage.
  - If no current outages were found, NMS will return an empty list. Otherwise it will return a list of current outages for the account requested back to the integration.
- Get NMS Planned Outage
  - The BPEL process uses the DB Adapter to interact with the NMS Database to invoke the NMS Switching History stored procedure to get the planned outage for the account.
  - NMS Switching History stored procedure accepts the account id and number of days of history parameters. If the number of days of history parameter is null, NMS will only return the current and future planned outages back to the integration. Since CSS only needs the current and future planned outages, integration only needs to pass the account id and leave the number of days of history to null in the request message.
  - CSS Language Code is not passed to NMS since NMS does not support Multilanguage.
  - If no planned outages were found, NMS will return an empty list. Otherwise it will return a list of planned outages for the account requested back to the integration.
- Response Message Transformation
  - The BPEL process will transform and pass back the CCB account service details and NMS outage details back to the CSS response format.
- Error Handling
  - When a business or technical error is encountered in NMS, or in the Integration, a fault will be returned to CSS with a specific message code. The message codes are obtained from the Configuration properties file.
  - When a business or technical error is encountered in CCB, the fault or error will just be ignored and only the NMS Outage Details will be returned back to CSS.
- Customization
  - If the extension point flag (***Extension. PreXformOUCSSReq)*** is enabled, it will invoke the PreXform CSS Request Custom Extension Service.
  - If the extension point flag (***Extension. PostXformOUCSSResponse)*** is enabled, it will invoke the PostXform CSS Response Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUNMSOutageDetailEBF	Self Service Outage Detail BPEL Process Synchronous BPEL process that accepts the CSS request message to get the outage details from NMS and get the account service details from CCB. The response from CCB and NMS will be sent back to CSS after appropriate transformations.

External Service Call

DB Adapter Service

Name	Description	NMS Stored Procedure
OUNMSJobHistStPrCurrentOutageDetailService	This adapter service invokes the Job History Stored Procedure to get the account's active outage details from NMS.	job_history (in NMS Package PK_CCB)

OUNMSGetPlannedOutageDetailService	This adapter service invokes the Switching History Stored Procedure to get the account's planned outages details from NMS.	switching_history (in NMS Package PK_CCB)

Web Services

Application	IWS Service Name	Description
CCB	WXServiceDetails	Retrieve Service Details This inbound service is used by the self-service application. It's responsible for retrieving service details for outage reporting.

CSS-MDM Flows

Get Usage Detail Integration Flow

Business Details

This process is used to get the usage details for an account from MDM and send back the information to CSS

Technical Details

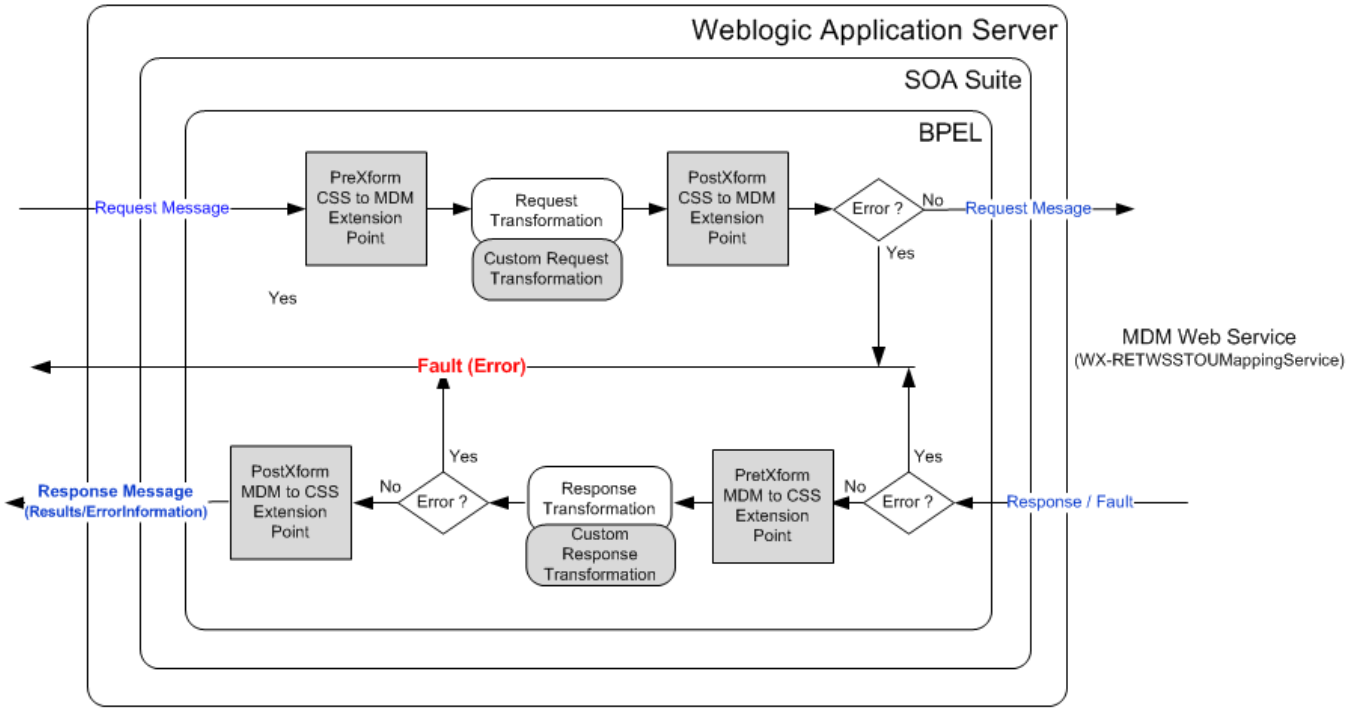
When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Get Usage Details
  - The BPEL process will invoke the MDM web service to get the account's usage details.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
  - If no record were found for the account or the account does not have any active service, MDM will return an empty response. Otherwise, it will return the usage details back to the integration.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM usage details back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back the error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS.
- Customization
  - If the extension point flag (***Extension.PreXformCSStoMDM2)*** is enabled, it will invoke the PreXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PostXformCSStoMDM2)*** is enabled, it will invoke the PostXform CSS to MDM Custom Extension Service.

- If the extension point flag (***Extension.PreXformMDM2toCSS)*** is enabled, it will invoke the PreXform MDM to CSS Custom Extension Service.
- If the extension point flag (***Extension.PostXformMDM2toCSS)*** is enabled, it will invoke the PostXform MDM to CSS Custom Extension Service.
- The extension point flags are defaulted from the Configuration properties file.
- Custom extension xsl templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDM2DirectUsageDetailReqEBF	Self Service Direct Usage Detail Request to MDM BPEL Process Synchronous BPEL process to transform incoming CSS request message to MDM format and retrieve the usage details for an account from MDM. Transform the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-RETWSSTOUMappingService	This inbound service retrieves usage details for a self-service user's account for some period (e.g., year, month or day). The system will attempt to retrieve

	usage information from MDM for each of the account's service agreements. This service may also return temperature information.

Get Consumption Summary (Meter Data Usage) Integration Flow

Business Details

This process is used to get the scalar meter consumption summary for an account from MDM and send back the information to CSS

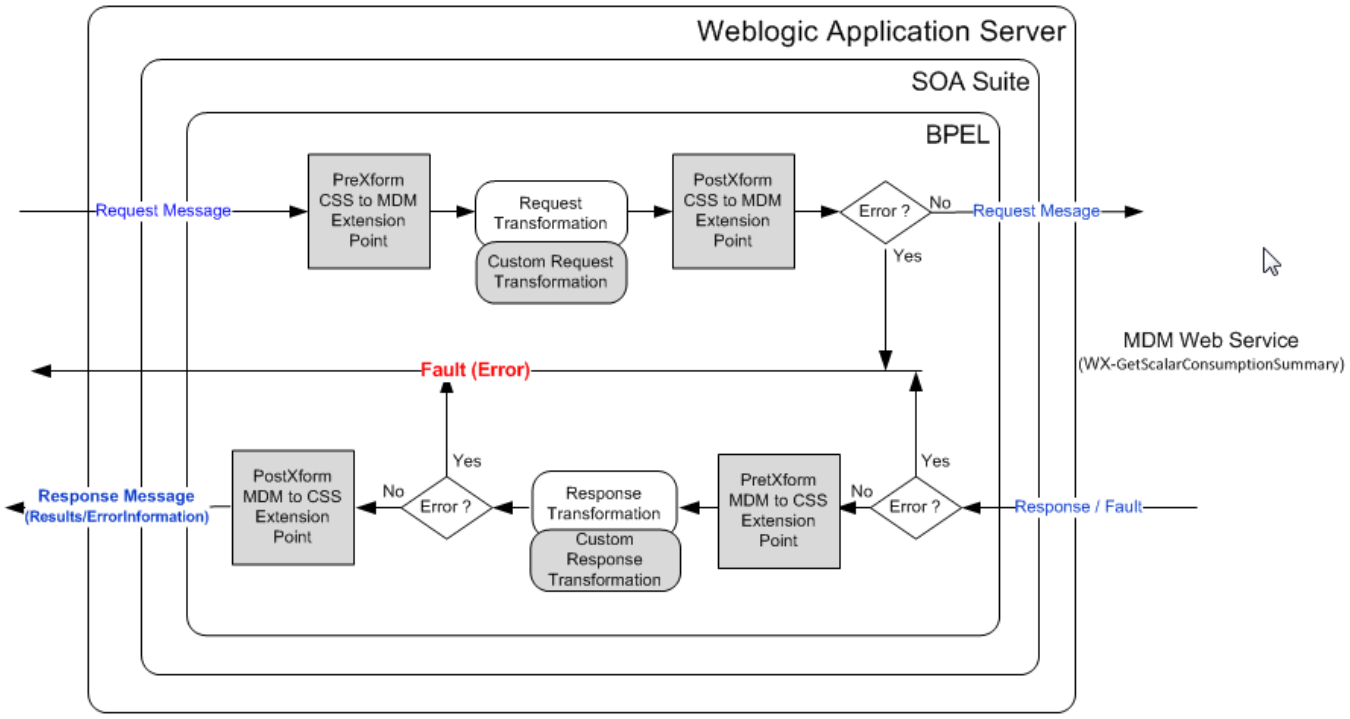
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Get Consumption Summary
  - The BPEL process will invoke the MDM web service, WX-GetScalarConsumptionSummary, to get the account's meter consumption summary.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM usage details back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back the error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS with a specific generic message code. The message code is obtained from the configuration properties file.
- Customization
  - If the extension point flag (***Extension.PreXformCSStoMDM2)*** is enabled, it will invoke the PreXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PostXformCSStoMDM2)*** is enabled, it will invoke the PostXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PreXformMDM2toCSS)*** is enabled, it will invoke the PreXform MDM to CSS Custom Extension Service.
  - If the extension point flag (***Extension.PostXformMDM2toCSS)*** is enabled, it will invoke the PostXform MDM to CSS Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDM2GetConsumptionSummaryEBF	Self Service Consumption Summary Request to MDM BPEL Process Synchronous BPEL process to transform incoming CSS request message to MDM format and retrieve the meter consumption summary for an account from MDM. Transform the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-GetScalarConsumptionSummary	This inbound service retrieves consumption information to display in the self-service application. It retrieves consumption for service agreements that do not require MDM bill determinants.

Add Scalar Meter Read Integration Flow

Business Details

This process is used to retrieve and add scalar meter reads from CSS to MDM.

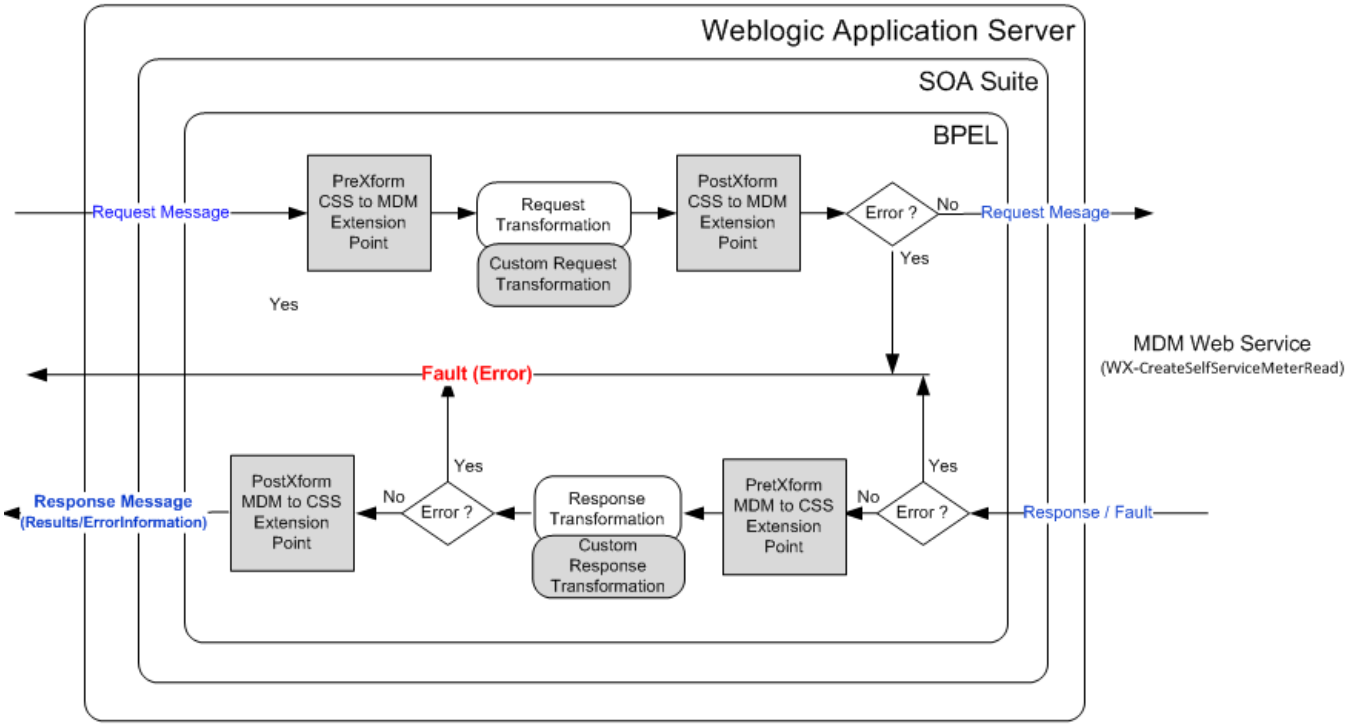
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Add Scalar Meter Read has 2 modes: ADD and READ
  - For ADD, BPEL process will invoke the MDM web service, WX-CreateSelfServiceMeterRead, to create IMD.
  - For READ, BPEL process will invoke the MDM web service, WX-CreateSelfServiceMeterRead, to return scalar register collection with last reading.
  - CSS pass the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM scalar meter read back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back the error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS with a specific generic message code. The message code is obtained from the configuration properties file.
- Customization
  - If the extension point flag (***Extension.PreXformCSStoMDM2)*** is enabled, it will invoke the PreXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PostXformCSStoMDM2)*** is enabled, it will invoke the PostXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PreXformMDM2toCSS)*** is enabled, it will invoke the PreXform MDM to CSS Custom Extension Service.
  - If the extension point flag (***Extension.PostXformMDM2toCSS)*** is enabled, it will invoke the PostXform MDM to CSS Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDM2AddScalarMeterReadEBF	Self Service Consumption Summary Request to MDM BPEL Process Synchronous BPEL process to transform incoming CSS request message to MDM format and retrieve/add the meter read from/in MDM. Transform the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-CreateSelfServiceMeterRead	This inbound service is used by the self-service application. It is responsible for retrieving and adding manual or scalar meter reads. When adding a new meter read, the service merely creates an instance of the Meter Read Creation business object defined on the Self-Service Integration master configuration.

Usage Download Integration Flow

Business Details

This process is used to retrieve specified number of day's interval usage data in CSS in either CSV or XML format. This BPEL process is used to get the usage download data from MDM for both residential and commercial accounts.

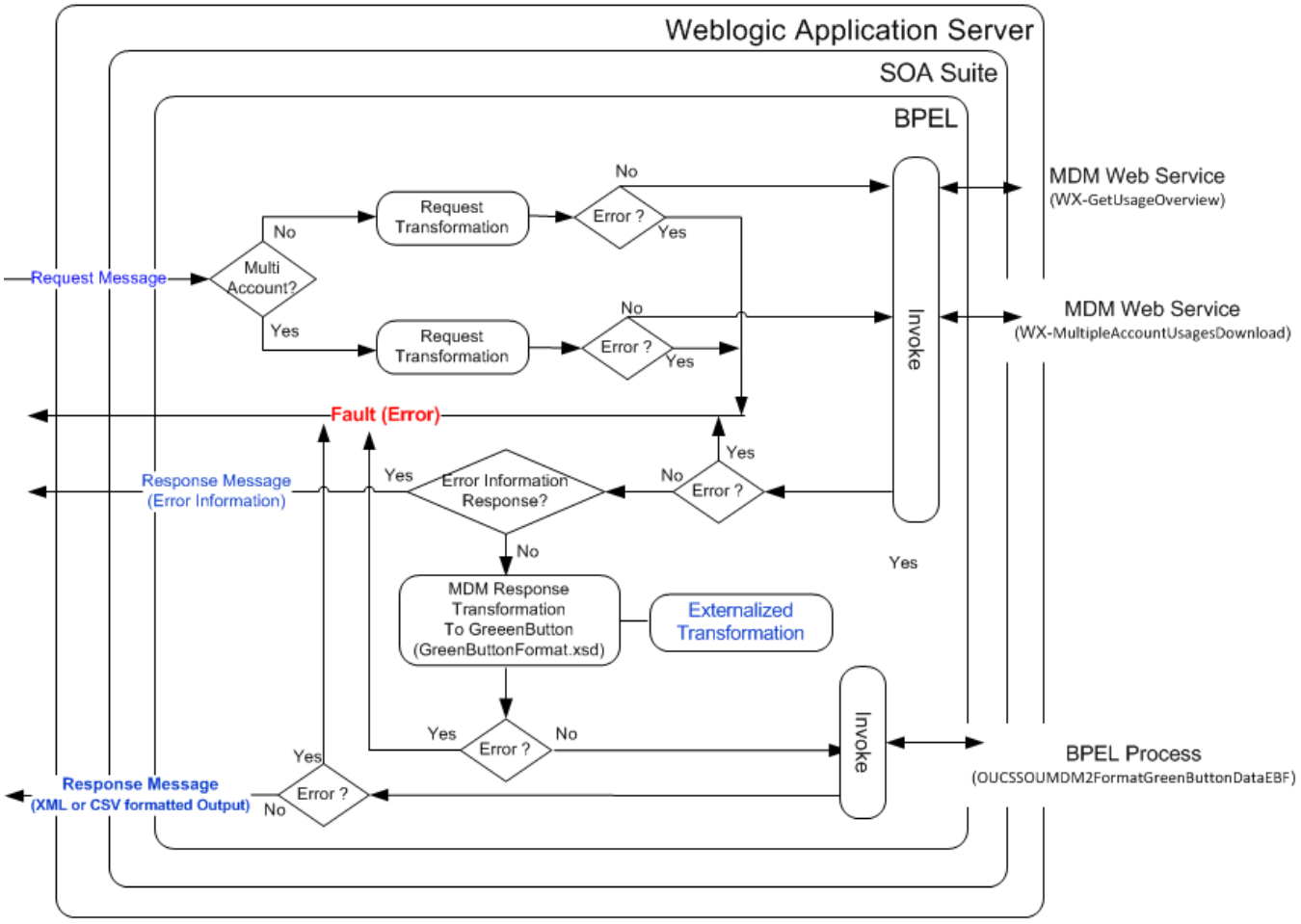
Technical Details

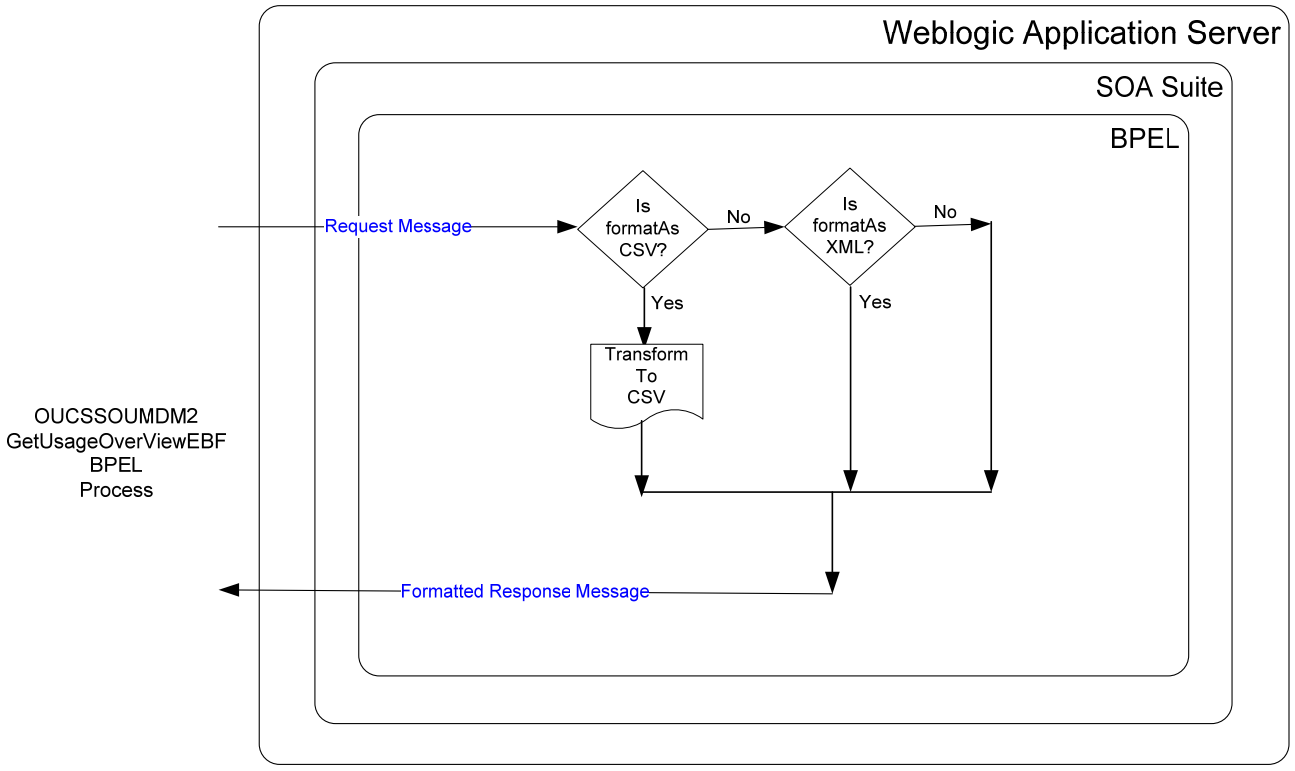
When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS request message to the equivalent MDM request message format.
- MDM service WX-GetUsageOverview
  - The Integration BPEL process will call the MDM service, WX-GetUsageOverview, to retrieve interval usage information for specified number of days. This is use to get usage data for residential accounts.
- MDM service WX-MultipleAccountUsagesDownload
  - The Integration BPEL process will call the MDM service, WX-MultipleAccountUsagesDownload, to retrieve interval usage information for specified number of days for a list of accounts provided. This is use to get usage data for commercial accounts.
- Response Message Transformation
  - MDM Response can contain the interval usage information or error information.
  - If MDM returns error information in the response, the BPEL process will send back this error information to CSS and it will not invoke the BPEL process OUCSSOUMDM2FormatGreenButtonDataEBF.
  - If MDM returns interval usage information in the response, the BPEL process will transform the MDM response to Usage Download format. This transformation is externalized. The transformation files reside in MDS.
- Invoke OUCSSOUMDM2FormatGreenButtonDataEBF
  - Usage Download formatted data is passed to this process which will generate a CSV or XML format output. This process can be modified to support more format options. The endpoint URL for this process is defined in the configuration properties file. The property in which the URL is defined is CSS.CreateGreenButtonFormat.Endpoint.URL.
- Response from FormatGreenButtonData process
  - Response is already in CSV or XML format and this will be passed back to CSS.
- Error Handling
  - When a business or technical error is encountered in CCB or MDM or in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.
- Customization
  - Usage Download format transformation is externalized.

Note: Refer to Usage Download - Externalized Transformations below for more information about customization.

Technical Flow





OUCSSOUMDM2FormatGreenButtonDataEBF

Integration Services

Name	Description
OUCSSOUMDM2GetUsageOverviewEBF	This BPEL process will receive the CSS request messages and invoke CCB or MDM Web Service. CCB will call MDM to retrieve interval usage data. The data is passed back to CSS in either CSV or XML format.
OUCSSOUMDM2FormatGreenButtonDataEBF	This BPEL process receives the Usage Download formatted data from GetUsageOverviewEBF and converts into either CSV or XML.

Web Services

Application	IWS Service Name	Description
MDM	WX-GetUsageOverview	This inbound service retrieves an x-day usage overview for a self-service user's account. The number of days is provided as input to this service. MDM will return Usage Overview information for each of the account's service agreements that require bill determinants
MDM	WX-MultipleAccountUsagesDownload	This inbound service invokes the Get Usage Overview service, to retrieve the input list of account's usages by usage subscription.

Direct Usage Overview Integration Flow

Business Details

This process is used to get the Usage Overview for an account from MDM and send back the information to CSS

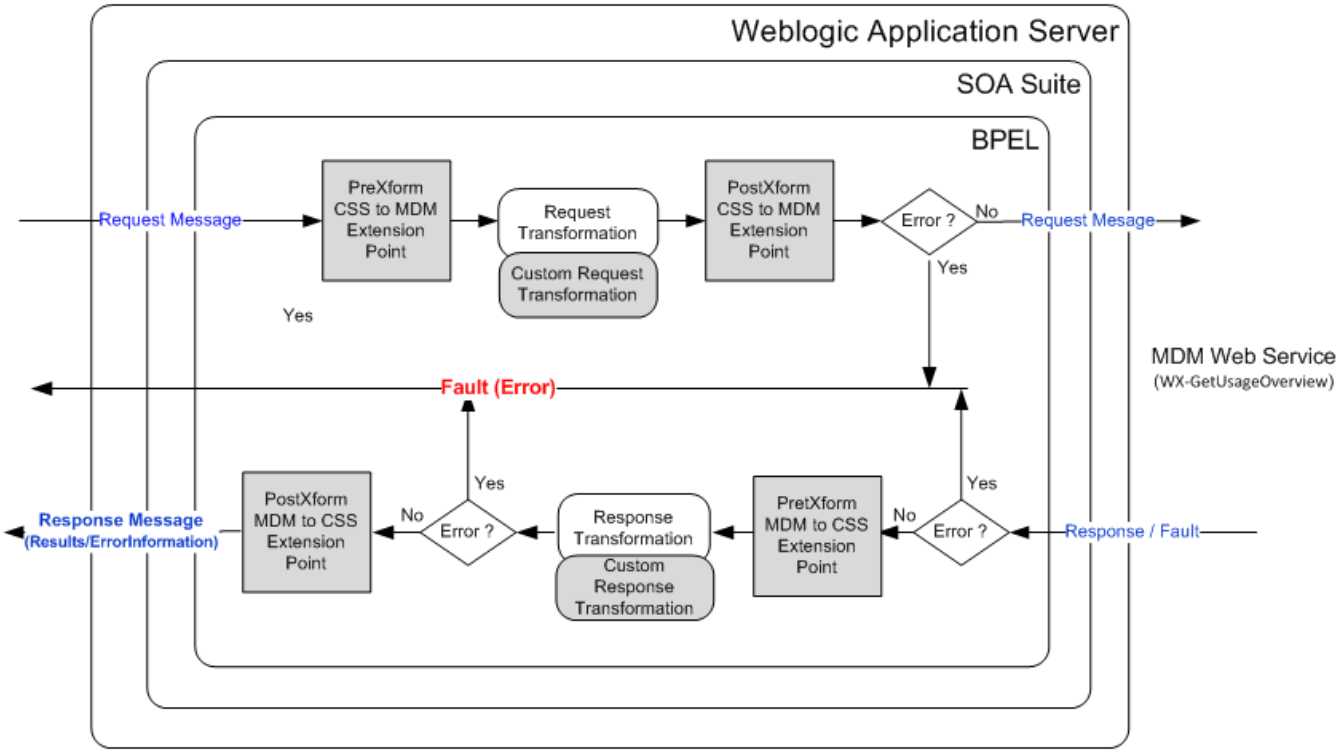
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Get Usage Overview
  - The BPEL process will invoke the MDM web service to get the account's Usage Overview information for the last 7 days.
  - CSS passes the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
  - If no record were found for the account, MDM will return an empty response. Otherwise, it will return the Usage Overview information back to the integration.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM Usage Overview information back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back this error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS with a specific generic message code. The message code is obtained from the configuration properties file.
- Customization
  - If the extension point flag (***Extension.PreXformCSStoMDM2)*** is enabled, it will invoke the PreXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PostXformCSStoMDM2)*** is enabled, it will invoke the PostXform CSS to MDM Custom Extension Service.
  - If the extension point flag (***Extension.PreXformMDM2toCSS)*** is enabled, it will invoke the PreXform MDM to CSS Custom Extension Service.
  - If the extension point flag (***Extension.PostXformMDM2toCSS)*** is enabled, it will invoke the PostXform MDM to CSS Custom Extension Service.
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom XSL templates are also provided for additional mapping.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDM2DirectUsageOverviewEBF	Self Service Direct Usage Overview to MDM BPEL Process Synchronous BPEL process to transform incoming CSS request message to MDM format and retrieve the Usage Overview information for an account from MDM. Transform the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-GetUsageOverview	This inbound service retrieves Usage Overview information for a self-service user's account for the last 7 days. MDM will return Usage information for each of the account's service agreements.

Multiple Account Usage Aggregation Integration Flow

Business Details

This process is used to get the aggregated usage information from MDM for the given list of accounts passed from CSS to be able to display aggregated usage information for business customers in the CSS Portal.

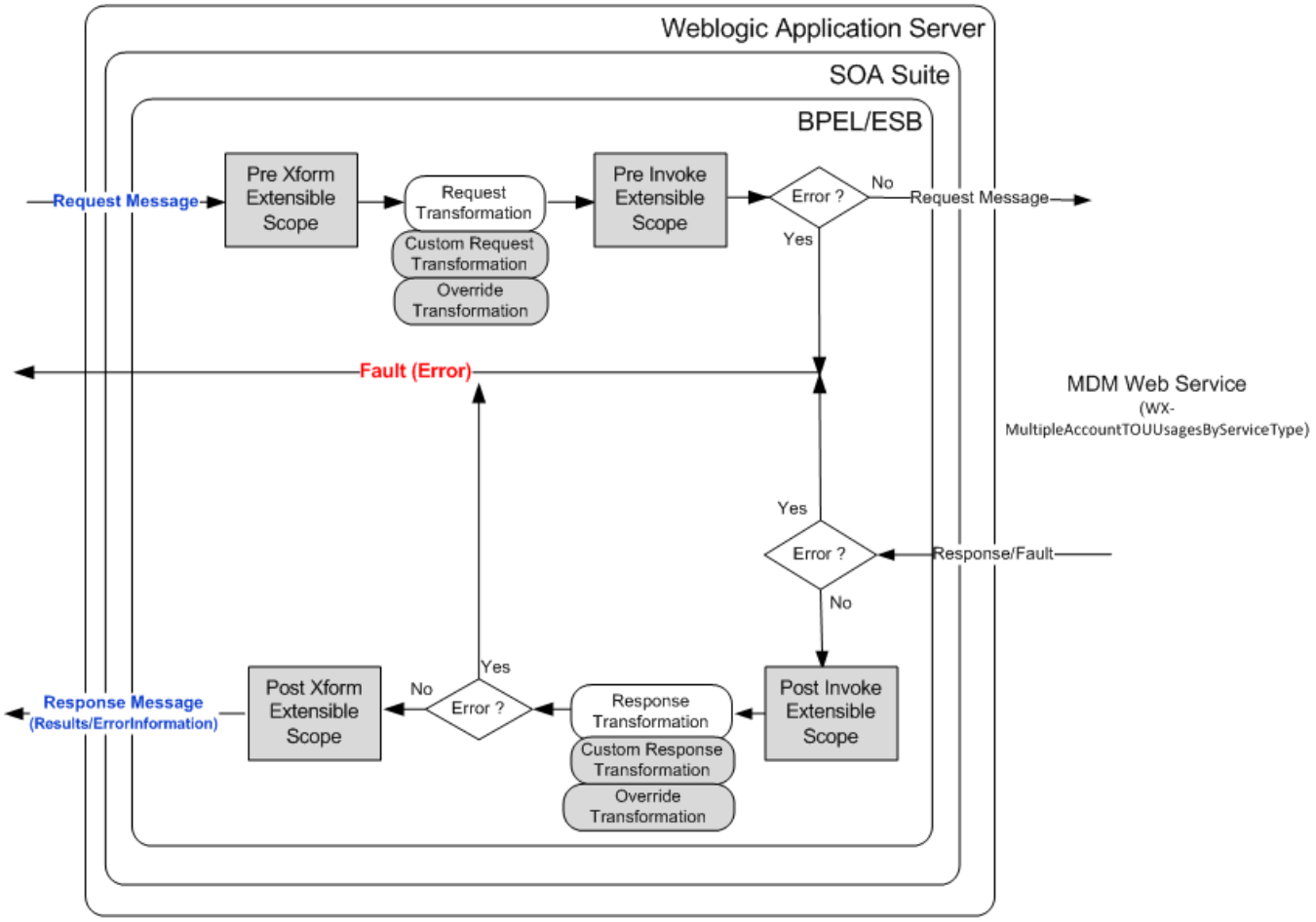
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Get Multiple Account Usage Aggregation
  - The BPEL process will invoke the MDM web service to get the multiple accounts usage aggregation information for the given list of accounts passed from CSS.
  - CSS passes the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
  - If no record were found for the account, MDM will return an empty response. Otherwise, it will return the Multiple Accounts' Usage Aggregation information back to the integration.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM Usage Aggregation information back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back this error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS with a specific generic message code. The message code is obtained from the configuration properties file.
- Customization
  - The following extensible scopes are provided so implementers can invoke any external web service:
  - Pre Transformation Extensible Scope
  - Pre Invoke Extensible Scope
  - Post Invoke Extensible Scope
  - Post Transformation Extensible Scope
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom XSL templates are also provided for additional mapping.
  - Custom Override XSL templates are also provided to override existing mappings.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDMMultiAccountUsageAggregationEBF	Self Service Multiple Account Usage Aggregation BPEL Process Synchronous BPEL process which transforms incoming CSS request message to MDM format and invoke the MDM inbound service to get the aggregated usage data for multiple accounts. Transforms the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-MultipleAccountTOUUsagesByServiceType	This inbound service invokes the Get Usage Details service to retrieve the input list of account's aggregated usages. Each account's usage is summarized by service type, UOM and SQI.

Multiple Account Usage Comparison Integration Flow

Business Details

This process is used to give commercial customers a method by which they can compare usage for their properties in a single view.

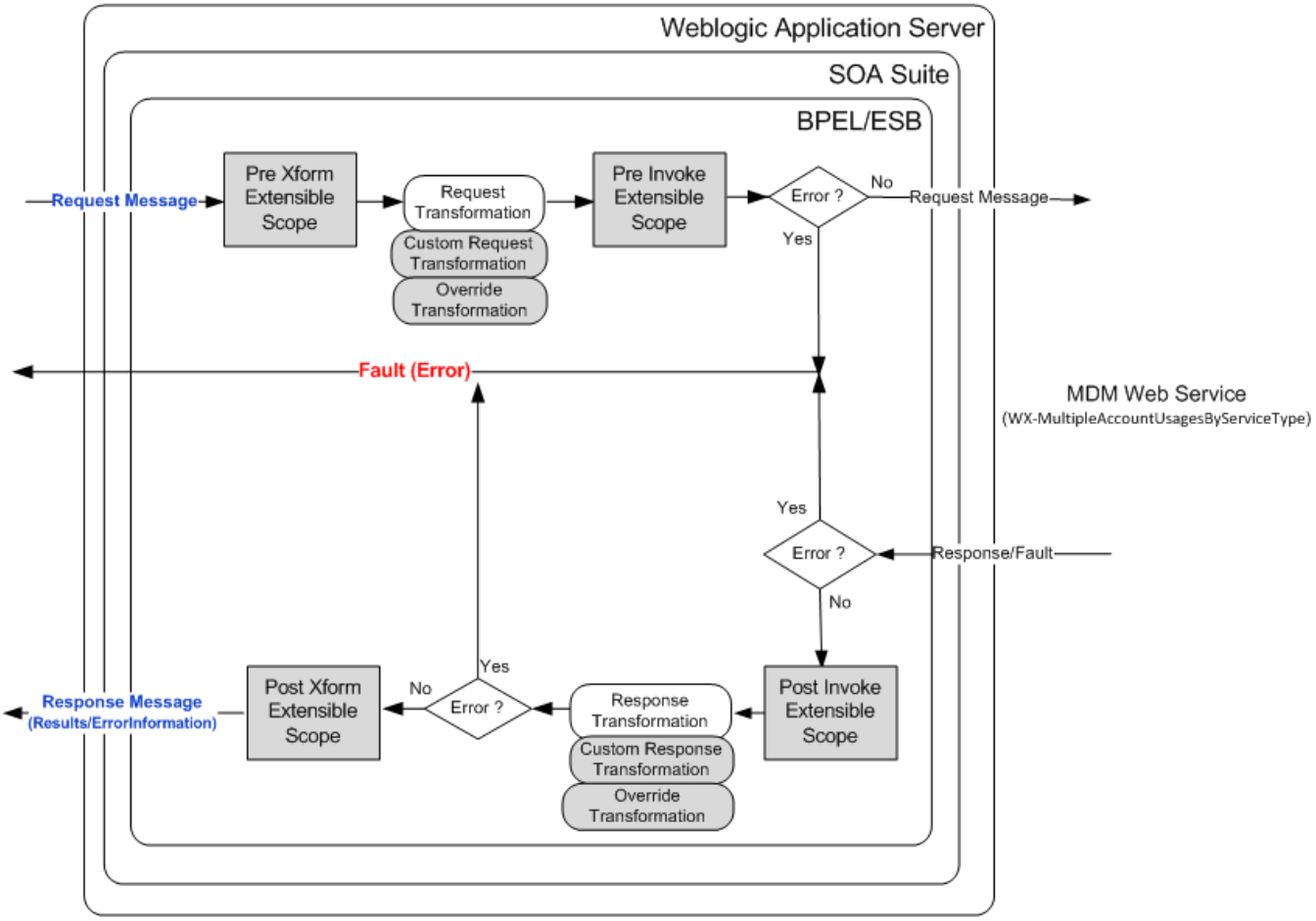
Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

- Request Message Transformation
  - Transforming the CSS input message to the equivalent MDM request message format.
- Get Multiple Account Usage Aggregation
  - The BPEL process will invoke the MDM web service to get the multiple accounts usage comparison information for the given list of accounts passed from CSS.
  - CSS passes the language code of the user as part of the request message's SOAP Header. BPEL passes the language code from CSS to MDM and MDM returns the language-related elements in the appropriate language passed.
  - If no record were found for the account, MDM will return an empty response. Otherwise, it will return the Multiple Accounts Usage ComparisonAggregation information back to the integration.
- Response Message Transformation
  - The BPEL process will transform and pass back the MDM Usage Comparison information back to the CSS response format.
- Error Handling
  - When MDM responds back with an errorInformation, integration will pass back this error information to CSS.
  - When MDM encounter an error and sends a fault back to integration, integration will send a SOAP fault back to CSS.
  - When a technical error or any fault is encountered in the Integration, a SOAP fault will also be returned to CSS with a specific generic message code. The message code is obtained from the configuration properties file.
- Customization
  - The following extensible scopes are provided so implementers can invoke any external web service:
  - Pre Transformation Extensible Scope
  - Pre Invoke Extensible Scope
  - Post Invoke Extensible Scope
  - Post Transformation Extensible Scope
  - The extension point flags are defaulted from the Configuration properties file.
  - Custom XSL templates are also provided for additional mapping.
  - Custom Override XSL templates are also provided to override existing mappings.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSOUMDMMultiAccountUsageComparisonEBF	Self Service Multiple Account Usage Comparison BPEL Process Synchronous BPEL process which transforms incoming CSS request message to MDM format and invoke the MDM inbound service to get the usage comparison data for multiple accounts. Transforms the response coming from MDM back to CSS format.

External Service Call

Web Services

Application	IWS Service Name	Description
MDM	WX-MultipleAccountUsagesByServiceType	This inbound service invokes the Get Usage Details service to retrieve the input list of account's usages. Each account's usage is summarized by account, service type, UOM and SQI.

CSS Account Documents Flows

Upload Document Integration Flow

Business Details

From CSS, as part of Issue Management, the user is able to upload electronic documents stored for the reported issue. Sometimes, a picture inclusion with the reported issue could have benefit (a tree touching power cables, an open cover over an electrical panel, etc.).

CSS will receive the document and pass it over to integration. Integration will be responsible for storing the document on the file system by default. Integration will also allow for customization by Implementation to upload to any third party document management systems.

Integration will pass the response back from integration to CSS indicating whether the upload was successful or not.

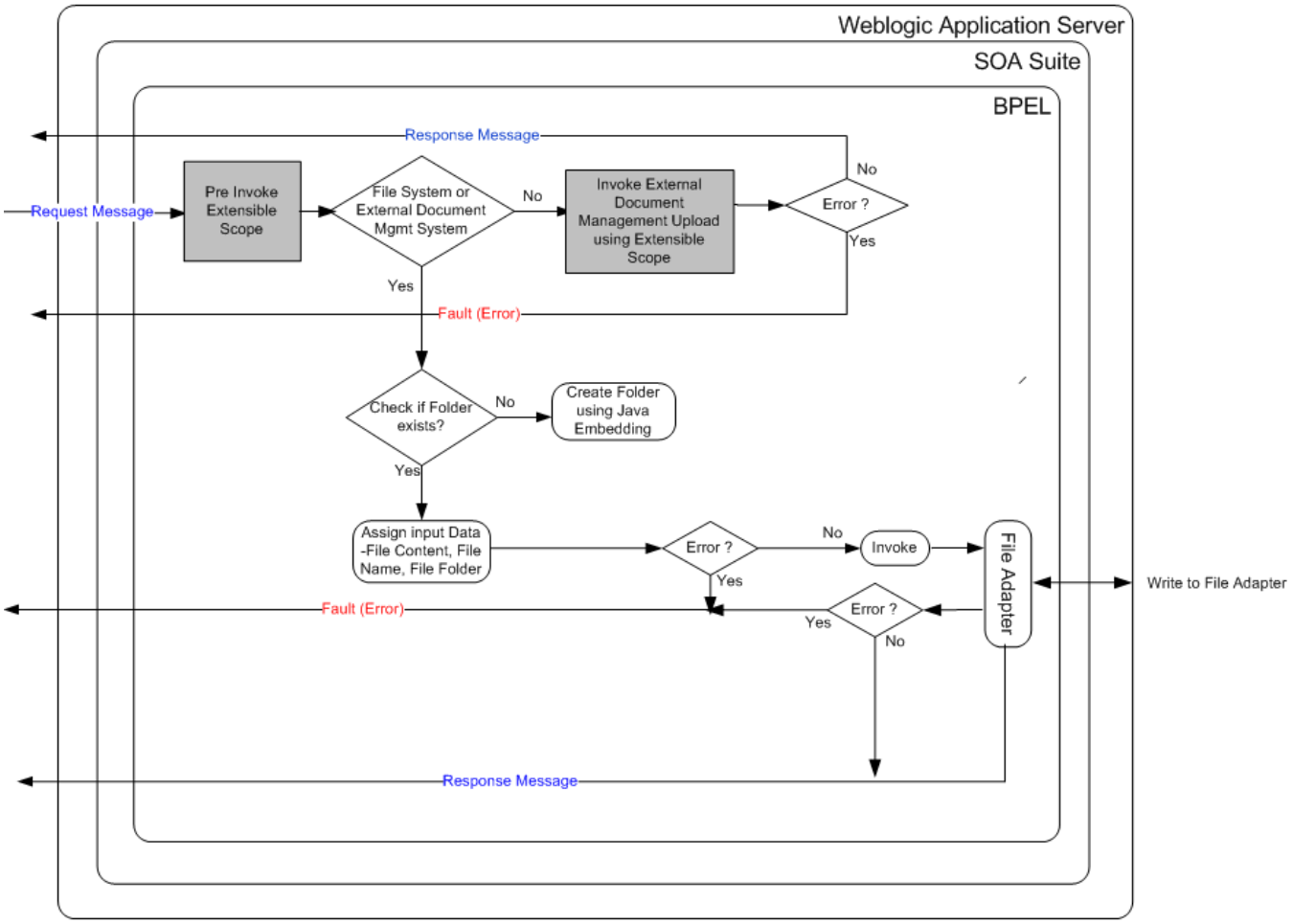
Technical Details

When the integration synchronous BPEL service receives a request from CSS, it will do the following:

- Get Configuration Property - CSS.FileSystemOrExternalDocumentManagementSystem
- Get Configuration Properties CSS.Base.DocumentsDirectory and CSS.Base.FormsDirectory to get the base forms folder
- Check if the property is set to FileSystem or ExternalDocumentManagementSystem
- If FileSystem
- If read was successful, assign response
  - Check if folder name of referenceNumber coming in as input exists on the File System under the base forms folder.
  - If not, create the folder.
  - Invoke File Adapter with the appropriate action to add (ADD/UPDATE action code) or delete the file.
  - If upload was successful send response back to CSS with URL of uploaded location. When returning theURL link to CSS after upload, the property of document URL is appended with the forms/accounts base folder, the reference number of the form or account number of the account document followed by the file name.
  - If upload failed, send errors back to CSS
- If not File System, invoke the extensible scope for integration with External Document Management Service.
- Error Handling
- When an error is encountered in the Integration, a SOAP fault will be returned to CSS
- Customization
- PreInvokeCSSRequest extension scope is invoked after the request is received.
- The ExternalDocMgmtSystem extension scope is invoked if external document management system is being used.
- PostInvokeCSSRequest extension scope is invoked before the response is sent back to CSS
- This extension scope will help the implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSUploadDocumentEBF	Self Service Upload Document BPEL Process. This BPEL process accepts the CSS request message to upload.

External Service Call

File Adapter Service

Name	Description
OUCSSUploadDocumentAdapter	CSS Upload Document Adapter Service. This file adapter service writes a document to the file system.

Read Document Integration Flow

Business Details

This flow allows the CSS user to read any electronic documents previously stored and returns the read document for CSS to render.

When retrieving documents needed for CSS, the flow reads the document from the file system by default. It does allow for customization by Implementation to read from any third party document management systems.

Response is passed back from integration to CSS indicating whether the read was successful or not and the actual content of the document read.

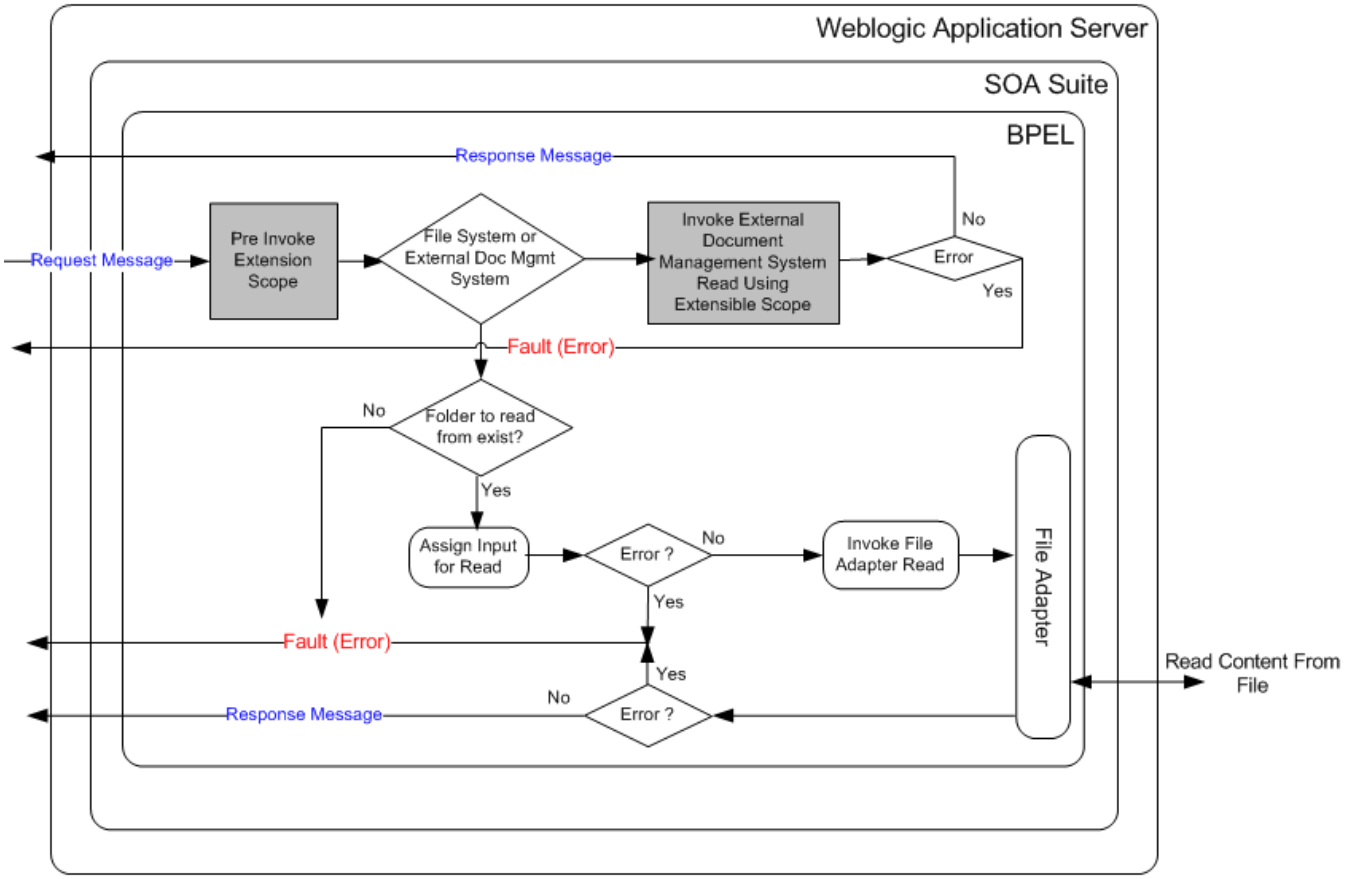
Technical Details

When the integration synchronous BPEL service receives a request from CSS, it will do the following:

- Get Configuration Property - CSS.FileSystemOrExternalDocumentManagementSystem
- Get Configuration Properties CSS.Base.DocumentsDirectory and CSS.Base.FormsDirectory/ CSS.Base.AccountsDirectory to get the base forms folder and base accounts folder
- Check if the property is set to FileSystem or ExternalDocumentManagementSystem
- If FileSystem
- If cssTrigger is FORM check if folder FORM exists on the file system under the base forms folder
- If cssTrigger is ACCOUNT, Check if folder name of ACCT_ID exists on the File System under the base accounts folder.
- If not throw errors back to CSS.
- Invoke File Adapter with the appropriate action to read the file in the folder with name ACCT_ID or referenceNumber depending on the cssTrigger..
- If read was successful, assign response
- Send response back to CSS with the base64 encoded content.
- If read failed, send errors back to CSS.
- If not File System, invoke the extensible scope for integration with External Document Management Service.
- Error Handling
- When an error is encountered in the Integration, a SOAP fault will be returned to CSS
- Customization
- PreInvokeCSSRequest extension scope is invoked after the request is received.
- The ExternalDocMgmtSystem extension scope is invoked if external document management system is being used.
- PostInvokeCSSRequest extension scope is invoked before the response is sent back to CSS
- This extension scope will help the implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUCSSReadDocumentEBF	This BPEL process that reads the account document and returns it to CSS

External Service Call

File Adapter Service

Name	Description
OUCSSReadDocumentAdapter	CSS Read Document Adapter Service. This file adapter service reads a document from the file system.

Implementing the CSS Direct BPEL Flows

Configuring the Integration

This section provides details about the configuration settings required for the integration, and also discusses details related to:

Setting up Oracle Utilities Customer Care and Billing

Setting up Oracle Utilities Meter Data Management

Setting up Network Management System

Data Synchronization between CCB and MDM

Data Synchronization between CIS and NMS

Oracle Utilities Customer Care and Billing Configuration

Refer to the Customer Care and Billing Configuration chapter.

Oracle Utilities Meter Data Management Configuration

Refer to the Meter Data Management Configuration chapter.

Network Management System Configuration

Refer to the Network Management System Configuration chapter.

Data Synchronization

CCB to MDM

Oracle Utilities Meter Data Management serves as the database of record for meter device connections and usage while Oracle Utilities Customer Care and Billing manages customers (persons), accounts (service agreements), and service points. The person, SP, SA, meter, meter configuration, and SP-meter history sync integration points add relevant SP/SA and meter data from Oracle Utilities Customer Care and Billing in Oracle Utilities Meter Data Management.

The data synchronization for rates is not completed by the integration product.

See the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 12.1* for data synchronization processes between two systems.

CIS to NMS

Customer information must be synchronized between the customer information system (e.g., CCB) and Oracle Utilities Network Management System (NMS) to be able to report an outage for a specific customer (account) and to be able to view customer's outage information retrieved from NMS in the Self Service Application.

If CCB is the CIS System used, see the *Implementation Guide for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Network Management System Release 12.1* for customer synchronization processes between the two systems.

CSS Direct Integration Product Configuration

The following sections describe the configuration needed in the integration to meet the requirements for this integration.

Configuration steps include setting the following:

Task	Remarks
Setting Configuration Properties	Update the ConfigurationProperties.xml file.
Setting System Properties	Set the Module Configurations properties that are shared by multiple integration flows and Service Configurations properties that are used by a specific BPEL process.
Domain Value Maps	Set the Domain value maps (DVMs) to map codes and other static values across applications.
Error Handling	Set up error notifications.

Setting Configuration Properties

The ConfigurationProperties.XML file contains properties which can be defaulted in the integration. Also, it contains flags to enable extension points within the integration.

ConfigurationProperties.XML is located in MDS under the directory apps/OUCSS/AIAMetaData/config.

Note. Whenever the ConfigurationProperties.XML file is updated, it must be reloaded to MDS for updates to be reflected in the applications or services that use the updated properties. You can perform the reload by rebooting the SOA server.

Setting System Properties

There are two sets of configuration properties described in this section:

- Module Configurations are the properties that are shared by multiple integration flows within the Oracle Utilities Self-Service Integration Pack.
- Service Configurations are the properties that are used by a specific BPEL process.

Module Configurations

Module Name	Default / Shipped Value	Description
CSS.TechnicalFault.MessageCode	SYSTEM_UNAVAILABLE_EXCP_MSG	This value should be the generic message code setup in CSS for technical errors (e.g., when the edge apps are down). This is the message code that the integration process passes back to CSS when a technical fault is encountered. Used by all the CSS-NMS flows except OUCSSGetAlertsEBF.
CSS.Generic.ExceptionCode	999999999	This is the exception code that the integration process passes back to CSS when a fault is encountered in the integration. Used by the following flows: •OUCSSOUMDM2AddScalarMeterReadEBF •OUCSSOUMDM2DirectUsageDetailReqEBF

		•OUCSSOUMDM2GetConsumptionSummaryEBF
SOA-INFRA.AuditLevel	ON	This property needs to be set to OFF if the Audit Level is set to OFF for the BPEL processes. If the setting is OFF, then error handling does not use the composite and component instance IDs to log the error message.
ErrorHandling.GenericEmailID		This property is used to set the administrator email ID for the error handling process to send out an email in case of a critical failure where even the Error handling process fails.
CSS.Base.DocumentsDirectory	/opt/apache2.2/htdocs/	This property is used to indicate the folder structure in the file system where the documents will be saved
CSS.Base.FormsDirectory	formsBaseDirectory	This property is used to indicate the name of the folder under the CSS.Base.DocumentsDirectory where the form documents would be saved. The CSS.Base.FormsDirectory is appended to CSS.Base.DocumentsDirectory to get the forms base directory.
CSS.Base.AccountsDirectory	accountsBaseDirectory	This property is used to indicate the name of the folder under the CSS.Base.DocumentsDirectory where the account documents would be saved. The CSS.Base. AccountsDirectory is appended to CSS.Base.DocumentsDirectory to get the accounts base directory.
CSS.Base.DocumentsURL	http://host:port/	This property is used to indicate the base URL to access the doc. When returning the URL link to CSS after upload, this property of document URL is appended with the forms/accounts base folder, the reference number of the form or the account number of the account followed by the file name.

Service Configurations

Property Name	Default / Shipped Value	Description
Service Name: OUCSSOUNMSOutageSummaryEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
Extension.PreXformOUCSStoOUNMS	false	Not in use.
Extension.PostXformOUCSStoOUNMS	false	If set to true, the post transformation extension service for the request message is invoked.
CSS.OutageSummary.BusinessFault.MessageCode	OUTAGE_SUMMARY_EXCP_MSG	The value should be the generic message code setup in CSS for business errors (when the edge apps or integration encounters a business error). This is the message code that the integration process passes back to CSS when a business fault is encountered.
Service Name : OUCSSGetOutageScreenInfoEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CCB.SkipBaseServiceFlag	false	If CCB is not installed or implementation does not want the integration to call the CCB Base service, set the value to *true*
NMS.SkipBaseServiceFlag	false	If NMS is not installed or implementation does not want the integration to call the

		NMS Stored Procedure, set the value to *true*.
Extension.PreXformOUCSStoOUCCB	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformOUCCBtoOUCSS	false	If set to true, the post transformation extension service for the response message is invoked.
OUCCB.ServiceInfo.Endpoint.URL		This value is the CCB Service Details Endpoint URL. Shipped with this value: @EdgeApplications.OUCCB.ManagedServer.protocol://@EdgeApplications.OUCCB.ManagedServer.hostname:@EdgeApplications.OUCCB.ManagedServer.portnumber/@EdgeApplications.OUCCB.ManagedServer.context/Webservices/WXServiceDetails. During install, the CCB edge application information will be tokenized to point to the correct CCB server being used.
CSS.OutageCodes.BusinessFault.MessageCode	OUTAGE_CODES_EXCP_MSG	The value should be the generic message code setup in CSS for business errors (when the edge apps or integration encounters a business error). This is the message code that the integration process passes back to CSS when a business fault is encountered.
Service Name : OUCSSOUNMSTroubleCallInterfaceEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
Extension.PreXformCSStoNMS	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCSStoNMS	false	If set to true, the post transformation extension service for the request message is invoked.
NMS.CSSCallSourceId	5	This value is the unique Call Source Id for CSS. This will ensure that the external ID created in the NMS trouble calls table will have a unique value because NMS will prefix the call source ID to the external ID generated. By looking at the external id, the first number, which is the Call Source ID, will indicate that the trouble call was created in CSS.
CSS.ReportOutage.BusinessFault.MessageCode	OUTAGE_EXCP_MSG	The value should be the generic message code setup in CSS for business errors (when the edge apps or integration encounters a business error).

		This is the message code that the integration process passes back to CSS when a business fault is encountered.
BusinessError.NotificationFlag	false	If set to true, business error notification is sent via Email.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUCSSGetAlertsEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CCB.SkipBaseServiceFlag	False	If CCB is not installed or implementation does not want the integration to call the CCB Base service, set the value to ***true***
NMS.SkipPlanOutBaseServiceFlag	False	If NMS is not installed or implementation does not want the integration to call the NMS Switching History Stored Procedure, set the value to ***true***.
NMS.SkipCurrOutBaseServiceFlag	False	If NMS is not installed or implementation does not want the integration to call the NMS Job History Stored Procedure, set the value to ***true***.
Extension.PreXformCSS	False	If set to true, the pre transformation extension service for the request message is invoked.
Extension. PostXformCSS	False	If set to true, the post transformation extension service for the response message is invoked.
NMS.CurrentOutageAlertHeader	OUTAGE	Text to define the Current Outage Alert Header.
NMS.CurrentOutageAlertText	Your service is currently reported as out. See Outages for more information.	Text to define the Current Outage Alert Text.
NMS.CurrentOutageAlertIconCode	CURRENT_OUTAGE_IMG	The value should be the Current Outage Alert Icon Code setup in CSS.
NMS.CurrentOutageAlertLinkCode	CURRENT_OUTAGE_URL	The value should be the Current Outage Link Code setup in CSS.
NMS.CurrentOutageAlertType	OATYPE	Define the Alert Type Code for Current Outage.
NMS.PendingOutageAlertHeader	PLANNED OUTAGE	Text to define the Pending Alert Header.
NMS.PendingOutageAlertText	Your service will have a planned outage on	Text to define the Pending Outage Alert Text.
NMS.PendingOutageAlertIconCode	PLANNED_OUTAGE_IMG	The value should be the Pending Outage Alert Icon Code setup in CSS.
NMS.PendingOutageAlertLinkCode	PLANNED_OUTAGE_URL	The value should be the Pending Outage Link Code setup in CSS
NMS.PendingOutageAlertType	PLATYPE	Define the Alert Type Code for Pending Outage.
CCB.GetAlerts.Endpoint.URL		This value is the CCB Get Alerts Endpoint URL. Shipped with this value: @EdgeApplications.OUCCB.ManagedServer.protocol://@EdgeApplications.OUCCB.ManagedServer.hostname:@EdgeApplications.OUCCB.ManagedServer.portnumber/@EdgeApplications.OUCCB.Manage

		dServer.context/Webservices/WXGetCCBAlerts During install, the CCB edge application information will be tokenized to point to the correct CCB server being used.
Service Name : OUCSSOUNMSOutageDetailEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CCB.SkipBaseServiceFlag	False	If CCB is not installed or implementation does not want the integration to call the CCB Base service, set the value to ***true***
NMS.SkipPlanOutBaseServiceFlag	False	If NMS is not installed or implementation does not want the integration to call the NMS Switching History Stored Procedure, set the value to ***true***.
NMS.SkipCurrOutBaseServiceFlag	False	If NMS is not installed or implementation does not want the integration to call the NMS Job History Stored Procedure, set the value to ***true***.
Extension. PreXformOUCSSReq	False	If set to true, the pre transformation extension service for the request message is invoked.
Extension. PostXformOUCSSResponse	False	If set to true, the post transformation extension service for the request message is invoked.
CSS.OutageDetail.BusinessFault.MessageCode	OUTAGE_DETAIL_EXCP_MSG	The value should be the generic message code setup in CSS for business errors (when the edge apps or integration encounters a business error). This is the message code that the integration process passes back to CSS when a business fault is encountered.
OUCCB.ServiceInfo.Endpoint.URL		This value is the CCB Service Details Endpoint URL. Shipped with this value: @EdgeApplications.OUCCB.ManagedServer.protocol://@EdgeApplications.OUCCB.ManagedServer.hostname:@EdgeApplications.OUCCB.ManagedServer.portnumber/@EdgeApplications.OUCCB.ManagedServer.context/Webservices/WXServiceDetails During install, the CCB edge application information will be tokenized to point to the correct CCB server being used.
Service Name : OUCSSOUNMSOutagePublicDetailEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CSS.OutagePublicDetail.BusinessFault.MessageCode	OUTAGE_PUBLIC_DETAIL_EXCP_MSG	The value should be the generic message code setup in CSS for business errors (when the edge apps or integration encounters a business error). This is the message code that the

		integration process passes back to CSS when a business fault is encountered.
Service Name : WXUsageDetail		
Default.SystemID	OU_CSS_01	Initiating system ID.
Extension.PreXformCSStoMDM2	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCSStoMDM2	false	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCSS	false	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCSS	false	If set to true, the post transformation extension service for the response message is invoked.
MDM2.UsageDetail.Endpoint.URL		This value is the MDM Retrieve WSS TOU Mapping Endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-RETWSSTOUMappingService During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2GetConsumptionSummary		
Default.SystemID	OU_CSS_01	Initiating system ID.
Extension.PreXformCSStoMDM2	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCSStoMDM2	false	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCSS	false	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCSS	false	If set to true, the post transformation extension service for the response message is invoked.
MDM2.GetConsumptionSummary.Endpoint.URL		This value is the MDM WXGetScalarConsumptionSummary endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMD

		M.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-GetScalarConsumptionSummary During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2AddScalarMeterReadEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
Extension.PreXformCSStoMDM2	False	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCSStoMDM2	False	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCSS	False	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCSS	false	If set to true, the post transformation extension service for the response message is invoked.
MDM2.AddScalarMeterRead.Endpoint.URL		This value is the MDM WX-CreateSelfServiceMeterRead endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-CreateSelfServiceMeterRead During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2GetUsageOverviewEBF *Note: This is a Usage Download integration flow*		
Default.SystemID	OU_CSS_01	System ID
BaseXFormFlag	true	If set to true, uses default provided base transformation If set to false, custom externalized transformation is used.
CSS.GetUsageOverview.BusinessFault.MessageCode	GREENBUTTON_ERROR_MSG	Default CSS error code
CSS.CreateGreenButtonFormat.Endpoint.URL		This value is the endpoint URL where the OUCSSOUMDM2FormatGreenButtonDataEBF is deployed. Shipped with this value: @SOAHostname.protocol://@SOAHostna

		me:@SOAPortnumber/soa-infra/services/OUCSS/OUCSSOUMDM2FormatGreenButtonDataEBF/OUCSSOUMDM2FormatGreenButtonData_client_ep During install, the OUCSSOUMDM2FormatGreenButtonDataEBF information will be tokenized to point to the correct URL.
OUMDM.DirectUsageOverview.Endpoint.URL		This value is the MDM WX-GetUsageOverview service endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-GetUsageOverview During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
OUMDM.MultipleAccountUsagesDownload.Endpoint.URL		This value is the MDM WX-MultipleAccountUsagesDownloadservice endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-MultipleAccountUsagesDownload During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2FormatGreenButtonDataEBF		
Default.SystemID	OU_CSS_01	System ID
Service Name : OUCSSOUMDM2DirectUsageOverviewEBF		
Default.SystemID	OU_CSS_01	System ID
Extension.PreXformCSStoMDM2	false	If set to true, the pre transformation extension service for the request message is invoked.
Extension.PostXformCSStoMDM2	false	If set to true, the post transformation extension service for the request message is invoked.
Extension.PreXformMDM2toCSS	false	If set to true, the pre transformation extension service for the response message is invoked.
Extension.PostXformMDM2toCSS	false	If set to true, the post transformation extension service for the response

		message is invoked.
MDM2.UsageOverview.Endpoint.URL		This value is the MDM WX-GetUsageOverview service endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-GetUsageOverview During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2MultipleAccountUsageAggregationEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
MDM2.MultipleAccountTOUUsagesByServiceType.Endpoint.URL		This value is the MDM WXMultipleAccountTOUUsagesByServiceType Endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-MultipleAccountTOUUsagesByServiceType During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSOUMDM2MultipleAccountUsageComparisonEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
MDM2.UsageDetail.Endpoint.URL		This value is the MDM WXMultipleAccountUsagesByServiceType Endpoint URL. Shipped with this value: @EdgeApplications.OUMDM.ManagedServer.protocol://@EdgeApplications.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/Webservices/WX-MultipleAccountUsagesByServiceType During install, the MDM edge application information will be tokenized to point to the correct MDM server being used.
Service Name : OUCSSUploadDocumentEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CSS.FileSystemOrExternalDocumentManagementS	FILE	This value determines if the documents are stored in a file system or in an external

ystem		document management system. Default Value : FILE
Service Name : OUCSSReadDocumentEBF		
Default.SystemID	OU_CSS_01	Initiating system ID.
CSS.FileSystemOrExternalDocumentManagementSystem	FILE	This value determines if the documents are stored in a file system or in an external document management system. Default Value : FILE

Note: the configuration values are case-sensitive.

Domain Value Maps

No DVMs are used for this integration.

Error Handling

Refer to the Integration Points section for the error handling for each integration flow.

Customization and Extension Methodology

The integration process allows extensibility of transaction messages using the following methods:

- Pre Transformation Extension Point
- Post Transformation Extension Point
- Pre Transformation Extension Scope
- Pre Invoke Extension Scope
- Post Invoke Extension Scope
- Post Transformation Extension Scope
- Custom Transformations

Override Transformations Pre Transformation Extension Point

The pretransformation extension point is invoked before the main transformation is executed. This transformation aids in transforming the source XML coming as an input to the integration process.

The integration layer defines an external call from the pretransformation extension point. This extension point accepts source XML as input and gives the source XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL This can assist the implementation in invoking any external Web service and transform the input XML.

Post Transformation Extension Point

The post transformation extension point is invoked after the main transformation is executed. This transformation aids in transforming the target XML going as an input to the target queue.

The integration layer defines an external call from the post transformation extension point. This extension point accepts the target XML as input and gives the target XML as output. The implementation can choose to plug in a concrete WSDL instead of the abstract WSDL. This can assist the implementation in invoking any external Web service and transform the output XML.

Pre Transformation Extension Scope

The pre transformation extension scope is invoked before the main transformation is executed. This will help the implementers to invoke any external web service and extend the input xml.

Pre Invoke Extension Scope

The pre invoke extension scope is invoked before invoking the target application service. This will help the implementers to invoke any external web service and enhance the input going to the target service.

Post Invoke Extension scope

The post invoke extension scope is invoked after the target application service invocation. This will help the implementers to invoke any external web service and enhance the output from the target service.

Post Transformation Extension scope

The post transformation extension scope is invoked after the main transformation is executed. This will help the implementers to invoke any external web service and enhance the output xml.

Custom Transformations

The custom transformations are used to add data to custom elements in the incoming and outgoing messages. The incoming and outgoing messages have custom elements defined in the message. These custom elements refer to a custom XML schema. The main transformation invokes custom transformation.

Empty custom transformation and custom schemas are shipped with the product. The implementation team can add additional fields in the custom schema and map them using the custom transformations.

Using custom transformations allows the implementation to define and pass additional data from the source system to the target system.

Override Transformations

The overrride transformations are used to override or change the value(s) to existing elements that are already mapped in the incoming and outgoing messages. These custom override transformation are called after the main transformation.

Override custom transformation are shipped with the product. Not all BPEL flows have this.

Implementation team can change existing field values using the override transformation.

Steps to Implement Extension Points

Each process in the integration has a pre- and post-transformation extension point which can be used to invoke Web services and transform the payload.

1. The desired extension point can be triggered from the process by enabling it using the ConfigurationProperties.xml pre- and post-transformation extension flags as described in section Setting Configuration Properties.

   - Each process has its own concrete wsdl which is used to read the endpoint location for the extension service.

   These concrete wsdl files are located in MDS under the following directories:

   ```
   /apps/OUCSS/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCSS
   ```

   Update the concrete wsdl file to define the binding and service details for the extension service to be called and move the concrete wsdl file to MDS. See *Sample* below.

2. To move the updated concrete wsdl to MDS, update the appropriate wsdl in the product install home

   - The directories to put the concrete wsdl in product install home are the following:

   ```
   $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCSS
   ```

   (Example: $PRODUCT_HOME is CSSDirectFlows).

3. Deploy the concrete wsdls to MDS by running the ant deploy command for updating MDS folder.

Note: For more information about the command to use to deploying to MDS, see the *Oracle Utilities Self Service Installation Guide*, "Updating MDS Folder" section.

4. After deploying the files to MDS, restart the SOA server.

After restarting the SOA server, the extension point invokes the Web service in the concrete WSDL.

Sample wsdl File with Binding and Service Details

For example: To enable the extension points for OUCSSGetAlertsEBF, add the binding and service elements to the OUCSSGetAlertsExtensionConcrete.wsdl.

```
    <binding name="OUCSSGetAlertsExtensionV1ExtensionServiceSOAP11Binding"
             type="cssext:OUCSSGetAlertsV1ExtensionService">
        <soap:binding style="document"
                      transport="http://schemas.xmlsoap.org/soap/http"/>
        <operation name="PreXformOUCSSReq">
            <soap:operation style="document"
                            soapAction="http://ouaf.oracle.com/spl/webservices
/WXGetCCBAlerts/WXGetCCBAlertsExtension/V1/PreXformOUCSSReq"/>
            <input>
                <soap:body use="literal" parts="PreXformCSSRequest"/>
            </input>
            <output>
                <soap:body use="literal" parts="PreXformCSSRequest"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
        </operation>
        <operation name="PostXformOUCSSResp">
            <soap:operation style="document"
                            soapAction="http://ouaf.oracle.com/spl/webservices
/WXGetCCBAlerts/WXGetCCBAlertsExtension/V1/PostXformOUCSSResp"/>
```

```
            <input>
                <soap:body use="literal" parts="PostXfromCSSResponse"/>
            </input>
            <output>
                <soap:body use="literal" parts="PostXfromCSSResponse"/>
            </output>
            <fault name="fault">
                <soap:fault name="fault" use="literal"/>
            </fault>
        </operation>
    </binding>
    <service name="OUCSSGetAlertsExtensionService">
        <port name="OUCSSGetAlertsV1ExtensionService_pt"
              binding="cssext:OUCSSGetAlertsExtensionV1ExtensionServiceSOAP11Binding">
            <soap:address location="http://ouaf.oracle.com:8056/soa-
infra/services/default/ExtService-Alerts/AlertsAbsExtService"/>
        </port>
    </service>
```

Note: The binding and service can be added using Oracle JDeveloper 11g.

Steps to Implement Custom Transformation

To implement custom transformations:

1. Each process in the integration has its own xsd file. The messages have custom elements which can be used to pass additional data from one application to another or vice versa. Refer to the message mappings to see the location of customElements in each message.
2. Each process uses two XSD files, one for the source application message (CSS) and one for the target application message (CCB, NMS, MDM).
3. Each XSD file has a corresponding CustomType XSD file in which the complexType elements for each customElements tag are defined. For this integration, not all the XSD files have a corresponding Custom XSD file. Some XSD files already have pre-defined custom elements in the schema.

   Example:

   CSS schema file (XSD) for Get Outage Screen Info is OUCSSGetOutageScreenInfo.xsd. It does not have a custom type xsd file.

   MDM schema file (XSD) for Usage Detail is: OUMDM2SSRETWSSTOUMappingService.xsd.

   Corresponding custom type schema file (Custom XSD) is: OUMDM2SSRETWSSTOUMappingServiceCustomType.xsd.

   The XSD files are located in product install home under the following directories:

   - $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCSS/V1/schemas
   - $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCCB/V1/schemas
   - $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/schemas
   - $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUNMS/V1/schemas
4. To pass additional elements in the customElements tag, the corresponding complexType needs to be modified in the customType xsd by adding the needed elements to the complexType elements.

For example: In the Usage Detail process, to pass WXUSageDetail > head > customElements > userId element in CSS to WX-RETWSSTOUMappingService > input > customElements > userId element in MDM, the following changes must be implemented:

D In WXUsageDetailCustomType.xsd, add the userId element to the schema. This custom xsd file is located in $PRODUCT_HOME/MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCSS/V1/schemas folder.

```
<xsd:complexType name="headCustomType">
     <xsd:sequence>
          <xsd:element name="userId" type="xsd:string"/>
     </xsd:sequence>
<xsd:complexType>
```

E In OUMDM2SSRETWSSTOUMappingServiceCustomType.xsd, add the userId element in the schema. This xsd file is located in $PRODUCT_HOME /MDS-Artifacts/OUCSS/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/schemas folder.

```
<xsd:complexType name="inputCustomType">
     <xsd:sequence>
          <xsd:element name="userId" type="xsd:string"/>
     </xsd:sequence>
<xsd:complexType>
```

Note: This step is only needed if the XSD file has a corresponding CustomType XSD file.

5 Each process has a main transformation which invokes custom templates. Each main transformation file has a corresponding Custom xsl and the custom templates are defined in the Custom xsl.

Example:

The transformation file (XSL) for Usage Overview request is: Xform_OUCSSUsageDetailReq_to_OUMDM2UsageDetailReq.xsl.

The corresponding custom transformation file (custom XSL) is: Xform_OUCSSUsageDetailReq_to_OUMDM2UsageDetailReq_Custom.xsl.

The custom XSL files are located in product home under the following directories:

`$PRODUCT_HOME/ services/industry/Utilities/EBF/<Process Name>/xsl`

6 These custom templates are invoked at the location where each customElements tag is present. The Custom xsl can be modified to add transformation for the newly added elements in the custom xsd files.

Example:

Transformation in Xform_OUCSSUsageDetailReq_to_OUMDM2UsageDetailReq_Custom.xsl

```
 <xsl:template name="input-customElements">
   <userId>
     <xsl:value-of select="/ns2:WXUSageDetail/ns2:head/ns2:customElements/ns0:userId"/>
  </userId>
</xsl:template>

** ns2 is the namespace ns2=http://oracle.com/WXUsageDetail.xsd
```

7 After updating the XSD and XSL files in the product install home, update MDS using the ant deploy command for updating the MDS folder.

Note: For more information about the command to use to deploying to MDS, see the *Oracle Utilities Self Service Installation Guide*, "Updating MDS Folder" section.

8 After deploying the files to MDS, restart the SOA server.

9 After restarting the SOA server, the changes to the custom xsd and xsl will be reflected in the integration.

Usage Download - Externalized Transformations

Note: This relates to only Usage Download flow. The BPEL process is called, OUCSSOUMDM2GetUsageOverviewEBF.

The BPEL component in the process composite invokes extended transformation. The extended transformation file invokes the base transformation or the implementation specific custom transformation based on a Service module level flag. Using this approach an implementation team can override the complete base transformation with their own transformation.

The transformation xsl files invoked from the BPEL components are stored in MDS.

1. In the PRODUCT installation folder these files are at the location <PRODUCT_HOME>/MDS-Artifacts/OUCSS/AIAMetaData/Transformation
2. The base transformation xsl files are also stored in MDS.
3. In the PRODUCT installation folder these files are at the location <PRODUCT_HOME>/MDS-Artifacts/OUCSS/AIAMetaData/Transformation
4. The ConfigurationProperties.xml has a BaseXFormFlag flag. If set to true the extended xsl file invokes the base xsl transformation. If the BaseXFormFlag is set to false the extended xsl transformation invokes the implementation specific xsl which can be plugged in into the extended xsl.
5. After modifying the extended xsl transformation file, the steps to update MDS need to be invoked and the server has to be restarted.

   For example: To override the Base xsl in the GetUsageOverView (Usage Download) flow the following needs to be done

   - In the ConfigurationProperties.xml set the BaseXFormFlag for the Usage Download service to false.
   - In the <PRODUCT_HOME>/MDS-Artifacts/OUCSS/AIAMetaData/Transformation folder modify the XformOUMDMOutput_GreenButtonSchemaFormat_extended.xsl file if invoking CCB. If the BPEL process is configured to invoke MDM directly, then modify the XformMDM_DirectCallOutput_GreenButtonSchemaFormat_extended.xsl file. Add the xsl code in this file which needs to be invoked instead of the base xsl.

```
<xsl:template match="/">
<xsl:choose>
<xsl:when test="$useBaseXFormFlag='false'"/>
<xsl:otherwise>
<xsl:call-template name="base"/>
</xsl:otherwise>
</xsl:choose>
</xsl:template>
```

6. Update MDS after the changes are made and restart the SOA server. After the server has been restarted, the custom xsl is invoked.

Data Mapping

Outage Summary Integration Flow

CSS			NMS View GEOGRAPHIC_OUTAGE_AREAS
Element Name	**Parent Name**	**Type**	**Fields**
requestDetails		Outermost Tag	

areaType	requestDetails	Field	
custom	requestDetails	Group	
field1	custom	Group	
name	field1	Field	
value	field1	Field	
field2	custom	Group	
name	field2	Field	
value	field2	Field	
field3	custom	Group	
name	field3	Field	
value	field3	Field	
field4	custom	Group	
name	field4	Field	
value	field4	Field	
field5	custom	Group	
name	field5	Field	
value	field5	Field	
field6	custom	Group	
name	field6	Field	
value	field6	Field	
field7	custom	Group	
name	field7	Field	
value	field7	Field	
field8	custom	Group	
name	field8	Field	
value	field8	Field	
field9	custom	Group	
name	field9	Field	
value	field9	Field	
field10	custom	Group	
name	field10	Field	
value	field10	Field	
responseDetails	Outermost Tag	Group	
GeographicOutageAreasList	responseDetails	List	
area	GeographicOutageAreasList	Field	area
custServed	GeographicOutageAreasList	Field	custServed
custOut	GeographicOutageAreasList	Field	custOut
numOutages	GeographicOutageAreasList	Field	numOutages
earliestBeginTime	GeographicOutageAreasList	Field	
eta	GeographicOutageAreasList	Field	

lastUpdated	GeographicOutageAreasList	Field
custom	responseDetails	Group
field1	custom	Group
name	field1	Field
value	field1	Field
field2	custom	Group
name	field2	Field
value	field2	Field
field3	custom	Group
name	field3	Field
value	field3	Field
field4	custom	Group
name	field4	Field
value	field4	Field
field5	custom	Group
name	field5	Field
value	field5	Field
field6	custom	Group
name	field6	Field
value	field6	Field
field7	custom	Group
name	field7	Field
value	field7	Field
field8	custom	Group
name	field8	Field
value	field8	Field
field9	custom	Group
name	field9	Field
value	field9	Field
field10	custom	Group
name	field10	Field
value	field10	Field

Note: Highlighted Columns are the mapped elements.

Get Outage Screen Info Integration Flow

CSS Message			CCB Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Name**	**Type**
requestDetails		Outermost Tag	WXServiceDetails		Outermost Tag

			head	WXServiceDetails	Group
action	requestDetails	Field	action	head	Field
key1	requestDetails	Group	key1	head	Group
name	key1	Field	name	key1	Field
value	key1	Field	value	key1	Field
key2	requestDetails	Group	key2	head	Group
name	key2	Field	name	key2	Field
value	key2	Field	value	key2	Field
key3	requestDetails	Group	key3	head	Group
name	key3	Field	name	key3	Field
value	key3	Field	value	key3	Field
key4	requestDetails	Group	key4	head	Group
name	key4	Field	name	key4	Field
value	key4	Field	value	key4	Field
key5	requestDetails	Group	key5	head	Group
name	key5	Field	name	key5	Field
value	key5	Field	value	key5	Field
emailAddress	requestDetails	Field	emailAddress	head	Field
webUserId	requestDetails	Field	webUserId	head	Field
ipAddress	requestDetails	Field	ipAddress	head	Field
custom	requestDetails	Group	custom	head	Group
field1	custom	Group	field1	custom	Group
name	field1	Field	name	field1	Field
value	field1	Field	value	field1	Field
field2	custom	Group	field2	custom	Group
name	field2	Field	name	field2	Field
value	field2	Field	value	field2	Field
field3	custom	Group	field3	custom	Group
name	field3	Field	name	field3	Field
value	field3	Field	value	field3	Field
field4	custom	Group	field4	custom	Group
name	field4	Field	name	field4	Field
value	field4	Field	value	field4	Field
field5	custom	Group	field5	custom	Group
name	field5	Field	name	field5	Field
value	field5	Field	value	field5	Field
field6	custom	Group	field6	custom	Group
name	field6	Field	name	field6	Field
value	field6	Field	value	field6	Field
field7	custom	Group	field7	custom	Group

name	field7	Field	name	field7	Field
value	field7	Field	value	field7	Field
field8	custom	Group	field8	custom	Group
name	field8	Field	name	field8	Field
value	field8	Field	value	field8	Field
field9	custom	Group	field9	custom	Group
name	field9	Field	name	field9	Field
value	field9	Field	value	field9	Field
field10	custom	Group	field10	custom	Group
name	field10	Field	name	field10	Field
value	field10	Field	value	field10	Field
responseDetails	Outermost Tag	Group			
outageCodes	responseDetails	Group			
outageCodeList	outageCodes	List			
outageCodeGroupOrder	outageCodeList	Field			
outageCodeName	outageCodeList	Field			
options	outageCodeList	List			
outageValue	options	Field			
outageValueName	options	Field			
accountInfo	responseDetails	Group	mainData	Outermost Tag	Group
name	accountInfo	Field	entityName	mainData	Field
phoneNumberList	accountInfo	List	phoneNbrList	mainData	List
sequence	phoneNumberList	Field	sequence	phoneNbrList	Field
phoneType	phoneNumberList	Field	phoneType	phoneNbrList	Field
phoneTypeDescr	phoneNumberList	Field	phoneTypeDescr	phoneNbrList	Field
phoneNumber	phoneNumberList	Field	phoneNumber	phoneNbrList	Field
premiseList	accountInfo	List	premiseList	mainData	List
premiseId	premiseList	Field	premiseId	premiseList	Field
premiseInfo	premiseList	Field	premiseInfo	premiseList	Field
address1	premiseList	Field	address1	premiseList	Field
city	premiseList	Field	city	premiseList	Field
state	premiseList	Field	state	premiseList	Field
postal	premiseList	Field	postal	premiseList	Field
spList	premiseList	List	spList	premiseList	Field
spId	spList	Field	spId	spList	List
spTypeCode	spList	Field	spType	spList	Field
spTypeDescription	spList	Field	spTypeDescr	spList	Field
custom	responseDetails	Group	custom	responseDetails	Group
field1	custom	Group	field1	custom	Group

name	field1	Field	name	field1	Field
value	field1	Field	value	field1	Field
field2	custom	Group	field2	custom	Group
name	field2	Field	name	field2	Field
value	field2	Field	value	field2	Field
field3	custom	Group	field3	custom	Group
name	field3	Field	name	field3	Field
value	field3	Field	value	field3	Field
field4	custom	Group	field4	custom	Group
name	field4	Field	name	field4	Field
value	field4	Field	value	field4	Field
field5	custom	Group	field5	custom	Group
name	field5	Field	name	field5	Field
value	field5	Field	value	field5	Field
field6	custom	Group	field6	custom	Group
name	field6	Field	name	field6	Field
value	field6	Field	value	field6	Field
field7	custom	Group	field7	custom	Group
name	field7	Field	name	field7	Field
value	field7	Field	value	field7	Field
field8	custom	Group	field8	custom	Group
name	field8	Field	name	field8	Field
value	field8	Field	value	field8	Field
field9	custom	Group	field9	custom	Group
name	field9	Field	name	field9	Field
value	field9	Field	value	field9	Field
field10	custom	Group	field10	custom	Group
name	field10	Field	name	field10	Field
value	field10	Field	value	field10	Field

Note: Highlighted columns are the mapped elements. **requestDetails/key1/value** is the CCB Account Id.

Trouble Calls Interface Integration Flow

CSS			NMS Stored Procedure
Element Name	**Parent Name**	**Type**	**Fields**
requestMessage		OutermostTag	
outageCallId	requestMessage	Field	
			call_source_id
spId	requestMessage	Field	service_point_id
premiseId	requestMessage	Field	

			external_id
accountId	requestMessage	Field	account_number
contactName	requestMessage	Field	first_name
contactNumber	requestMessage	Field	phone
callIdentifier	requestMessage	Field	
callDateTime	requestMessage	Field	
OutageCallAction	requestMessage	Field	update_flag
Status	requestMessage	Field	
comments	requestMessage	Field	call_comment
userId	requestMessage	Field	call_taker
userFirstName	requestMessage	Field	
userLastName	requestMessage	Field	
country	requestMessage	Field	
addressLine1	requestMessage	Field	addr_street *
addressLine2	requestMessage	Field	
addressLine3	requestMessage	Field	
addressLine4	requestMessage	Field	
city	requestMessage	Field	city_state **
geographic	requestMessage	Field	
county	requestMessage	Field	
state	requestMessage	Field	city_state **
postal	requestMessage	Field	
locationType	requestMessage	Field	
blockNumber	requestMessage	Field	addr_building
location1	requestMessage	Field	addr_street *
location2	requestMessage	Field	addr_cross_street
locationCity	requestMessage	Field	addr_city_state **
locationState	requestMessage	Field	addr_city_state **
meetDateTime	requestMessage	Field	
meetType	requestMessage	Field	
outageCodes	requestMessage	Field	trouble_code
transformerId	requestMessage	Field	
callbackRequested	requestMessage	Field	callback_flag
callbackDateTime	requestMessage	Field	
callbackNumber	requestMessage	Field	
customElements			
custom	requestMessage	Group	
field1	custom	Group	
name	field1	Field	
value	field1	Field	

field2	custom	Group
name	field2	Field
value	field2	Field
field3	custom	Group
name	field3	Field
value	field3	Field
field4	custom	Group
name	field4	Field
value	field4	Field
field5	custom	Group
name	field5	Field
value	field5	Field
field6	custom	Group
name	field6	Field
value	field6	Field
field7	custom	Group
name	field7	Field
value	field7	Field
field8	custom	Group
name	field8	Field
value	field8	Field
field9	custom	Group
name	field9	Field
value	field9	Field
field10	custom	Group
name	field10	Field
value	field10	Field
processResponse	Outermost Tag	Group
result	processResponse	Field
custom	processResponse	Group
field1	custom	Group
name	field1	Field
value	field1	Field
field2	custom	Group
name	field2	Field
value	field2	Field
field3	custom	Group
name	field3	Field
value	field3	Field
field4	custom	Group

name	field4	Field
value	field4	Field
field5	custom	Group
name	field5	Field
value	field5	Field
field6	custom	Group
name	field6	Field
value	field6	Field
field7	custom	Group
name	field7	Field
value	field7	Field
field8	custom	Group
name	field8	Field
value	field8	Field
field9	custom	Group
name	field9	Field
value	field9	Field
field10	custom	Group
name	field10	Field
value	field10	Field

Note: Highlighted columns are the mapped elements:

*** addr_street**: If the trouble call is related to an SP, the integration maps it to the **addressLine1**. If fuzzy call or relates to a non-premise outage, the integration maps it to location1.

**** adrr_city_state**: If the trouble call is related to an SP, the integration maps it to the **city, state**. If it is not related to an SP, or the call relates to a non-premise outage, the integration maps it to location city.location state.

The following table contains additional mapping information about the Trouble Call Interface.

NMS Field	CSS Message Element	Notes
Call Source Id call_source_id	5	This value is normally populated by the integration to indicate the source of the call. Since trouble calls can be created from different external systems such as CCB, IVR, or a web call entry. Each external system sending trouble calls to NMS has a unique call_source_id. This determines the origin of the trouble call and ensures the external id passed to NMS is unique. NMS prefixes this value to the external id to make it unique. Default is 5.
Service Point Id service_point_id	spId	This is the SP Id. For non-premise outages, CSS passes an empty value.
External Id external_id		NMS generate the external Id.
Account Number	accountId	This is the customer's account Id.

account_number		For non-premise outages, CSS passes an empty value.
Trouble Code trouble_code	outageCodes	Trouble code mapping setup between CSS and NMS must be the same. In NMS, the total length of the string is the total number of distinct groups in the SRS_TROUBLE_CODES table.
Call Time call_time		Default from NMS.
CallbackFlag callback_flag	callbackRequested	Allowed values in NMS: •0 = callback not requested •1 = callback requested NMS defaults to 1 if no value is passed by CSS.
Contact Phone phone	contactNumber	CSS passes the contact phone when reporting an outage for a premise or non-premise. When CSS passes this field, it will only contain the numeric values of the phone number.
Contact Name first_name	contactName	CSS passes the contact name when reporting an outage for a premise or non-premise.
Address Street addr_street	address1 or location1	CSS passes the premise address1 when reporting an outage for a premise. CSS passes the location1 when reporting an outage for a non-premise. (It can contain a street name or free-format location description).
Address Street 2 addr_cross_street	location2	CSS can pass location2, which is a cross street, when reporting an outage for a non-premise. This is optional in CSS.
Address City State addr_city_state	city state or locationCity locationState	CSS passes the premise city and state when reporting an outage for a premise. CSS passes the location city and location state when reporting an outage for a non-premise. Integration concatenates the city and state passed by CSS. The delimiter provided by the integration is always comma (,). If city or state is blank, no delimiter (,) is needed.
Call Id call_id	userId	CSS passes 'OUCSS '
Call Comment call_comment	comments	
Update Existing Record Flag update_flag	outageCallAction	Allowed values in NMS: •0 = insert new call •1 = update existing call NMS defaults to 0 if no value is passed by CSS.

Get Alerts Integration Flow

CSS Message			CCB Message			NMS Stored Procedure	
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Name**	**Type**	**Job History Fields**	**Switching Plan Fields**
WXGetCCBAlerts		Outermost Tag	WXServiceDetails		Outermost Tag		
head	WXGetCCBAlerts		head	WXGetCCBAlerts	Group		
action	head	Field	action	head	Field		
key1	head	Group	key1	head	Group		
name	key1	Field	name	key1	Field		
value	key1	Field	value	key1	Field	ACCOUNTNUMBER	ACCOUNTNUMBER
key2	head	Group	key2	head	Group		
name	key2	Field	name	key2	Field		
value	key2	Field	value	key2	Field		
key3	head	Group	key3	head	Group		
name	key3	Field	name	key3	Field		
value	key3	Field	value	key3	Field		
key4	head	Group	key4	head	Group		
name	key4	Field	name	key4	Field		
value	key4	Field	value	key4	Field		
key5	head	Group	key5	head	Group		
name	key5	Field	name	key5	Field		
value	key5	Field	value	key5	Field		
emailAddress	head	Field	emailAddress	head	Field		
webUserId	head	Field	webUserId	head	Field		
ipAddress	head	Field	ipAddress	head	Field		
mainData	Outermost Tag	Group	mainData	Outermost Tag	Group		
alertType	mainData	List	alertType	mainData	List	From Configuration Properties	From Configuration Properties
header	OutageDetailList	List	header	OutageDetailList	List	From Configuration Properties	From Configuration Properties
icon	OutageDetailList	Field	icon	OutageDetailList	Field	From Configuration Properties	From Configuration Properties
description	OutageDetailList	Field	description	OutageDetailList	Field	From Configuration Properties	From Configuration Properties
link	OutageDetailList	Field	link	OutageDetailList	Field	From Configuration Properties	From Configuration Properties
date	OutageDetailList	Field	date	OutageDetailList	Field		
priority	OutageDetailList	Field	priority	OutageDetailList	Field		
custom	responseDetails	Group	custom	responseDetails	Group		
field1	custom	Group	field1	custom	Group		
name	field1	Field	name	field1	Field		
value	field1	Field	value	field1	Field		

field2	custom	Group	field2	custom	Group
name	field2	Field	name	field2	Field
value	field2	Field	value	field2	Field
field3	custom	Group	field3	custom	Group
name	field3	Field	name	field3	Field
value	field3	Field	value	field3	Field
field4	custom	Group	field4	custom	Group
name	field4	Field	name	field4	Field
value	field4	Field	value	field4	Field
field5	custom	Group	field5	custom	Group
name	field5	Field	name	field5	Field
value	field5	Field	value	field5	Field
field6	custom	Group	field6	custom	Group
name	field6	Field	name	field6	Field
value	field6	Field	value	field6	Field
field7	custom	Group	field7	custom	Group
name	field7	Field	name	field7	Field
value	field7	Field	value	field7	Field
field8	custom	Group	field8	custom	Group
name	field8	Field	name	field8	Field
value	field8	Field	value	field8	Field
field9	custom	Group	field9	custom	Group
name	field9	Field	name	field9	Field
value	field9	Field	value	field9	Field
field10	custom	Group	field10	custom	Group
name	field10	Field	name	field10	Field
value	field10	Field	value	field10	Field

Notes:

- Highlighted Columns are the mapped elements. **WXGetCCBAlerts/head/key1/value** is the CCB Account Id.
- When Job History Stored Procedure return current outage(s), the current outage alert elements returned to CSS are obtained from the Configuration Properties File.
- When Switching History Stored Procedure returns planned outage(s), the planned outage alert elements returned back to CSS are obtained from the Configuration Properties File.

Outage Public Detail Integration Flow

CSS			NMS Table GEOGRAPHIC_OUTAGE_AREAS_D
Element Name	**Parent Name**	**Type**	**Fields**
requestDetails		OutermostTag	
areaType	requestDetails	Field	
area	requestDetails	Field	AREA
custom	requestDetails	Group	AREA_TYPE
field1	custom	Group	

name	field1	Field	
value	field1	Field	
field2	custom	Group	
name	field2	Field	
value	field2	Field	
field3	custom	Group	
name	field3	Field	
value	field3	Field	
field4	custom	Group	
name	field4	Field	
value	field4	Field	
field5	custom	Group	
name	field5	Field	
value	field5	Field	
field6	custom	Group	
name	field6	Field	
value	field6	Field	
field7	custom	Group	
name	field7	Field	
value	field7	Field	
field8	custom	Group	
name	field8	Field	
value	field8	Field	
field9	custom	Group	
name	field9	Field	
value	field9	Field	
field10	custom	Group	
name	field10	Field	
value	field10	Field	
responseDetails	OutermostTag	Group	
GeographicOutageAreaDetails	responseDetails	List	
area	GeographicOutageAreaDetails	Field	AREA
areaType	GeographicOutageAreaDetails	Field	AREA_TYPE
eventId	GeographicOutageAreaDetails	Field	EVENT_ID
outageType	GeographicOutageAreaDetails	Field	OUTAGE_TYPE
Status	GeographicOutageAreaDetails	Field	STATUS
custOut	GeographicOutageAreaDetails	Field	CUST_OUT
earliestBeginTime	GeographicOutageAreaDetails	Field	BEGIN_TIME
eta	GeographicOutageAreaDetails	Field	EST_REST_TIME
lastUpdated	GeographicOutageAreaDetails	Field	LAST_UPDATE_TIME

custom	responseDetails	Group
field1	custom	Group
name	field1	Field
value	field1	Field
field2	custom	Group
name	field2	Field
value	field2	Field
field3	custom	Group
name	field3	Field
value	field3	Field
field4	custom	Group
name	field4	Field
value	field4	Field
field5	custom	Group
name	field5	Field
value	field5	Field
field6	custom	Group
name	field6	Field
value	field6	Field
field7	custom	Group
name	field7	Field
value	field7	Field
field8	custom	Group
name	field8	Field
value	field8	Field
field9	custom	Group
name	field9	Field
value	field9	Field
field10	custom	Group
name	field10	Field
value	field10	Field

Outage Detail Integration Flow

CSS Message			CCB Message			NMS Stored Procedure	
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Name**	**Type**	**Job History Fields**	**Switching Plan Fields**
requestDetails		Outermost Tag	WXServiceDetails		Outermost Tag		

			head	WXServiceDetails	Group		
action	requestDetails	Field	action	head	Field		
key1	requestDetails	Group	key1	head	Group		
name	key1	Field	name	key1	Field		
value	key1	Field	value	key1	Field	ACCOUNTNUMBER	ACCOUNTNUMBER
key2	requestDetails	Group	key2	head	Group		
name	key2	Field	name	key2	Field		
value	key2	Field	value	key2	Field		
key3	requestDetails	Group	key3	head	Group		
name	key3	Field	name	key3	Field		
value	key3	Field	value	key3	Field		
key4	requestDetails	Group	key4	head	Group		
name	key4	Field	name	key4	Field		
value	key4	Field	value	key4	Field		
key5	requestDetails	Group	key5	head	Group		
name	key5	Field	name	key5	Field		
value	key5	Field	value	key5	Field		
custom	requestDetails	Group	custom	head	Group		
field1	custom	Group	field1	custom	Group		
name	field1	Field	name	field1	Field		
value	field1	Field	value	field1	Field		
field2	custom	Group	field2	custom	Group		
name	field2	Field	name	field2	Field		
value	field2	Field	value	field2	Field		
field3	custom	Group	field3	custom	Group		
name	field3	Field	name	field3	Field		
value	field3	Field	value	field3	Field		
field4	custom	Group	field4	custom	Group		
name	field4	Field	name	field4	Field		
value	field4	Field	value	field4	Field		
field5	custom	Group	field5	custom	Group		
name	field5	Field	name	field5	Field		
value	field5	Field	value	field5	Field		
field6	custom	Group	field6	custom	Group		
name	field6	Field	name	field6	Field		
value	field6	Field	value	field6	Field		
field7	custom	Group	field7	custom	Group		
name	field7	Field	name	field7	Field		
value	field7	Field	value	field7	Field		
field8	custom	Group	field8	custom	Group		
name	field8	Field	name	field8	Field		
value	field8	Field	value	field8	Field		
field9	custom	Group	field9	custom	Group		
name	field9	Field	name	field9	Field		
value	field9	Field	value	field9	Field		

field10	custom	Group	field10	custom	Group		
name	field10	Field	name	field10	Field		
value	field10	Field	value	field10	Field		
responseDetails	Outermost Tag	Group					
OutageDetailList	responseDetails	List					
plannedOutageFlag	OutageDetailList	List				N	Y
spId	OutageDetailList	Field				CID	CID
crewStatus	OutageDetailList	Field				ALARM_STATE_DESC	STATE
cause	OutageDetailList	Field				DESCRIPTION	WORK_DESCRIPTION
custOut	OutageDetailList	Field				NUM_CUST_OUT	
earliestBeginTime	OutageDetailList	Field				BEGIN_TIME	START_DATE
eta	OutageDetailList	Field				EST_REST_TIME	END_DATE
lastUpdated	OutageDetailList	Field				LAST_UPDATE_TIME	
accountInfo	responseDetails	Group	mainData	Outermost Tag	Group		
name	accountInfo	Field	entityName	mainData	Field		
phoneNumberList	accountInfo	List	phoneNbrList	mainData	List		
sequence	phoneNumberList	Field	sequence	phoneNbrList	Field		
phoneType	phoneNumberList	Field	phoneType	phoneNbrList	Field		
phoneTypeDescr	phoneNumberList	Field	phoneTypeDescr	phoneNbrList	Field		
phoneNumber	phoneNumberList	Field	phoneNumber	phoneNbrList	Field		
premiseList	accountInfo	List	premiseList	mainData	List		
premiseId	premiseList	Field	premiseId	premiseList	Field		
premiseInfo	premiseList	Field	premiseInfo	premiseList	Field		
address1	premiseList	Field	address1	premiseList	Field		
city	premiseList	Field	city	premiseList	Field		
state	premiseList	Field	state	premiseList	Field		
postal	premiseList	Field	postal	premiseList	Field		
spList	premiseList	List	spList	premiseList	Field		
spId	spList	Field	spId	spList	List		
spTypeCode	spList	Field	spType	spList	Field		
spTypeDescription	spList	Field	spTypeDescr	spList	Field		
custom	responseDetails	Group	custom	responseDetails	Group		
field1	custom	Group	field1	custom	Group		
name	field1	Field	name	field1	Field		
value	field1	Field	value	field1	Field		
field2	custom	Group	field2	custom	Group		
name	field2	Field	name	field2	Field		
value	field2	Field	value	field2	Field		
field3	custom	Group	field3	custom	Group		
name	field3	Field	name	field3	Field		
value	field3	Field	value	field3	Field		
field4	custom	Group	field4	custom	Group		
name	field4	Field	name	field4	Field		
value	field4	Field	value	field4	Field		

field5	custom	Group	field5	custom	Group
name	field5	Field	name	field5	Field
value	field5	Field	value	field5	Field
field6	custom	Group	field6	custom	Group
name	field6	Field	name	field6	Field
value	field6	Field	value	field6	Field
field7	custom	Group	field7	custom	Group
name	field7	Field	name	field7	Field
value	field7	Field	value	field7	Field
field8	custom	Group	field8	custom	Group
name	field8	Field	name	field8	Field
value	field8	Field	value	field8	Field
field9	custom	Group	field9	custom	Group
name	field9	Field	name	field9	Field
value	field9	Field	value	field9	Field
field10	custom	Group	field10	custom	Group
name	field10	Field	name	field10	Field
value	field10	Field	value	field10	Field

Note: Highlighted Columns are the mapped elements. Key 1 value is the CCB Account Id.

Get Usage Detail Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXUsageDetail		OutermostTag	WX-RETWSSTOUMappingService		OutermostTag
head	WXUsageDetail	Group	input	WX-RETWSSTOUMappingService	Group
action	head	Field			
key1	head	Group			
name	key1	Field			
value	key1	Field	externalAccountId	input	Field
key2	head	Group			
name	key2	Field			
value	key2	Field			
key3	head	Group			
name	key3	Field			
value	key3	Field			
key4	head	Group			
name	key4	Field			
value	key4	Field			
key5	head	Group			
name	key5	Field			
value	key5	Field			

emailAddress	head	Field			
webUserId	head	Field			
ipAddress	head	Field			
saId	head	Field			
displayMode	head	Field	displayMode	input	Field
previousNext	head	Field	previousNext	input	Field
overlayMode	head	Field	overlayMode	input	Field
referenceDateTime	head	Field	referenceDateTime	Input	Field
startDate	head	Field	startDate	input	Field
endDate	head	Field	endDate	input	Field
			usageSubscriptions	input	List
			usId	usageSubscriptions	Field
			usExternalId	usageSubscriptions	Field
			overrideUom	input	Field
			overrideTou	input	Field
			overrideSqi	input	Field
customElements	head	Group	customElements	input	Group
mainData	WXUsageDetail	Group	output	WX-RETWSSTOUMappingService	Group
results	mainData	List	results	output	List
			usId	results	Field
saId	results	Field	usExternalId	results	Field
saInfo	results	Field	usInfo	results	Field
cisDivision	results	Field			
cisDivisionDescription	results	Field			
saType	results	Field	usType	results	Field
saTypeDescription	results	Field	usTypeDescription	results	Field
serviceType	results	Field	serviceType	results	Field
serviceTypeDescription	results	Field	serviceTypeDescription	results	Field
isSkipped	results	Field	isSkipped	results	Field
skipReasonDescription	results	Field	skipReasonDescription	results	Field
			skipReason	results	Field
usageUom	results	Field	usageUom	results	Field
usageUomDescription	results	Field	usageUomDescription	results	Field
usageSqi	results	Field	usageSqi	results	Field
usageDescription	results	Field	usageSqiDescription	results	Field
overlayUom	results	Field	overlayUom	results	Field
overlayUomDescription	results	Field	overlayUomDescription	results	Field
latestMeasurementDateTime	results	Field	latestMeasurementDateTime	results	Field
periods	results	List	periods	results	List
dateTime	periods	Field	dateTime	periods	Field
touQuantities	periods	List	touUsages	periods	List
tou	touQuantities	Field	tou	touUsages	Field
touDescription	touQuantities	Field	touDescription	touUsages	Field
quantity	touQuantities	Field	quantity	touUsages	Field

overlayQuantities	periods	List	overlayQuantities	periods	List
quantity	overlayQuantities	Field	quantity	overlayQuantities	Field
customElements	results	Group	customElements	results	Group
customElements	mainData	Group	customElements	output	Group
errorInformation	mainData	Group	errorInformation	mainData	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
customElements	WXUsageDetail	Group			

Notes:

- Highlighted Columns are the mapped elements. . Key 1 value is the CCB Account Id.
- When CSS do not pass the referenceDateTime, it is defaulted to the currentDateTime.

Get Consumption Summary Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXGetConsumptionSummary		Outermost Tag	WX-GetScalarConsumptionSumm ary		Outermost Tag
head	WXGetConsumption Summary	Group	head	WX-GetScalarConsumptionSummary	Group
action	head	Field	action	head	Field
key1	head	Group	key1	head	Group
name	key1	Field	name	key1	Field
value	key1	Field	value	key1	Field
key2	head	Group	key2	head	Group
name	key2	Field	name	key2	Field
value	key2	Field	value	key2	Field
key3	head	Group	key3	head	Group
name	key3	Field	name	key3	Field
value	key3	Field	value	key3	Field
key4	head	Group	key4	head	Group
name	key4	Field	name	key4	Field
value	key4	Field	value	key4	Field
key5	head	Group	key5	head	Group
name	key5	Field	name	key5	Field
value	key5	Field	value	key5	Field
emailAddress	head	Field	emailAddress	head	Field
webUserId	head	Field	webUserID	head	Field
ipAddress	head	Field	ipAddress	head	Field
saId	head	Field	usId	head	

			usExternalId		
mainData	WXGetConsumption Summary	Group	mainData	WX-GetScalarConsumptionSummary	Group
saList	mainData	List	usList	mainData	List
			usExternalId	usList	Field
saId	saList	Field	usId	usList	Field
			usExternalId		
saInfo	saList	Field	usInfo	usList	Field
saType	saList	Field	usType	usList	Field
saTypeDescription	saList	Field	usTypeDescription	usList	Field
serviceType	saList	Field	serviceType	usList	Field
serviceTypeDescription	saList	Field	serviceTypeDescription	usList	Field
uomDescription	saList	Field	uomDescription	usList	Field
division	saList	Field			Field
divisionDescription	saList	Field			Field
history	saList	Group	history	usList	Group
sq	history	List	serviceQuantityList	history	List
uom	sq	Field	uom	serviceQuantityList	Field
tou	sq	Field	tou	serviceQuantityList	Field
sqi	sq	Field	sqi	serviceQuantityList	Field
date	sq	Field	date	serviceQuantityList	Field
quantity	sq	Field	q	serviceQuantityList	Field
latestSQ	saList	Group	latestSQ	usList	Group
uom	latestSQ	Field	uom	latestSQ	Field
tou	latestSQ	Field	tou	latestSQ	Field
sqi	latestSQ	Field	sqi	latestSQ	Field
quantity	latestSQ	Field	q	latestSQ	Field
billId	mainData	Field			List
errorInformation	WXGetConsumption Summary	Group	errorInformation	mainData	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
custom	WXGetConsumption Summary	Group	custom	WX-GetScalarConsumptionSummary	Group
field1	custom	Group	field1	custom	Group
name	field1	Field	name	field1	Field
value	field1	Field	value	field1	Field
field2	custom	Group	field2	custom	Group
name	field2	Field	name	field2	Field
value	field2	Field	value	field2	Field
field3	custom	Group	field3	custom	Group
name	field3	Field	name	field3	Field
value	field3	Field	value	field3	Field
field4	custom	Group	field4	custom	Group

name	field4	Field	name	field4	Field
value	field4	Field	value	field4	Field
field5	custom	Group	field5	custom	Group
name	field5	Field	name	field5	Field
value	field5	Field	value	field5	Field
field6	custom	Group	field6	custom	Group
name	field6	Field	name	field6	Field
value	field6	Field	value	field6	Field
field7	custom	Group	field7	custom	Group
name	field7	Field	name	field7	Field
value	field7	Field	value	field7	Field
field8	custom	Group	field8	custom	Group
name	field8	Field	name	field8	Field
value	field8	Field	value	field8	Field
field9	custom	Group	field9	custom	Group
name	field9	Field	name	field9	Field
value	field9	Field	value	field9	Field
field10	custom	Group	field10	custom	Group
name	field10	Field	name	field10	Field
value	field10	Field	value	field10	Field

Note: Highlighted Columns are the mapped elements. Key 1 value is the CCB Account Id.

Add Scalar Meter Read Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXCreateMeterRead		OutermostTag	WX-CreateSelfServiceMeterRead		OutermostTag
head	WXCreateMeterRead	Group	head	WX-CreateSelfServiceMeterRead	Group
action	head	Field	action	head	Field
key1	head	Group	key1	head	Group
name	key1	Field	name	head	Field
value	key1	Field	value	head	Field
key2	head	Group	key2	head	Group
name	key2	Field	name	head	Field
value	key2	Field	value	head	Field
key3	head	Group	key3	head	Group
name	key3	Field	name	head	Field
value	key3	Field	value	head	Field
key4	head	Group	key4	head	Group
name	key4	Field	name	head	Field
value	key4	Field	value	head	Field
key5	head	Group	key5	head	Group
name	key5	Field	name	head	Field

value	key5	Field	value	head	Field
emailAddress	head	Field	emailAddress	head	Field
webUserId	head	Field	webUserID	head	Field
ipAddress	head	Field	ipAddress	head	Field
saId	head	Field	usId	head	Field
			usExternalId	head	Field
spId	head	Field	spId	head	Field
mainData	WXCreateMeterRead	Group	mainData	WX-CreateSelfServiceMeterRead	Group
referenceId	mainData	Field	referenceId	mainData	Field
meterConfigurationId	mainData	Field	deviceConfigurationId	mainData	Field
readDateTime	mainData	Field	readDateTime	mainData	Field
spMeterCollection	mainData	Group	spDeviceConfigurations	mainData	Group
spMeters	spMeterCollection	List	spDeviceConfigurationList	spDeviceConfigurations	List
spId	spMeters	Field	spId	spDeviceConfigurationList	Field
meterConfigurationId	spMeters	Field	deviceConfigurationId	spDeviceConfigurationList	Field
spMeterInfo	spMeters	Field	spDeviceConfigurationInfo	spDeviceConfigurationList	Field
registerCollection	spMeters	Group	measuringComponents	spDeviceConfigurationList	Group
registers	registerCollection	List	measuringComponentsList	measuringComponents	List
readSequence	registers	Field	readSequence	measuringComponentsList	Field
registerId	registers	Field	measuringComponentId	measuringComponentsList	Field
registerInfo	registers	Field	measuringComponentInfo	measuringComponentsList	Field
lastReadDateTime	registers	Field	lastReadDateTime	measuringComponentsList	Field
lastReading	registers	Field	lastReading	measuringComponentsList	Field
			readDateTime	measuringComponentsList	Field
reading	registers	Field	reading	measuringComponentsList	Field
errorInformation	WXCreateMeterRead	Group	errorInformation	mainData	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
custom	WXCreateMeterRead	Group			
field1	custom	Group			
name	field1	Field			
value	field1	Field			
field2	custom	Group			
name	field2	Field			
value	field2	Field			
field3	custom	Group			
name	field3	Field			
value	field3	Field			
field4	custom	Group			
name	field4	Field			
value	field4	Field			

field5	custom	Group
name	field5	Field
value	field5	Field
field6	custom	Group
name	field6	Field
value	field6	Field
field7	custom	Group
name	field7	Field
value	field7	Field
field8	custom	Group
name	field8	Field
value	field8	Field
field9	custom	Group
name	field9	Field
value	field9	Field
field10	custom	Group
name	field10	Field
value	field10	Field

Note: Highlighted Columns are the mapped elements. Key 1 value is the CCB Account Id.

Usage Download Integration Flow

CSS Message (Input)			MDM Message (Residential)			MDM Message (Commercial)		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**
input		OutermostTag	WX-GetUsageOverview		OutermostTag	WX-MultipleAccountUsagesDownload		OutermostTag
						head	WX-MultipleAccountUsagesDownload	Group
action	input	Field						
selfServiceKeys	input	List				selfServiceKeys	head	List
key1	selfServiceKeys	Group				key1	selfServiceKeys	Group
name	key1	Field				name	key1	Field
value	key1	Field	externalAccountId			value	key1	Field
key2	selfServiceKeys	Group				key2	selfServiceKeys	Group
name	key2	Field				name	key2	Field
value	key2	Field				value	key2	Field
key3	selfServiceKeys	Group				key3	selfServiceKeys	Group

name	key3	Field				name	key3	Field
value	key3	Field				value	key3	Field
key4	selfServiceKeys	Group				key4	selfServiceKeys	Group
name	key4	Field				name	key4	Field
value	key4	Field				value	key4	Field
key5	selfServiceKeys	Group				key5	selfServiceKeys	Group
name	key5	Field				name	key5	Field
value	key5	Field				value	key5	Field
emailAddress	input	Field				emailAddress	head	Field
webUserId	input	Field				webUserId	head	Field
ipAddress	input	Field				ipAddress	head	Field
formatAs	input	Field						
			overviewMode (default to USGD)	input	Field	overviewMode (default to USGD)	head	Field
startDate	input	Field	refererenceDateTime	input	Field	refererenceDateTime	head	Field
numberOfDays	input	Field	usageHistory	input	Field	usageHistory	head	Field
usageSubscriptions	input	List			List			
			qtyToDateStartDateTime	usageSubscriptions	Field			
			externalId	usageSubscriptions	Field			
saId	usageSubscriptions	Field						
spId	usageSubscriptions	Field						
usId	usageSubscriptions	Field	usId	usageSubscriptions	Field			
custom	input	Group	customElements	input	Group			

MDM Message (Output)			**Usage Download Externalized Transform Message** (In MDS:<PRODUCT_HOME>/MDS-Artifacts/OUCSS/AIAMetaData/Transformation folder modify the XformOUMDMOutput_GreenButtonSchemaFormat_extended.xsl)		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WX-GetUsageOverview		OutermostTag	UsageData		OutermostTag
output	WX-GetUsageOverview	Group			
results	output	List	usageDetails	UsageData	List

			serviceInfo	usageDetails	Group
usId	results	Field			
usInfo	results	Field	saInfo	serviceInfo	Field
externalId	results	Field			
externalAccountId	results	Field	accountId	serviceInfo	Field
usType	results	Field			
usTypeDescription	results	Field			
serviceType	results	Field			
serviceTypeDescription	results	Field	serviceTypeDescription	serviceInfo	Field
skipped	results	Field			
skipReason	results	Field			
skipReasonDescription	results	Field			

startDateTime	results	Field	startDateTime	serviceInfo	Field
endDateTime	results	Field			
uom	results	Field	uom	usageDetails	Group
uomDescription	results	Field	uomDescription	usageDetails	Field
secondsPerInterval	results	Field	spi		
latestMeasurementDateTime	results	Field	latestMeasurementDateTime	usageDetails	Field
msrs	resluts	Group			
mList	msrs	List	measurements	usageDetails	List
s	mList	Field	sequence	measurements	Field
q	mList	Field	quantity	measurements	Field
qtyToDateStartDateTime	results	Field	qtyToDateStartDateTime	usageDetails	Field
qtyToDate	results	Field	qtyToDate	usageDetails	Field
customElements	head	Group			

Direct Usage Overview Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXUsageOverview		OutermostTag	WX-GetUsageOverview		OutermostTag
head	WXUsageOverview	Group	input	WX-GetUsageOverview	Group
action	head	Field			
key1	head	Group			
name	key1	Field			
value	key1	Field	externalAccountId	input	Field
key2	head	Group			
name	key2	Field			

value	key2	Field			
key3	head	Group			
name	key3	Field			
value	key3	Field			
key4	head	Group			
name	key4	Field			
value	key4	Field			
key5	head	Group			
name	key5	Field			
value	key5	Field			
emailAddress	head	Field			
webUserId	head	Field			
ipAddress	head	Field			
saId	head	Field			
referenceDateTime	head	Field	refererenceDateTime	input	Field
usageDays	head	Field	usageHistory	input	Field
		Field	overviewMode	input	Field
			usageSubscriptions	input	List
			usId	usageSubscriptions	Field
			externalId	usageSubscriptions	Field
			qtyToDateStartDateTime	usageSubscriptions	Field
customElements	head	Group	customElements	input	Group
mainData	WXUsageOverview	Group	output	WX-GetUsageOverview	Group
results	mainData	List	results	output	List
			externalAccountId	results	Field
			externalId	results	Field
saId	results	Field	usId	results	Field
saInfo	results	Field	usInfo	results	Field
cisDivision	results	Field			
cisDivisionDescription	results	Field			
saType	results	Field	usType	results	Field
saTypeDescription	results	Field	usTypeDescription	results	Field
serviceType	results	Field	serviceType	results	Field
serviceTypeDescription	results	Field	serviceTypeDescription	results	Field
isSkipped	results	Field	skipped	results	Field
skipReasonDescription	results	Field	skipReasonDescription	results	Field
			skipReason	results	Field
uom	results	Field	uom	results	Field
uomDescription	results	Field	uomDescription	results	Field
spi	results	Field	secondsPerInterval	results	Field
latestMeasurementDateTime	results	Field	latestMeasurementDateTime	results	Field
qtyToDateStartDateTime	results	Field	qtyToDateStartDateTime	results	Field
qtyToDate	results	Field	qtyToDate	results	Field
startDateTime	results	Field	startDateTime	results	Field
endDateTime	results	Field	endDateTime	results	Field

			msrs	results	Group
measurements	results	List	mList	msrs	List
sequence	measurements	Field	s	mList	Field
quantity	measurements	Field	q	mList	Field
customElements	results	Group	customElements	results	Group
errorInformation	WXUsageOverview	Group	errorInformation	output	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
			customElements	output	Group
custom	WXUsageOverview	Group			
field1	custom	Group			
name	field1	Field			
value	field1	Field			
field2	custom	Group			
name	field2	Field			
value	field2	Field			
field3	custom	Group			
name	field3	Field			
value	field3	Field			
field4	custom	Group			
name	field4	Field			
value	field4	Field			
field5	custom	Group			
name	field5	Field			
value	field5	Field			
field6	custom	Group			
name	field6	Field			
value	field6	Field			
field7	custom	Group			
name	field7	Field			
value	field7	Field			
field8	custom	Group			
name	field8	Field			
value	field8	Field			
field9	custom	Group			
name	field9	Field			
value	field9	Field			
field10	custom	Group			
name	field10	Field			
value	field10	Field			

Notes:

- Highlighted Columns are the mapped elements. . Key 1 value is the CCB Account Id

- When CSS do not pass the referenceDateTime, it is defaulted to the currentDateTime.

Multiple Account Usage Aggregation Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXMultipleAccount TOUUsagesByServiceType		OutermostTag	WX-MultipleAccount TOUUsagesByServiceType		OutermostTag
head	WXMultipleAccount TOUUsagesByServiceType	Group	head	WX-MultipleAccount TOUUsagesByServiceType	Group
selfServiceKeys	head	List	selfServiceKeys	head	List
key1	selfServiceKeys	Group	key1	selfServiceKeys	Group
name	key1	Field	name	key1	Field
value	key1	Field	value	key1	Field
key2	selfServiceKeys	Group	key2	selfServiceKeys	Group
name	key2	Field	name	key2	Field
value	key2	Field	value	key2	Field
key3	selfServiceKeys	Group	key3	selfServiceKeys	Group
name	key3	Field	name	key3	Field
value	key3	Field	value	key3	Field
key4	selfServiceKeys	Group	key4	selfServiceKeys	Group
name	key4	Field	name	key4	Field
value	key4	Field	value	key4	Field
key5	selfServiceKeys	Group	key5	selfServiceKeys	Group
name	key5	Field	name	key5	Field
value	key5	Field	value	key5	Field
emailAddress	head	Field	emailAddress	head	Field
webUserId	head	Field	webUserId	head	Field
ipAddress	head	Field	ipAddress	head	Field
displayMode	head	Field	displayMode	input	Field
previousNext	head	Field	previousNext	input	Field
overlayMode	head	Field	overlayMode	input	Field
referenceDateTime	head	Field	referenceDateTime	input	Field
startDate	head	Field	startDate	head	Field
endDate	head	Field	endDate	head	Field
latestMeasurementDateTime	head	Field	latestMeasurementDateTime	head	Field
			customElements	head	Group
mainData	WXMultipleAccount TOUUsagesByServiceType	Group	mainData	WX-MultipleAccount TOUUsagesByServiceType	Group
results	mainData	List	results	mainData	List
serviceType	results	Field	serviceType	results	Field
serviceTypeDescription	results	Field	serviceTypeDescription	results	Field

latestMeasurementDateTime	results	Field	latestMeasurementDateTime	results	Field
usageUom	results	Field	usageUom	results	Field
usageUomDescription	results	Field	usageUomDescription	results	Field
usageSQI	results	Field	usageSQI	results	Field
usageSQIDescription	results	Field	usageSQIDescription	results	Field
overlayUom	results	Field	overlayUOM	results	Field
overlayUomDescription	results	Field	overlayUOMDescription	results	Field
periods	results	List	periods	results	List
dateTime	periods	Field	dateTime	periods	Field
touQuantities	periods	List	touQuantities	periods	List
tou	touQuantities	Field	tou	touQuantities	Field
touDescription	touQuantities	Field	touDescription	touQuantities	Field
quantity	touQuantities	Field	quantity	touQuantities	Field
overlayQuantities	periods	List	overlayQuantities	periods	List
quantity	overlayQuantities	Field	quantity	overlayQuantities	Field
earliestUSStartDateTime	results	Field	earliestUSStartDateTime	results	Field
			customElements	results	Group
skippedAccounts	mainData	List	skippedAccountUS	mainData	List
accountId	skippedAccounts	Field	externalAccountId	skippedAccountUS	Field
saId	skippedAccounts	Field	usId	skippedAccountUS	Field
			usExternalId	skippedAccountUS	Field
saInformation	skippedAccounts	Field	usInfo	skippedAccountUS	Field
skipReasonDescription	skippedAccounts	Field	skipReasonDescription	skippedAccountUS	Field
serviceType	skippedAccounts	Field	serviceType	skippedAccountUS	Field
errorInformation	mainData	Group	errorInformation	mainData	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
			customElements	mainData	Group

Notes:

- Highlighted Columns are the mapped elements. . Key 1 value is the CCB Account Id
- Attribute datetimeTagFormat is passed from CSS to MDM.

Multiple Account Usage Comparison Integration Flow

CSS Message			MDM Message		
Element Name	**Parent Name**	**Type**	**Element Name**	**Parent Element**	**Type**
WXMultipleAccount		OutermostTag	WX-MultipleAccount		OutermostTag
UsagesByServiceType			UsagesByServiceType		

head	WXMultipleAccount UsagesByServiceType	Group	head	WX-MultipleAccount UsagesByServiceType	Group
selfServiceKeys	head	List	selfServiceKeys	head	List
key1	selfServiceKeys	Group	key1	selfServiceKeys	Group
name	key1	Field	name	key1	Field
value	key1	Field	value	key1	Field
key2	selfServiceKeys	Group	key2	selfServiceKeys	Group
name	key2	Field	name	key2	Field
value	key2	Field	value	key2	Field
key3	selfServiceKeys	Group	key3	selfServiceKeys	Group
name	key3	Field	name	key3	Field
value	key3	Field	value	key3	Field
key4	selfServiceKeys	Group	key4	selfServiceKeys	Group
name	key4	Field	name	key4	Field
value	key4	Field	value	key4	Field
key5	selfServiceKeys	Group	key5	selfServiceKeys	Group
name	key5	Field	name	key5	Field
value	key5	Field	value	key5	Field
emailAddress	head	Field	emailAddress	head	Field
webUserId	head	Field	webUserId	head	Field
ipAddress	head	Field	ipAddress	head	Field
displayMode	head	Field	displayMode	input	Field
previousNext	head	Field	previousNext	input	Field
referenceDateTime	head	Field	referenceDateTime	input	Field
startDate	head	Field	startDate	head	Field
endDate	head	Field	endDate	head	Field
latestMeasurementDateTime	head	Field	latestMeasurementDateTime	head	Field
			customElements	head	Group
mainData	WXMultipleAccount UsagesByServiceType	Group	mainData	WX-MultipleAccount UsagesByServiceType	Group
results	mainData	List	results	mainData	List
accountId	results	Field	externalAccountId	results	Field
serviceType	results	Field	serviceType	results	Field
serviceTypeDescription	results	Field	serviceTypeDescription	results	Field
latestMeasurementDateTime	results	Field	latestMeasurementDateTime	results	Field
usageUom	results	Field	usageUom	results	Field
usageUomDescription	results	Field	usageUomDescription	results	Field
usageSQI	results	Field	usageSQI	results	Field
usageSQIDescription	results	Field	usageSQIDescription	results	Field
periods	results	List	periods	results	List
dateTime	periods	Field	dateTime	periods	Field
quantity	periods	Field	quantity	periods	Field
customElements	periods	Group	customElements	periods	Group
earliestUSStartDateTime	results	Field	earliestUSStartDateTime	results	Field
			customElements	results	Group

skippedAccounts	mainData	List	skippedAccountUS	mainData	List
accounted	skippedAccounts	Field	externalAccountId	skippedAccountUS	Field
saId	skippedAccounts	Field	usId	skippedAccountUS	Field
			usExternalId	skippedAccountUS	Field
saInformation	skippedAccounts	Field	usInfo	skippedAccountUS	Field
skipReasonDescription	skippedAccounts	Field	skipReasonDescription	skippedAccountUS	Field
serviceType	skippedAccounts	Field	serviceType	skippedAccountUS	Field
errorInformation	mainData	Group	errorInformation	mainData	Group
isInError	errorInformation	Field	isInError	errorInformation	Field
errorReference	errorInformation	Group	errorReference	errorInformation	Group
messageCategory	errorReference	Field	messageCategory	errorReference	Field
messageNumber	errorReference	Field	messageNumber	errorReference	Field
errorMessage	errorInformation	Field	errorMessage	errorInformation	Field
			customElements	mainData	Group

Chapter 8

Notifications

The Oracle Utilities Notification Center (OUNC) is pre-integrated with Oracle Utilities Customer SelfService (OUCSS), Oracle Utilities Customer Care and Billing (OUCCB), and Oracle Utilities Network Management System (OUNMS), and facilitates the processing and sending of messages to customers.

Oracle Utilities Customer Care and Billing (CCB) and Network Management System (NMS) provide a mechanism to send messages (or notifications) to customers. The means of delivery are SMS or email, and additional delivery channels can be introduced via extensions. OUNC processes all the notifications sent by the edge applications and sends the notification messages to customers. OUCSS provides a unified "hub" by which all these differing notifications can be managed by the customer. Customers are able to define a Notification profile that captures how they wish to receive the notifications and preferences that capture the type of notifications they want to receive for the account.

With OUCSS v2.2.0 and OUCCB 2.5.0.2, notification delivery channel and preferences can be setup to be stored in either OUNC or CCB.

On the Portal side, this is determined by the configuration option "oucss.notification.owner" (Admin -> Configuration). If the value of this configuration option is set to 'OUNC', it means delivery channels and preferences are stored and retrieved from OUNC .Notifications -> Profile screen is displayed to set up delivery channels and Notification -> Preferences displays a screen which retrieves notification preferences and delivery channels from OUNC tables. If the value of this configuration option is set to 'CCB', it means the notification delivery channels and preferences are stored in CCB. Notifications -> Profile screen is hidden. Person contacts are set up in Details -> Personal Information-> Person Contact screen. Notification -> Preferences displays a screen which retrieves notification preferences and delivery channels from OUCCB

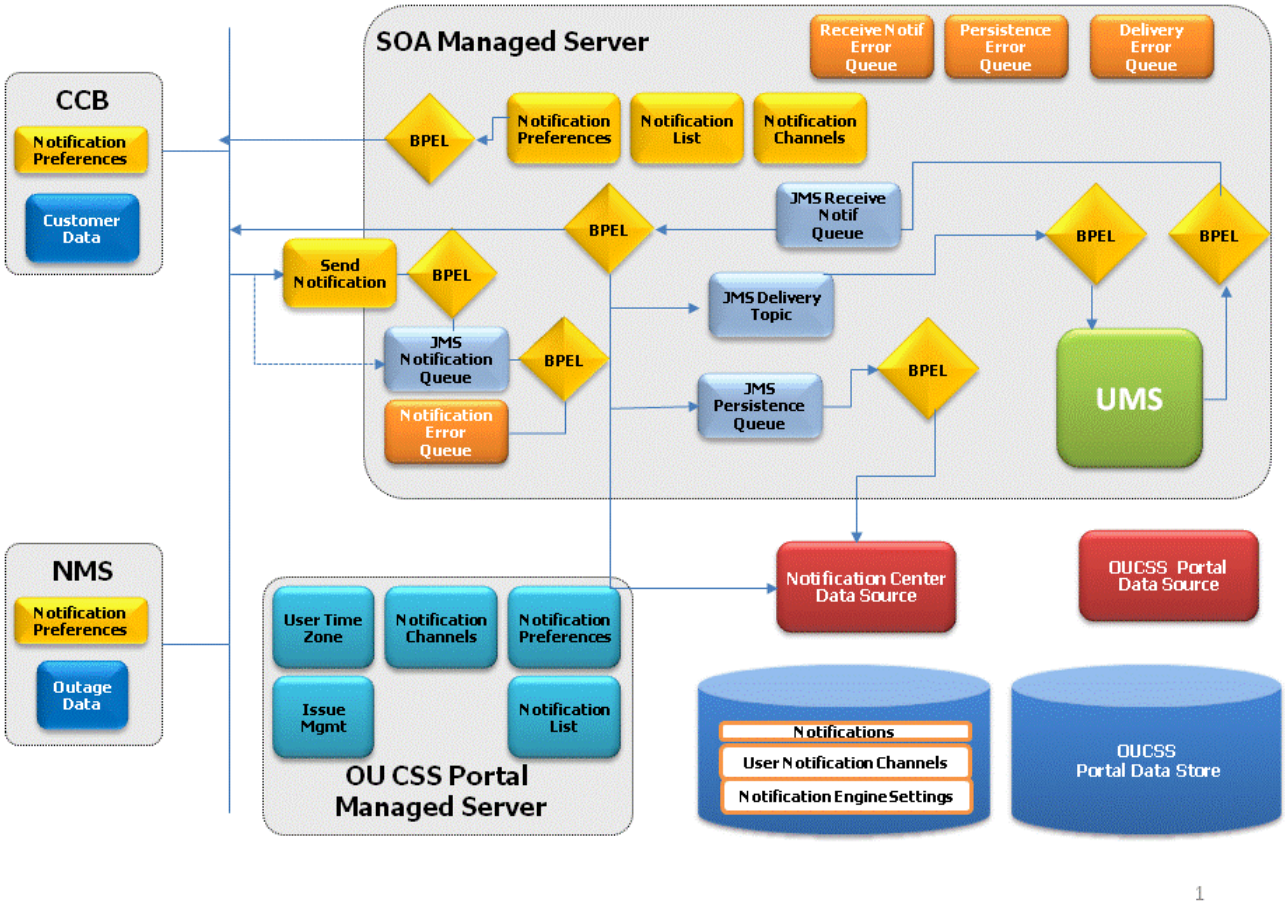
On the integration side, the configuration property "Notification.Owner" set in ConfigurationProperties.xml could be set to either 'OUNC' or 'CCB' to indicate if notifications preferences and delivery channels are stored in OUNC or CCB respectively. BPEL processes use this property to handle the logic.

Note: Implementers must ensure that both oucss.notification.owner on the Portal side and Notification.Owner" on the Integration side must point to the same value when set to either CCB or OUNC.

Note: The original Bill Notification types (prior to OUCCB 2.5.0.2) are also part of Bill Notification taskflow on Details -> Personal Information -> Billing Notification Preferences screen. If Notification Center is installed, the Bill Notification taskflow need not be added to the OUCSS Portal

Notification Center Architecture Model 1

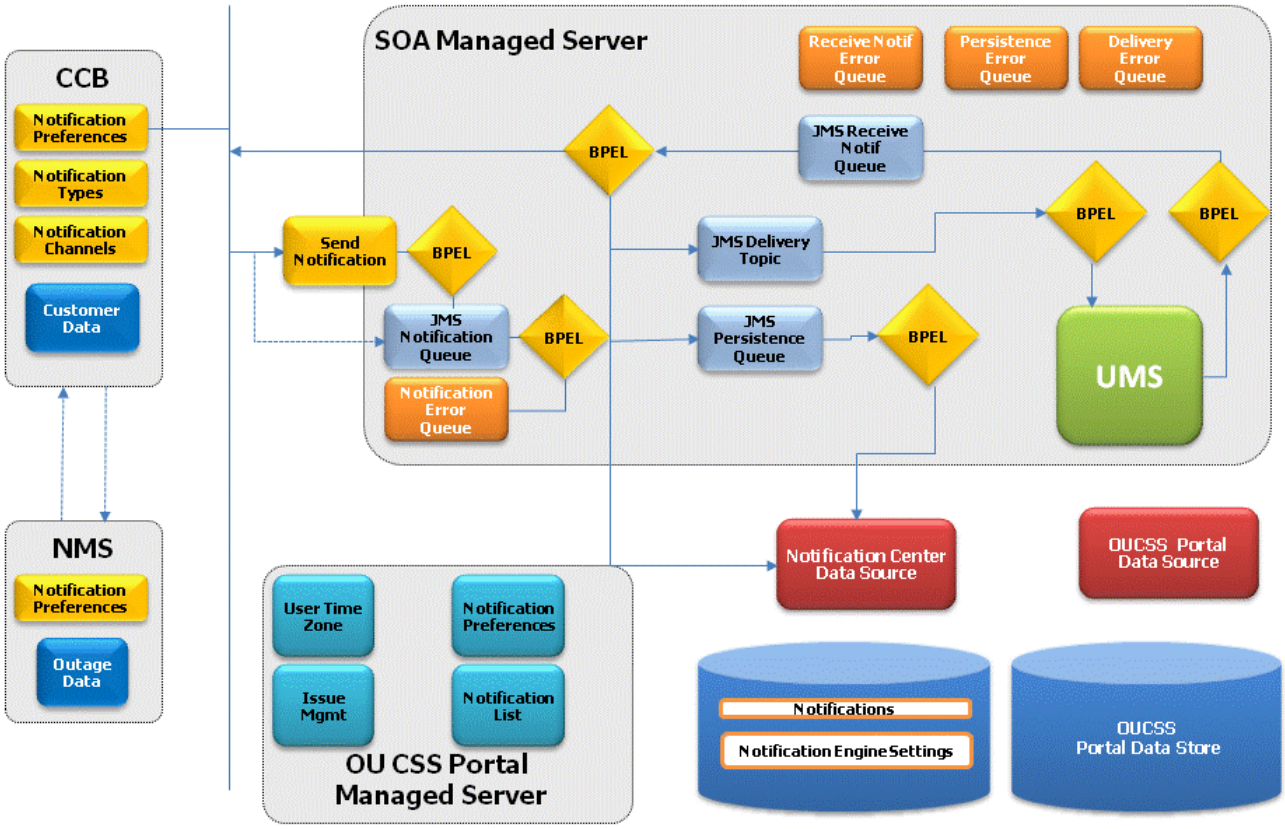
When OUNC owns the notification preferences mapping and notification channels. The integration configuration property Notifications.Owner is set to 'OUNC'.



1

Notification Center Architecture Model 2

When CCB owns the notification preferences and notification channels information. The integration configuration property Notifications.Owner is set to 'CCB'.



1

The Notification Center can be separated into the following parts:

- The Notification Center Engine that is housed in a SOA managed server.
- Notification Center data store. Can be deployed in the same schema as OUCSS portal or in a separate one.
- Notification Management portal pages for Notification Center packaged with OUCSS portal.
- Notification Management Web Services that back the OUCSS Portal screens, packaged with Notification Center.

Notification Management

Opt-in

The Notification Center provides support for either Single Opt-in, Double Opt-in, or No Opt-in features when integration configuration property Notifications.owner is set to 'OUNC' only. When Notifications.Owner = 'CCB' the Opt-in lifecycle for person contacts is managed in CCB. For details, see the *Oracle Utilities Customer Care and Billing Administrative User Guide* and the *Oracle Utilities Customer Care and Billing Business User Guide* (both are available for download from the Oracle Technology Network at http://docs.oracle.com/cd/E75711_01/index.htm).

Single Opt-in

- The user navigates to the Notifications Profile screen and enters in new phone number(s) for SMS messaging.
- The user receive s a confirmation text for each new phone number added as a channel for notifications.
- If the user replies STOP (or equivalent) the phone number will be removed as a valid notification channel as well as all notifications set up using that channel. The customer will receive the configured stop message as SMS text.

  Example:

  ```
  This message confirms you have discontinued YourCompany Alerts and you will no longer
  receive messages. For help call 1-999-999-9999
  ```

- If the user replies HELP (or equivalent) they will receive configured help information texted back to them:

  Example:

  ```
  YourCompany Alerts # msgs varies/user. For info go to <URL> or call 1-999-999-9999 Reply
  STOP to cancel. Msg&Data rates may apply.
  ```

Double Opt-in

- The user navigates to the Notifications Profile screen and enters a new phone number (or numbers) for SMS messaging.
- The user will receive a confirmation text for each new phone number added as a channel for notifications.
- The phone number will not be available as a valid notification channel in the Notification Preferences screen until the user texts back the “CONFIRM”(or equivalent) message.
- After the numbers are validated, the user will receive a configured confirmation text message.

  Example:

  ```
  Thanks for subscribing to YourCompany Alerts. Msg&Data rates may apply. # msgs
  varies/user. Reply HELP for help, Reply STOP to cancel. 1-999-999-9999
  ```

- If the user replies STOP, the phone number should be removed as a valid notification channel as well as all notifications set up using that channel. The customer will receive the configured stop information via text.

  Example:

  ```
  This message confirms you have discontinued YourCompany Alerts and you will no longer
  receive messages. For help call 1-999-999-9999
  ```

- If the user replies HELP, the user should receive help information texted back.

  Example:

  ```
  YourCompany Alerts # msgs varies/user. For info go to <URL> or call 1-999-999-9999 Reply
  STOP to cancel. Msg&Data rates may apply.
  ```

No Opt-in

- The user navigates to the Notification Profile screen and enters in a new phone number (or numbers) for SMS messaging.
- No CONFIRM, HELP, STOP, or equivalent messages are sent back to the user.
- Phone numbers entered are available as a valid delivery channel in the Notification Preferences screen.

Opt In Notification Types

A new **APP_ID** value called **AUDIT_SMS** is introduced to represent the Opt-in feature.

The following notification types are introduced to represent the Opt-in messages:

- **SEND_NEW**: The message corresponding to this notification type is sent when a new SMS number (not used by any user) is added in the case of a Single Opt-in, and a new and non-verified SMS number (not used by any user) in the case of Double Opt-in.
- **SEND_REPEAT**: The message corresponding to this notification type is sent when a new SMS number (already used by another user) is added in the case of a Single Opt-in, and new or verified SMS number (already used by another user) in the case of Double Opt-in.
- **CONFIRM**: The message corresponding to this notification type is sent when the user replies with CONFIRM or an equivalent keyword.
- **HELP**: The message corresponding to this notification type is sent when the user replies with HELP or an equivalent keyword.
- **STOP**: The message corresponding to this notification type is sent when the user replies with STOP or an equivalent keyword.
- **STOPALL**: This notification type is **not** used when Notification Center manages the Optin lifecycle of the contacts. It is used only when CCB manages the notification preferences and the Optin lifecycle of the person contacts.– when Notifications.owner in configuration properties is set to 'CCB'

Portal Pages

The following notification portal pages are available for both Residential and Commercial users.

Portal Page	Description
Profile	The customer defines a Notification Profile through the OUCSS interface. This interface allows the customer to set up Delivery Channels such as SMS numbers, email addresses, etc where notifications can be delivered. This screen is not relevant when the notification preferences along with notification channel details are coming from CCB.This is indicated by the portal configuration parameter oucss.notifications.owner = 'CCB' set on Portal Admin Configuration screen. Thus, this link is hidden when oucss.notifications.owner = 'CCB'. When oucss.notification.owner = 'CCB', a CSS user can set up contacts through the Details>Personal Information->Person Contact screen.
Preferences	OUCSS captures the Notification Preferences for supported notification types. This interface allows customers to choose the types of notification they want to receive and how they want to receive those notifications. When portal configuration parameter oucss.notifications.owner = 'OUNC' ,the list of supported notification types for each edge application is retrieved dynamically for CCB and from the OUNC database for NMS. The Edge Applications may require additional information for a notification type. When portal configuration parameter oucss.notifications.owner = 'CCB' ,the list of supported notification types for each edge application , notification channels and preferences is retrieved dynamically from CCB.
Inbox	This service retrieves all notifications sent out from the edge applications for a given account. Optin messages are filtered

Notification Management OUCSS Integration Services

The following are the base services invoked by Oracle Utilities Customer Self Service to retrieve content of various portal pages.

Notes:

- For more information on configuring CCB services and configurations related to Notifications, see Chapter 3, Customer Care and Billing Configuration, and the Oracle Utilities Customer Care and Billing user documentation.
- For more information on configuring BPEL services, see Chapter 7, CSS Direct BPEL Flows.

OUCSS Module Name	Service Description	CCB Service	BPEL Service	Notes
Profile	Responsible for retrieving delivery channels information for the self-service user.		OUNCWXDeliveryChannels	
Preferences	This service is used to retrieve preferences for a given account to a self-service user. This module calls different service based on who the owner of the notifications is			
	When portal configuration property oucss.notifications.owner = OUNC, the notification types for the preferences are retrieved dynamically for CCB and from the OUNC database for NMS	WXSetNotificationPreferences	OUNCWXNotificationPreferences	
	When portal configuration property oucss.notification.owner=CCB, the notification types, notification preferences and notification channels are retrieved from CCB	WXMaintainCommPreferences	OUCSSOUCCBWXCommunicationPreference (Optional BPEL service)	
Inbox	This service retrieves all notifications sent out from the edge applications for a given account. These notifications, which are previously sent out from the edge applications, are retrieved from the OUNC database. Optin messages are filtered		OUNCWXGetNotifications	

Profile Notification Integration Flow

Business Details

This synchronous BPEL process is responsible for retrieving notification delivery channels information for the self-service user.

Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

1. Check if the request action is a Read or an Update of the Delivery Channels of the specific user.
2. If Request message Action is Read, go to Step 4.
3. If Request message Action is Update, for each user delivery channel:
   - Check if Delivery Channel needs to be Deleted, if yes
     - If notification preference was created for this delivery channel, invoke GetNotifPrefForUser DBAdapter to get notification preferences for this user
     - Use the XForm_PreferenceData.xsl to transform the message to a format grouping notification preferences per user per account
     - For each of this record, transform it to request format of OUNCWXNotificationPreference on 'READ'
     - Invoke OUNCWXNotificationPreference with 'READ' action
     - Transform the response on Read to the request format of OUNCWXNotificationPreference 'UPDATE' action, marking the 'deleted' element as 'Y' for notification preferences for this delivery channel.
     - Call OUNCWXNotificationPreference with 'UPDATE' action.
   - Invoke the DB Adapter service OUNCDeleteUserChannelDBAdapter to delete the delivery channel.
     - If the delivery channel needs to be Updated, then check if the delivery type is SMS. If it is SMS and if the optIn type is SINGLE then,
     - If this SMS number was not used before, assign 'SEND_NEW' to notification type
     - If this SMS number was used before by any user, assign 'SEND_REPEAT' to notificationtype
     - If the delivery type is SMS and optIn type is DOUBLE and If this SMS number was not used before, assign 'SEND_NEW' to notificationtype
     - If this SMS number was used before by any user and was verified, assign 'SEND_REPEAT' to notificationtype else assign 'SEND_NEW' to notificationtype
     - Transform the message to the request format of OUNCSendOptInMessage BPEL process
     - Invoke OUNCSendOptInMessage BPEL process to send out appropriate Opt-in message based on the notification type
   - Invoke the DB Adapter service OUNCUpdateChannelDBAdapter to update/save the user channnel.
4. Get User Delivery Channels using the DB Adapter Service OUNCUserChannelsDBAdapter.
5. Get Channel Types using the DB Adapter Service OUNCChannelTypesDBAdapter.
6. Transform the message to a format that needs to be returned to CSS as response to the message.
7. Send the response message.

- **Error Handling**

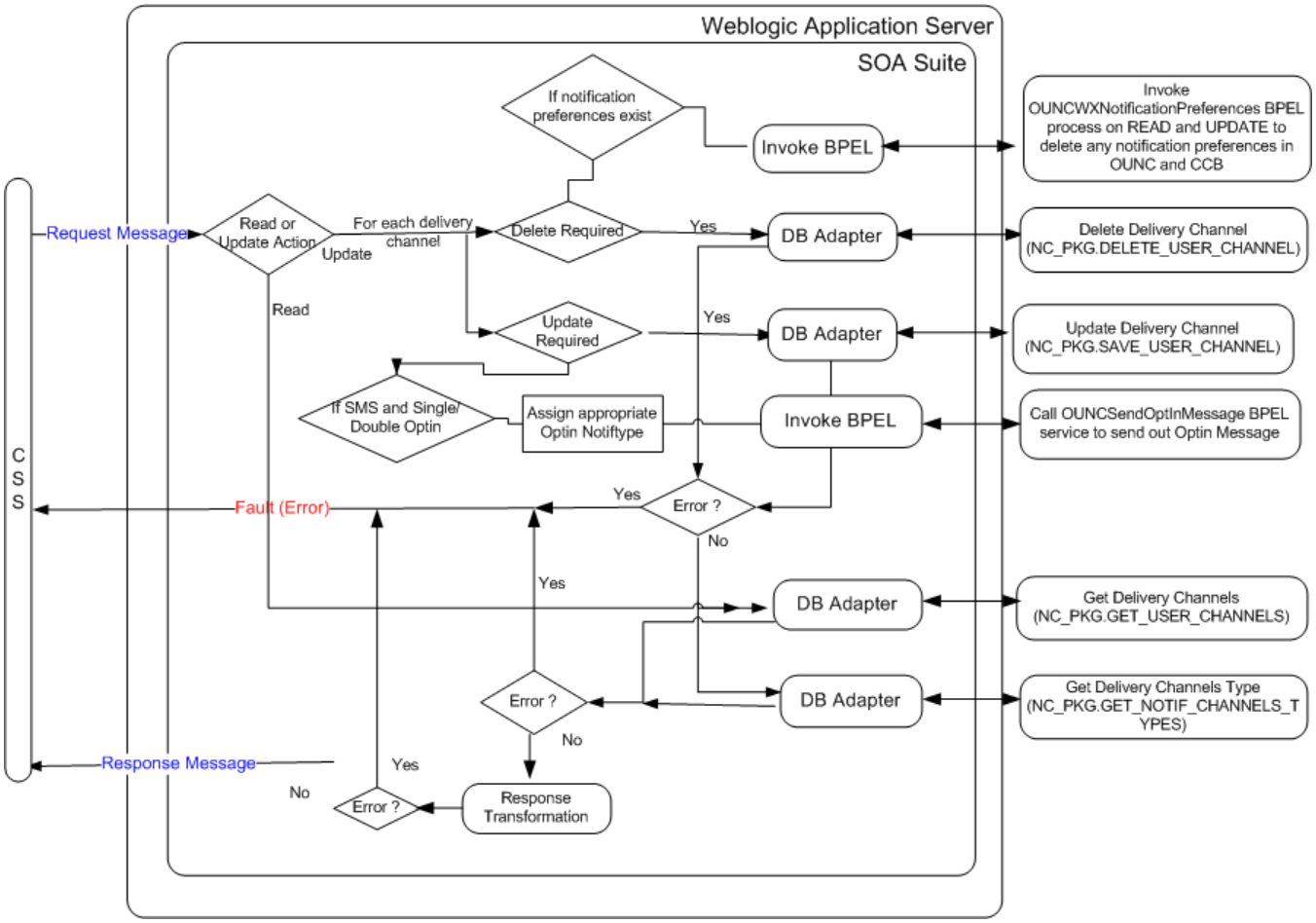
When a business or technical error is encountered in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.

- **Customization**
  - The PreInvokeNCWXDeliveryChannelsRequest extension scope is invoked after the request message is received, and the PreInvokeGetUserChannels extension scope is invoked before the get user channels adapter service is invoked.
  - PreInvokeGetChannelTypes extension scope is invoked before the get channel types adapter service is invoked.
  - PreInvokeNCWXDeliveryChannelsResponse extension scope is invoked before the response.
  - ExtScope_PreInvoke_SendOptIn extension scope is invoked before OUNCSendOptInMessage BPEL process is called

- ExtScope_PostInvoke_SendOptIn extension scope is invoked after OUNCSendOptInMessage BPEL process is called
- This extension scope will help implementers change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCWXDeliveryChannels	This service is responsible for retrieving delivery channels information for the self-service user.

DB Adapter Services

Name	Description	Operation
OUNCDeleteUserChannelDBAdapter	This adapter service invokes the plsql package to delete the delivery channel.	NC_PKG.DELETE_USER_CHANNEL
OUNCUpdateChannelDBAdapter	This adapter service invokes the plsql package to update the delivery channel.	NC_PKG.SAVE_USER_CHANNEL

OUNCUserChannelsDBAdapter	This adapter service invokes the plsql package to get the user delivery channels	NC_PKG.GET_USER_CHANNELS
OUNCChannelTypesDBAdapter	This adapter service invokes the plsql package to get the channel types	NC_PKG.GET_NOTIF_CHANNELS_TYPES

Preferences Notification Integration Flow

Business Details

This synchronous BPEL process is responsible for retrieving preferences information for the given account to a self-service user when Notification Center(OUNC) is the notifications owners. i.e **when Notifications.Owner = OUNC** in ConfigurationProperties.

The notification types for the preferences are retrieved dynamically for CCB and from the OUNC database for NMS.

Note: When Notifications.Owner = CCB, the Notifications ->Preferences portal screen which displays a Communication Preference taskflow which invokes WXCommunicationPreference CCB flow directly or through Optional BPEL flow OUCSSOUCCBWXMaintainCommPreferencesEBF

Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

1. Check if the request action is a Read or an Update of the Delivery Channels of the specific user.
2. If the Request message Action is Read:
   - **F** Assign the CCB parameters required to invoke the CCB web service to retrieve CCB notification types, notification options, and available delivery types.
   - **G** Invoke the CCB inbound web service WXSetNotificationPreferences with Read Action.
   - **H** Go to Step 4.
3. If Request message Action is Update, for each user notification preference type:
   - **I** Check if User Preference needs to be deleted, if yes invoke the DB Adapter service OUNCUserPrefDeleteDBAdapter to delete the user account preference.
   - **J** Invoke the DB Adapter service OUNCUserPrefSaveDBAdapter to update user preferences
   - **K** Check if NMS Enabled flag from Configuration Properties is true. If True:
     - **i** Get available NMS Notification types using the DB adapter OUNCGetAvailNotifTypesDBAdapter.
     - **ii** Delete NMS notification preferences if required using the DB Adapter OUNCDelNMSPreferenceDBAdapter.
     - **iii** Update NMS Notification preferences using the DB Adapter OUNCAddNMSPreferenceDBAdapter.
   - **L** Check if the CCB Enabled flag from Configuration Properties is true. If True:
     - **iv** Transform the request message to invoke the CCB web service.
     - **v** Invoke the CCB inbound web service WXSetNotificationPreferences with Update Action.
4. Get Channel Types using the DB Adapter Service OUNCChannelTypesDBAdapter.
5. Get NMS Notification types descriptions using the DB adapter OUNCGetNotifTypesDBAdapter, passing in the user's locale.
6. Get Notification Preferences from the database using the DB Adapter Service OUNCGetNotifLocalPrefsDBAdapter.
7. Transform the message to a format that needs to be returned to CSS as a response to the message using the outputs from step 2 *or* 3, and steps 4, 5, and 6 as inputs to the transformation.

8 Send the response message back to CSS.

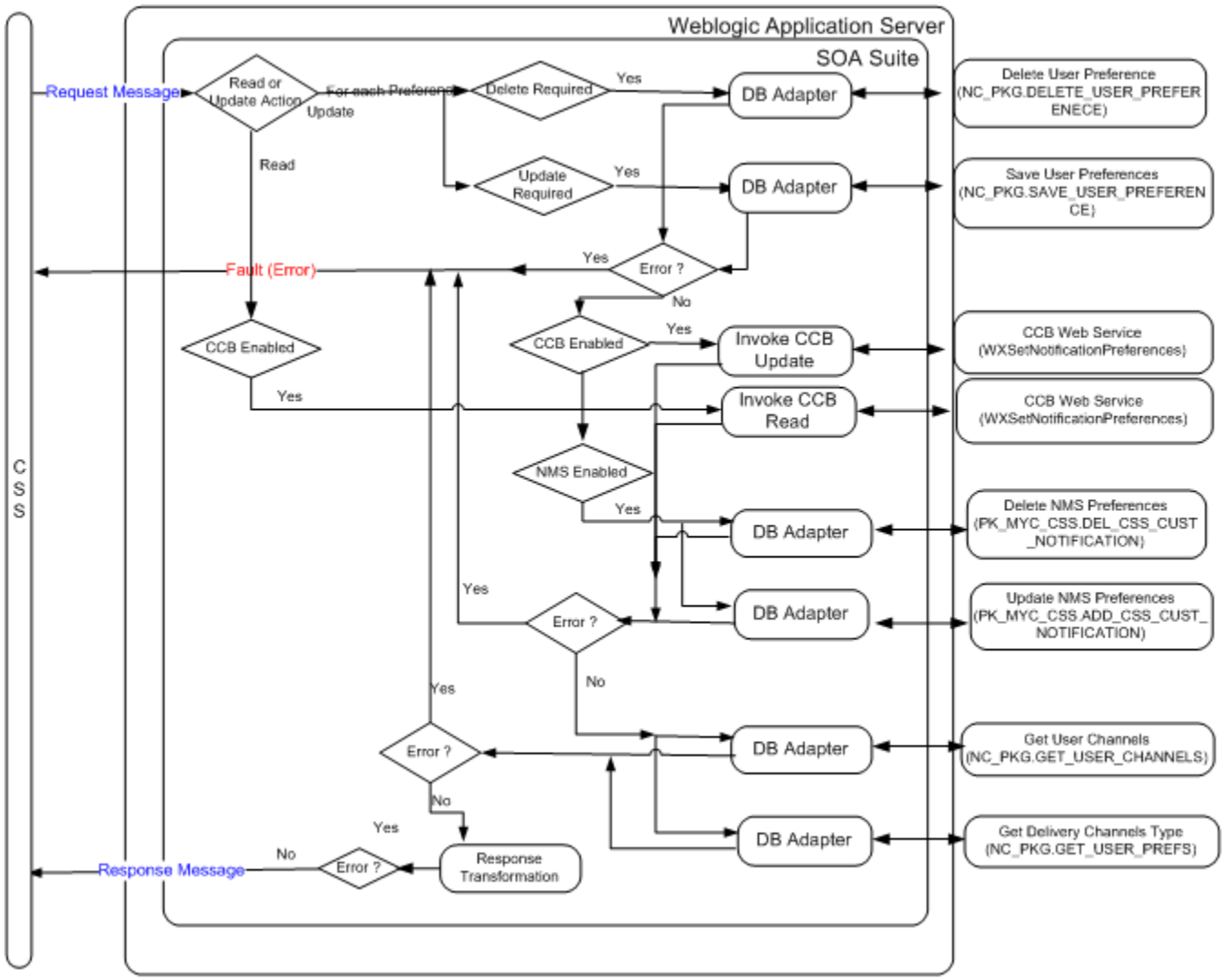
- **Error Handling**

When a business or technical error is encountered in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.

- **Customization**
  - The **PreInvokeNCWXGetNotificationPrefsRequest** extension scope is invoked after the request message is received and the PreInvokeGetAvailNotifTypes extension scope is invoked before the NMS adapter services are invoked.
  - The **PreInvokeCallCCBSetNotifPreferences** extension scope is invoked before the CCB web service is invoked.
  - The **PreInvokeGetUserChannels** extension scope is invoked before getting the user channels.
  - The **PreInvokeGetNotifTypes** extension scope is invoked before getting the notification type descriptions.
  - The **PreInvokeGetNotifPrefs** extension scope is invoked before getting the notification preferences.
  - The **PreInvokeNCWXGetNotificationPrefsReply** extension scope is invoked before sending the response back to CSS after the response transformation.
  - This extension scope will help implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCWXNotificationPreferences	This service is used to retrieve preferences for a given account to a self-service user. The CCB notification types used in the notification preference page are retrieved dynamically from CCB by calling the CCB Set Preference Web Service. The NMS notification types used in the notification preferences page are retrieved from the OUNC database. Note: Previously, NMS notification types were retrieved from the configuration properties file. If you apply Patch 19022578, notification types will be retrieved from the OUNC database and *not* from the Configuration Properties File.

DB Adapter Services

Name	Description	Operation
OUNCUserPrefDeleteDBAdapter	This adapter service invokes the plsql package to delete the notification preference for the user account.	NC_PKG.DELETE_USER_PREFERENECE
OUNCUserPrefSaveDBAdapter	This adapter service invokes the plsql package to update the delivery channel.	NC_PKG.SAVE_USER_PREFERENCE
OUNCGetAvailNotifTypesDBAdapter	This adapter service invokes the plsql package to get the available notification types	NC_PKG.GET_AVAIL_NOTIF_TYPES_APP
OUNCDelNMSPreferenceDBAdapter	This adapter service invokes NMS database to delete notification preferences	PK_MYC_CSS.DEL_CSS_CUST_NOTIFICATION
OUNCAddNMSPreferenceDBAdapter	This adapter service invokes NMS database to update notification preferences	PK_MYC_CSS.ADD_CSS_CUST_NOTIFICATION
OUNCUserChannelsDBAdapter	This adapter service invokes the plsql package to get the user delivery channels	NC_PKG.GET_USER_CHANNELS
OUNCGetNotifLocalPrefsDBAdapter	This adapter service invokes the plsql package to get the notification preferences for user.	NC_PKG.GET_USER_PREFS
OUNCGetNotifTypesDBAdapter	This adapter service invokes the plsql package to get the notification types.	NC_PKG.GET_NOTIF_TYPES

Web Services

Name	Description
WXSetNotificationPreferences	This web service is used to invoke

Inbox Notification Integration Flow

Business Details

This synchronous BPEL process is responsible for retrieving existing notifications for the self-service user account.

Technical Details

When the integration BPEL service receives a request from CSS, it will do the following:

1. Check if the request action is a Delete or a Resend of the Notifications of the specific user.
2. If Request message Action is Delete notification invoke delete DB Adapter service OUNCDeleteNotifDBAdapter
3. If Request message Action is Resend notification invoke resend DB Adapter service OUNCResendNotifDBAdapter passing also the Notifications owner configuration property.
4. Get User Notifications List using the DB Adapter Service OUNCNotifListDBAdapter passing also the Notifications owner configuration property.

5 Get Notification Channels using the DB Adapter Service OUNCGetNotifChannelsDBAdapter. There will not be any notification channels returned when Notifications owner configuration property is set to 'CCB' since there is no notification channels stored in OUNC datasource

6 Transform the message to a format that needs to be returned to CSS as response to the message.

7 Send response message.

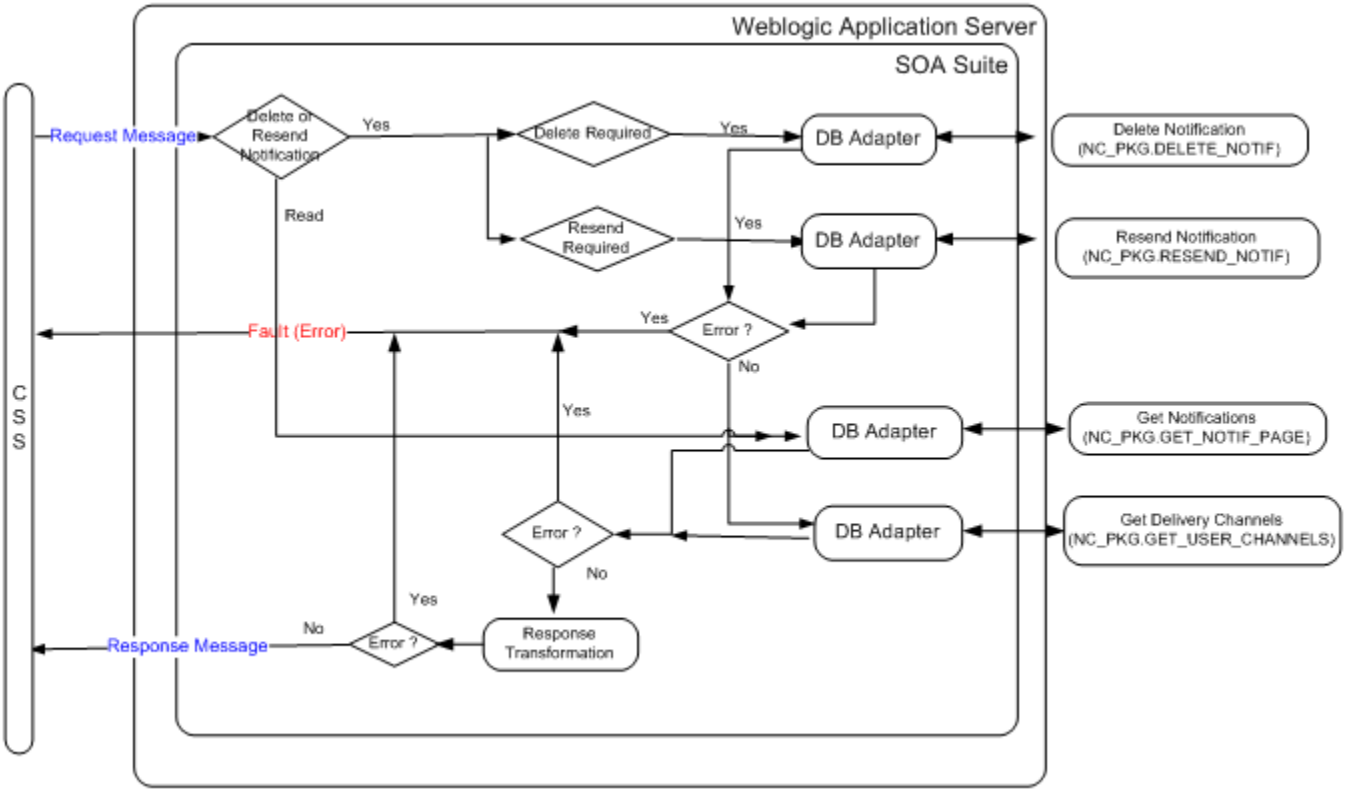
- **Error Handling**

When a business or technical error is encountered in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.

- **Customization**
  - The PreInvokeNCWXGetNotificationRequest extension scope is invoked after the request message is received and
  - PreInvokeDeleteNotification extension scope is invoked before the delete notification adapter service is invoked.
  - PreInvokeResendNotification extension scope is invoked before the resend notification adapter service is invoked.
  - PreInvokeGetNotificationDBService extension scope is invoked before the get notification db adapter service.
  - PreInvokeGetNotifChannels extension scope is invoked before the get notification channels db adapter service.
  - PreInvokeNCWXGetNotificationReply extension scope is invoked before the response is sent back to CSS.
  - This extension scope will help implementers change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



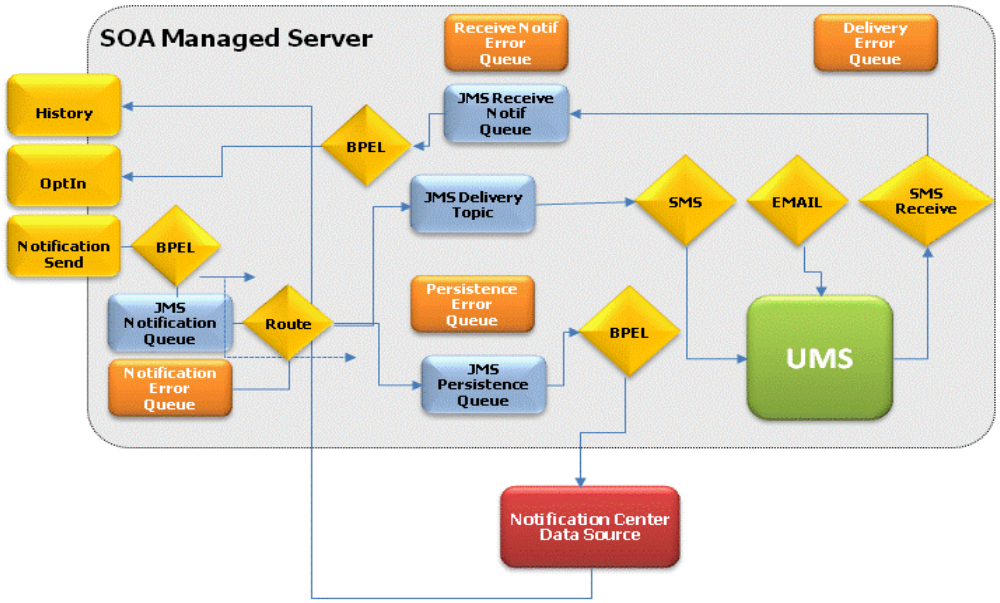
Integration Services

Name	Description
OUNCWXGetNotifications	This service retrieves all notifications sent out from edge application for a given account. Notifications previously sent from edge applications are retrieved from the OUNC database.

DB Adapter Services

Name	Description	Operation
OUNCDeleteNotifDBAdapter	This adapter service invokes the plsql package to delete a specific notification.	NC_PKG.DELETE_NOTIF
OUNCResendNotifDBAdapter	This adapter service invokes the plsql package to resend a specific notification.	NC_PKG.RESEND_NOTIF
OUNCNotifListDBAdapter	This adapter service invokes the plsql package to get the notification list.	NC_PKG.GET_NOTIF_PAGE
OUNCGetNotifChannelsDBAdapter	This adapter service invokes the plsql package to get the user channels.	NC_PKG.GET_USER_CHANNELS

Notification Center Engine



The Notification Center Engine has the following structure:

- The interface for notification submission. The edge applications can post a notification either via a Web Service (eg. CCB) or write a message directly to a JMS queue. The Web service allows third party notification providers to send notifications. BPEL processes OUNCNotificationInbound , CCBOUNCNotificationInbound handles inbound notification from web service and OUNCNMSNotificationInbound polls NMS database for available notifications. Both the BPEL processes publishes the message in the JMS Notification queue (NotifInboundQueue).
- BPEL Process *OUNCRouteNotification* reads a message from the Notification queue and fetches Notification preferences for the specific notification type and account from the the Notification Center's database schema.
  - If the message cannot be processed, it goes to the Notification Error Queue.
  - If the preferences were processed successfully, the process publishes messages to the following JMS destinations:
    - A JMS Topic (NotifDeliveryTopic) that contains individual messages (e.g., -- SMS, number, text) that need to be processed and sent out to the external system
    - A JMS queue (NotifPersistenceQueue) that contains messages that need to be persisted. Also, it contains Do Not Disturb (DND) messages; e.g., if the message can't be delivered now it will be stored with DND settings and will later be picked up by OUNCDeliverDeferred BPEL process.
- *OUNCSmppConnector* and *OUNCEmailConnector* consume messages (durable subscribers) from *NotifDeliveryTopic* and relay those messages to Universal Messaging Server (UMS).
- *OUNCPersistNotification* BPEL process persists all notification messages to the Notification Center database (both regular messages as well as the deferred messages)
- *OUNCDeliverDeferred* BPEL process polls the database table for deferred messages that can be delivered and puts them in *NotifDeliveryTopic*.
- *OUNCSmppConnector* and *OUNCEmailConnector* consume messages (durable subscribers) from *NotifDeliveryTopic* and relay those messages to Universal Messaging Server (UMS).
- *OUNCPersistNotification* BPEL process persists all notification messages to the Notification Center database (both regular messages as well as the deferred messages)
- *OUNCDeliverDeferred* BPEL process polls the database table for deferred messages that can be delivered and puts them in *NotifDeliveryTopic*.
- *OUNCReceiveAuditMessage* BPEL process listens to SMS messages (Optin Responses) coming in to UMS and moves the message to NotifReceiveQueue
- *OUNCProcessAuditMessage BPEL* process consumes messages from NotifReceiveQueue and processes the SMS response of the user as needed based on the Notifications owner.
- *OUNCSendOptInMessage* BPEL process retreives the Optin message for the corresponding Optin notif ication type and pushes the message into NotifPersistenceQueue and NotifDeliveryTopic to be persisted in the OUNC tables and delivered to the delivery channel
- *CCBOUNCNotificationInbound BPEL* process handles inbound notification from web service.
- *CCBOUNCNotificationHistory BPEL* delivers the history records from NC_NOTIFICATION table to the calling web service (CCB)

Note: When CCB owns the notification preferences , notification channels and contact information, CCB is the single source communicating with Notification center for all the notifications, preferences and channels. CCB integrates through BPEL with NMS to send notification preferences and NMS sends the notifications to CCB based on the preferences set. These flows related to CCB-NMS integration on Notifications are part of CCB-NMS Integration Patch 23333992 which must be installed for this functionality. For details see the Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Network Management System Implementation Guide.

Notification Engine Integration Services

The following are the BPEL processes used by the Notification Engine to send Notifications out:

Name	Description
OUNCNMSNotificationInbound	NMS Inbound Notification BPEL Process BPEL process that polls the NMS database for available notifications and publishes the message in the JMS Notification queue (NotifInboundQueue).
OUNCNotificationInbound	Inbound Notification BPEL Process for CCB or any external system that sends notifications by invoking a web service. This BPEL process reads the input message and converts it into generic format and publishes the message in the JMS Notification queue (NotificationInboundQueue).
OUNCRouteNotification	This BPEL Process reads a message from the Notification queue and fetches Notification preferences for the specific notification type and account from the the Notification Center's database schema. The process publishes messages to the following JMS destinations: JMS Topic (NotifDeliveryTopic) that contains individual messages (eg. -- SMS, number, text) that need to be processed and sent out to the external system and JMS queue (NotifPersistenceQueue) that contains messages that need to be persisted. Also, it contains Do Not Disturb (DND) messages
OUNCPersistNotification	This BPEL process persists all notification messages to the Notification Center database (both regular messages as well as the deferred messages)
OUNCSmppConnector	This BPEL Process consumes SMS messages (durable subscribers) from *NotifDeliveryTopic* and relays those messages to Universal Messaging Server (UMS)
OUNCEmailConnector	This BPEL Process consumes EMAIL messages (durable subscribers) from *NotifDeliveryTopic* and relays those messages to Universal Messaging Server (UMS)
OUNCDeliverDeferred	This BPEL process polls the database table for any deferred messages that can be delivered and puts them in NotifDeliveryTopic.
OUNCReceiveAuditMessage	• This BPEL process listens to SMS messages coming in to UMS and moves the message to NotifReceiveQueue
OUNCProcessAuditMessage	• This BPEL process consumes messages from NotifReceiveQueue and processes the SMS response of the user.
OUNCSendOptInMessage	• This BPEL process retreives the Optin message for the corresponding Optin notif type and pushes the message into NotifPersistenceQueue and NotifDeliveryTopic to be persisted in the OUNC tables and delivered to the delivery channel
CCBOUNCNotificationInbound	• This BPEL process handles inbound notification from web service.This is configured and called from CCB when Notifications.Owner configuration property is set to CCB
CCBOUNCNotificationHistory	• This BPEL process delivers the history records from

NC_NOTIFICATION table to the calling web service.

Get Inbound Notification Integration Flow

Business Details

This process is used to receive an inbound notification via a web service call

CCB or any external system will send a notification message via a web service call to the integration BPEL service. The BPEL service publishes the message in the JMS Notification queue.

Note: There are two separate BPEL process to receive and inbound notification depending on if OUNC or CCB owns the preferences and notification channels.

OUNCNotificationInbound integrates with C1-NotifPrefOutboundMessage from CCB. These messages are generated when OUNC owns the preferences and notification channels.

CCBOUNCNotificationInbound integrates with C1-NotificationMessage from CCB. These messages are generated when CCB owns the preferences and notification channels.

Technical Details

When the integration BPEL service receives a request from CCB or any external system, it will do the following:

- Message Transformation

Transform the CCB notification message to a generic format.

- Split Message if required

CCB may send multiple notification messages meant for different delivery channels as part of the same request message. The BPEL process splits the request message into individual messages (for a given notification type and delivery channel).

- Push Message to Queue

Individual messages are put into the JMS Notification queue (NotifInboundQueue).

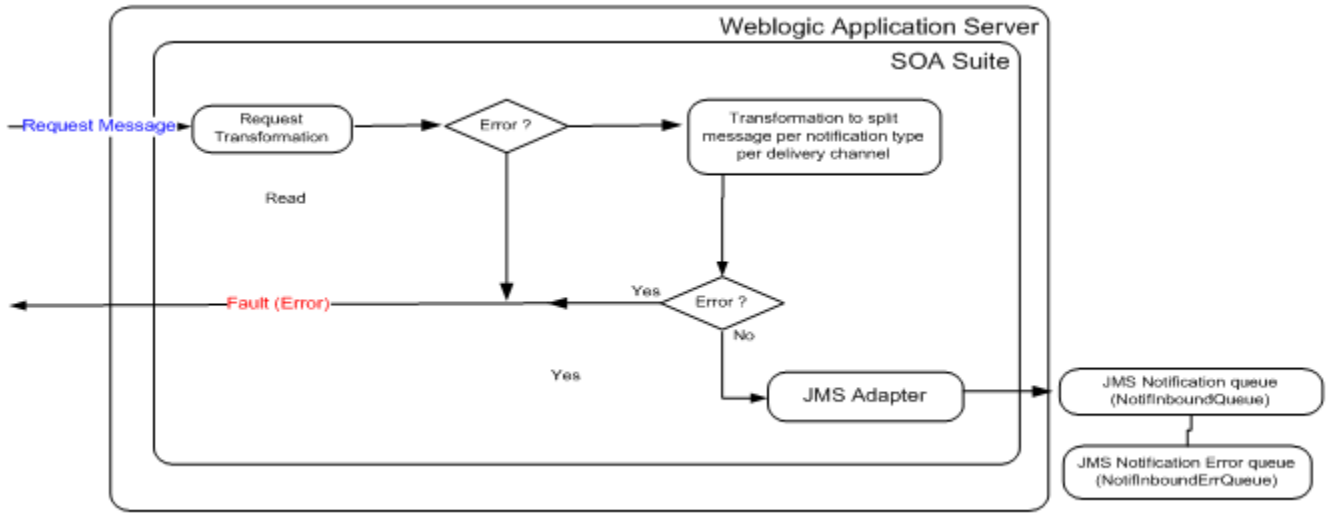
- **Error Handling**

When a business or technical error is encountered in the Integration, a SOAP fault will be returned to CCB with a specific business or technical message code. The message codes are obtained from the configuration properties file.

- **Customization**
  - The pre transformation extension scope is invoked before the main transformation is executed.
  - The pre invoke extension scope is invoked before putting the message into the queue. This will help the implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCNotificationInbound	Inbound Notification BPEL Process for CCB or any external system that sends notifications by invoking a web service when OUNC owns preferences and delivery channels.
CCBOUNCNotificationInbound	Inbound Notification BPEL Process for CCB or any external system that sends notifications by invoking a web service when CCB owns preferences and delivery channels. These BPEL processes reads the input message and converts it into generic format and publishes the message in the JMS Notification queue (NotificationInboundQueue).

JMS Adapter Services

Name	Description
OUNCQueueInboundMessage	This adapter service puts the message into JMS Queue NotifInboundQueue

Get NMS Inbound Notification Integration Flow

Business Details

NMS Inbound Notification BPEL Process polls the NMS database for available notifications and publishes the message in the JMS Notification queue (NotifInboundQueue).

Technical Details

The integration BPEL process does the following:

- Poll NMS Database for Notifications

The input DB Adapter polls the NMS database table/view for available NMS notifications

- Message Transformation

Transform the NMS notification message to a generic format, using notification types retrieved from the OUNC DB.

- Push Message to Queue

Individual messages are put into the JMS Notification queue (NotifInboundQueue).

- **Error Handling**

When a business or technical error is encountered in the Integration, a SOAP fault will be returned with a specific business or technical message code. The message codes are obtained from the configuration properties file.

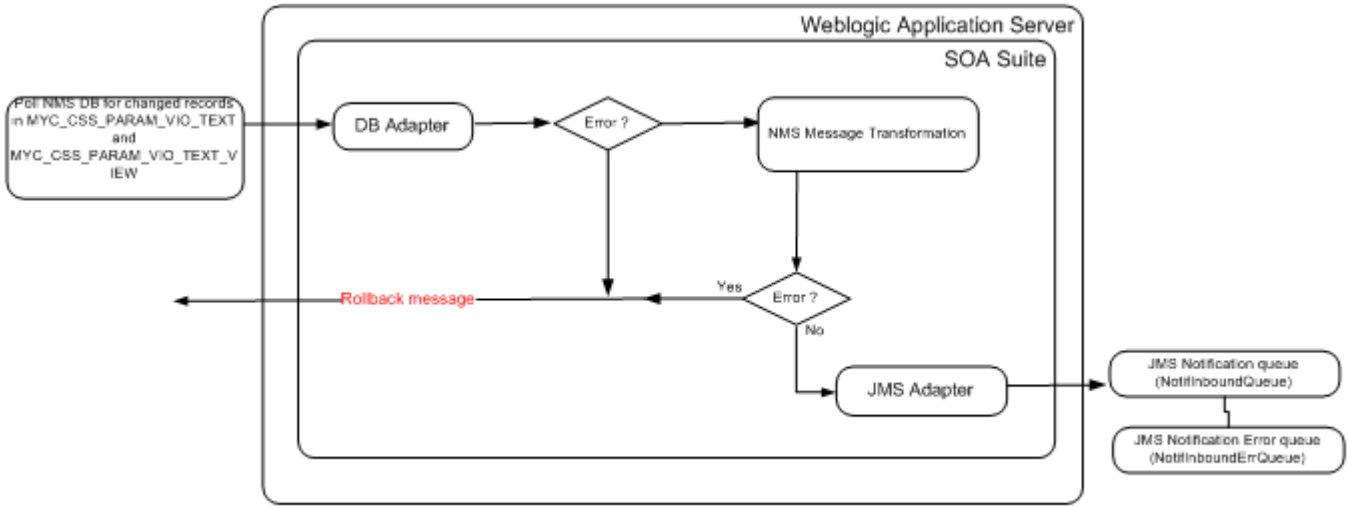
- **Customization**

The pre-transformation extension scope is invoked before the main transformation is executed.

The pre-invoke extension scope is invoked before putting the message into the queue. This will help implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCNMSNotificationInbound	NMS Inbound Notification BPEL Process BPEL process that polls the NMS database for available notifications and publishes the message in the JMS Notification queue (NotifInboundQueue).

DB Adapter Services

Name	Description	Operation
OUNCNMSNotifPollDBAdapter	This adapter service polls the NMS DB table/view for changed records and deletes the records after	Poll NMS DB for changed records in MYC_CSS_PARAM_VIO_TEXT and MYC_CSS_PARAM_VIO_TEXT_VI

	they are read from NMS db.	EW
OUNCGetNotifTypesDBAdapter	This adapter service invokes the plsql package to get the notification types	NC_PKG.GET_NOTIF_TYPES

JMS Adapter Services

Name	Description
OUNCQueueInboundMessage	This adapter service puts the message into JMS Queue NotifInboundQueue

Route Notification Integration Flow

Business Details

This process reads from the inbound Notification queue and fetches Notification route preferences for the specific notification type and account from the the Notification Center's database. It then uses the preferences to route the received notification to a JMS topic for delivery and to a JMS Queue to persist the notification to the Notification Center Database.

Technical Details

Integration BPEL process does the following:

When Notification.owner configuration property is set to OUNC

- Consume message

OUNCRetrieveInboundMessage consumes message from the JMS Notification queue (NotifInboundQueue).

- Get Preferences

A DB Adapter service OUNCGetRoutesDBAdapter retrieves the preferences for a given account and notification type from the Notification Center database by invoking the PLSQL package procedure NC_PKG.GET_ROUTE_PAGE. This returns all the delivery contacts to whom the notification message needs to be delivered. If a Optin.Type configuration property is set to 'DOUBLE', the this PLSQL stored procedure returns SMS delivery contacts only that have been confirmed.

- Relay Message
  - Message Transformation for Relay

    For each preference received for the account and notification type, if not deferred, transform the notification message to a format to be put into the topic.

  - Push Notification Message to Delivery Topic

    After transformation, invoke JMS adapter to push the message into the Notification Delivery Topic (NotifDeliveryTopic)
- Store Message
  - Message Transformation for Persistence Queue

    For each preference received for the account and notification type, transform the notification message to a format to be put into the queue for persistence.

  - Push Notification Message to Persist

    After transformation, invoke JMS adapter to push the message into the Notification Persistence Queue (NotifPersistenceQueue)

When the Notification.owner configuration property is set to CCB:

Note: The assumption is that CCB environment and Middleware environment are in the same time zone for DND time comparison

- If DoNotDeliverFlag is false or empty, the message is not pushed to NotifDeliveryTopic to be relayed out
- If OverrideDoNotDisturb is set to false and if DNDStart and DNDEndTime is present, OUNCCheckDNDDBAdapter is called to check if the current time is DND time. If it is, then the delivery of the message is deferred.If the current time is not DND time, then the message is transformed and pushed to NotifDeliveryTopic to be relayed by invocation of OUNCRelayNotificationToGateway JMS adapter
- The message is transformed to the message format of NotifPersistenceQueue and the message is pushed into NotifPersistenceQueue by invocation of OUNCPersistNotification JMS adapter.

  Note: When Notifications.owner = 'CCB' the columns key3 and key4 are used to store the person contact id and notification preference Id, respectively.

- Error Handling

When a business or technical error is encountered in the Integration, a SOAP fault will be returned with a specific business or technical message code. The message codes are obtained from the configuration properties file.

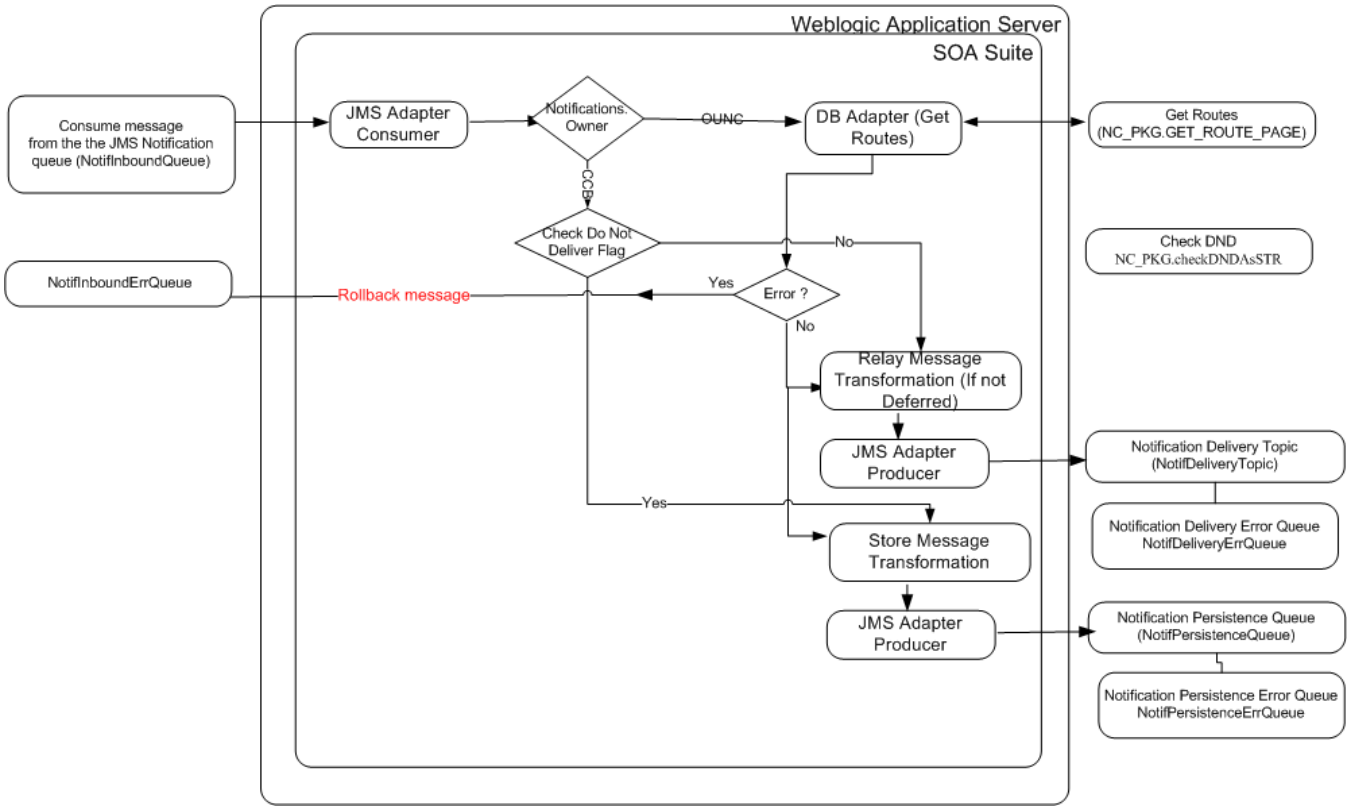
- Customization

The pre transformation extension scope is invoked before the message is routed.

The pre invoke extension scopes are invoked before putting the message into the Delivery Topic as well as the Persistence queue. This will help the implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCRouteNotification	This BPEL Process reads a message from the Notification queue and fetches Notification preferences for the specific notification type and account from the the Notification Center's database. The process publishes messages to the following JMS destinations: JMS Topic (NotifDeliveryTopic) that contains individual messages (eg. -- SMS, number, text) that need to be processed and sent out to the external system and JMS queue (NotifPersistenceQueue) that contains messages that need to be persisted. Also, it contains Do Not Disturb (DND) messages

DB Adapter Services

Name	Description	Operation
OUNCGetRoutesDBAdapter	This adapter service invokes the plsql package to get the route preferences	NC_PKG.GET_ROUTE_PAGE
OUNCCheckDNDDBAdapter	This adapter service invokes plsql stored procedure to check if the current time is in the DND time period. The current time and DND times are converted to UTC timezone and compared	NC_PKG.checkDNDAsSTR

JMS Adapter Services

Name	Description
OUNCRetrieveInboundMessage	This adapter service consumes the message from the JMS Queue NotifInboundQueue
OUNCRelayNotificationToGateway	This adapter service puts the notification message to be relayed to the topic Notification Delivery Topic
OUNCPersistNotification	This adapter service puts the notification message into the persistence queue NotifPersistenceQueue

Persist Notification Integration Flow

Business Details

This BPEL process stores the notification messages to the Notification Center database, both regular messages as well as the deferred messages.

Technical Details

Integration BPEL process does the following:

- Consume message

OUNCNotifPersistenceJMSConsumer consumes message from the JMS Notification queue (NotifPersistenceQueue).

- Message Transformation for Store

Transform the notification message to a format to be put into DB for persistence.

- Persist message

Persist message by invoking the DB Adapter OUNCNotificationPersistDBAdapter.

- **Error Handling**

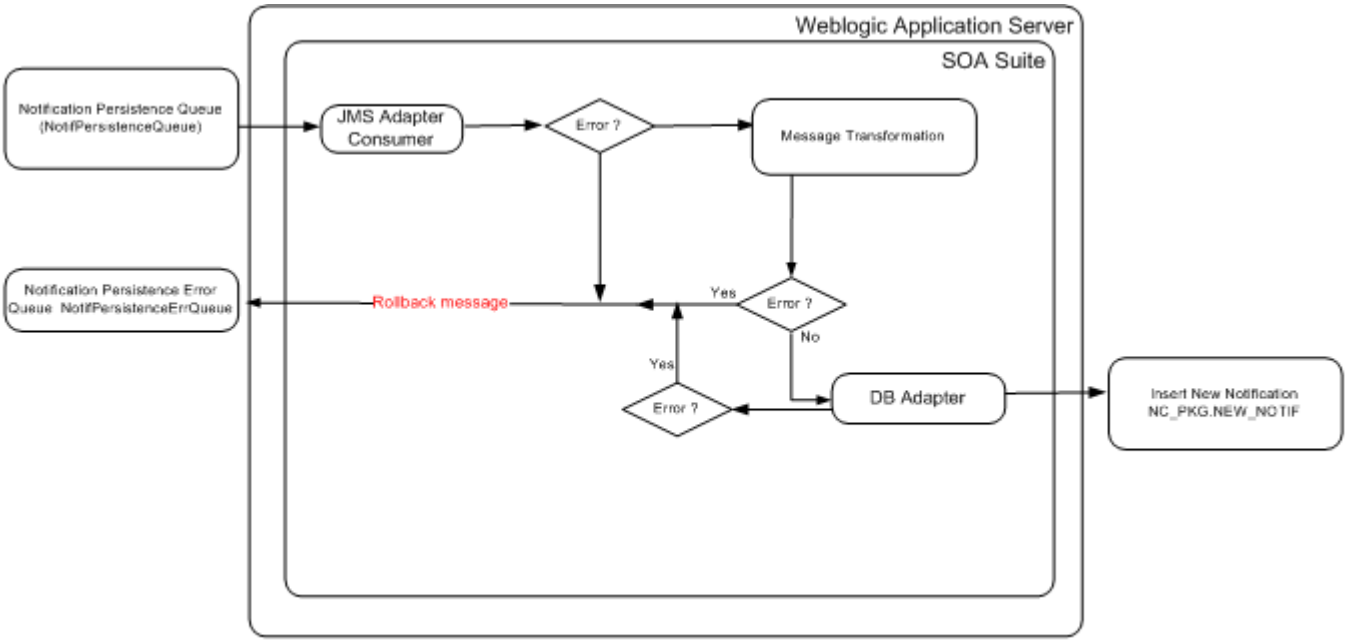
When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **Customization**

The pre invoke extension scopes is invoked before putting the message into the Database.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCPersistNotification	This BPEL process persists all notification messages to the Notification Center database (both regular messages as well as the deferred messages)

DB Adapter Services

Name	Description	Operation
OUNCNotificationPersistDBAdapter	This adapter service invokes the plsql package to persist the notification.	NC_PKG.NEW_NOTIF

JMS Adapter Services

Name	Description
OUNCNotifPersistenceJMSConsumer	This adapter service consumes the message from the JMS Queue NotifPersistenceQueue

SMPP Connector Notification Integration Flow

Business Details

This BPEL Process consumes SMS messages (durable subscribers) from *NotifDeliveryTopic* and relays those messages to Universal Messaging Server (UMS).

Technical Details

Integration BPEL process does the following:

- Consume message

OUNCNotifDeliveryTopicSmppConsumer consumes message from the JMS Notification Delivery Topic (NotifDeliveryTopic).

- Assign SMS Parameters

Assign the correct values into SMS parameters needed to invoke Notification service using UMS

- Invoke Notification Service

Invoke notification service using UMS to send the actual SMS message.

- **Error Handling**

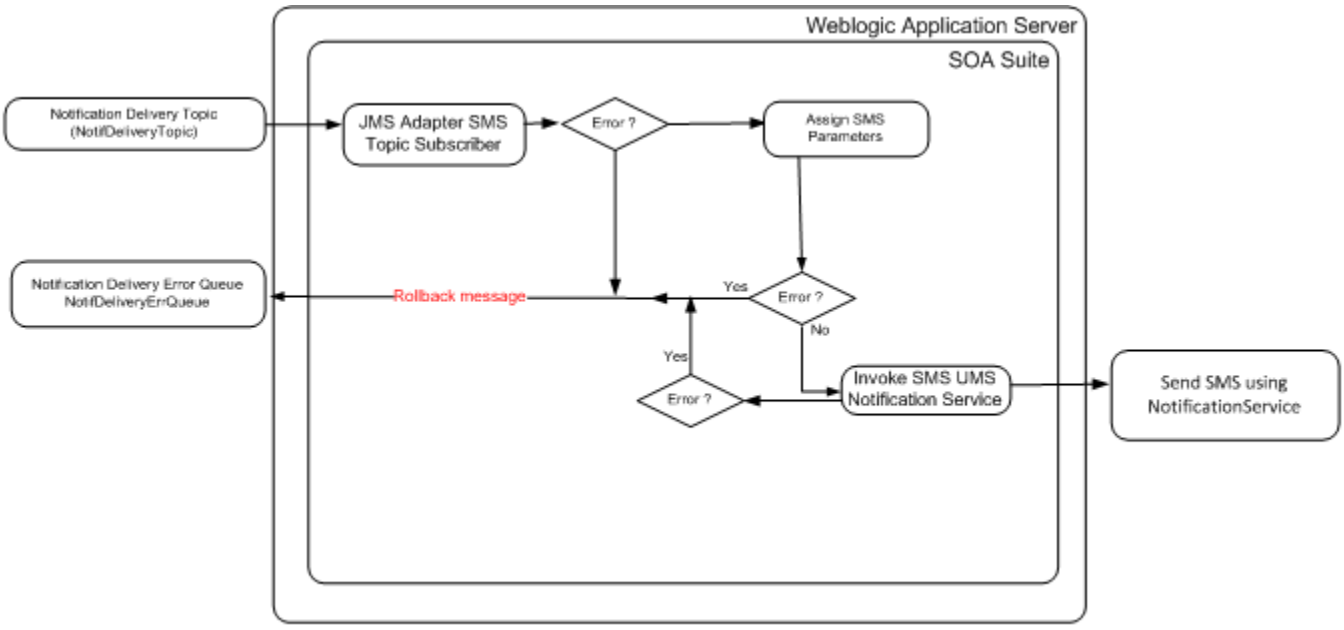
When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **Customization**

The pre invoke extension scopes is invoked before invoking the UMS Notification service.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCSmppConnector	This BPEL Process consumes SMS messages (durable subscribers) from NotifDeliveryTopic and relays those messages to Universal Messaging Server (UMS)

JMS Adapter Service

Name	Description
OUNCNotifDeliveryTopicJMSConsumer	This adapter service consumes the message from the JMS Queue NotifPersistenceQueue

Web Services

Name	Description
NotificationService	This web service is used to invoke the notification service to send SMS using UMS

EMAIL Connector Notification Integration Flow

Business Details

This BPEL Process consumes Email messages (durable subscribers) from *NotifDeliveryTopic* and relays those messages to Universal Messaging Server (UMS).

Technical Details

Integration BPEL process does the following:

- Consume message

OUNCNotifDeliveryTopicEmailConsumer consumes message from the JMS Notification Delivery Topic (NotifDeliveryTopic).

- Assign Email Parameters

Assign the correct values into Email parameters needed to invoke Notification service using UMS.

Note: The From Account of the Notification service specifies the name of the account used to send this message. The default account is named Default and is editable from the Workflow Notification Properties page in Oracle Enterprise Manager Fusion Middleware Control as mentioned in the previous section. To add additional accounts, you must use the System MBean Browser in Oracle Enterprise Manager Fusion Middleware Control.

For information on editing “From Account Name” property in Oracle Enterprise Manager Fusion Middleware Control, see *Oracle Fusion Middleware Administrator's Guide for Oracle SOA Suite and Oracle Business Process Management Suite*.

Create ASNSDriver using the notification type as the driver Name for each of the notification types in the system (eg. OUT, RST, C1MP, C1PC, C1PP, WSLP, WSPR, WXBD, WXBR, C1FI). Use setASNSDriver and associate a different sender email addresses for each notification type. When email is received, depending on the notification type, the “From” email address will be different.

The “from” email address coming from CCB payload is mapped to the "replyTo" of the notification service.

- Invoke Notification Service

Invoke notification service using UMS to send the actual Email message.

- **Error Handling**

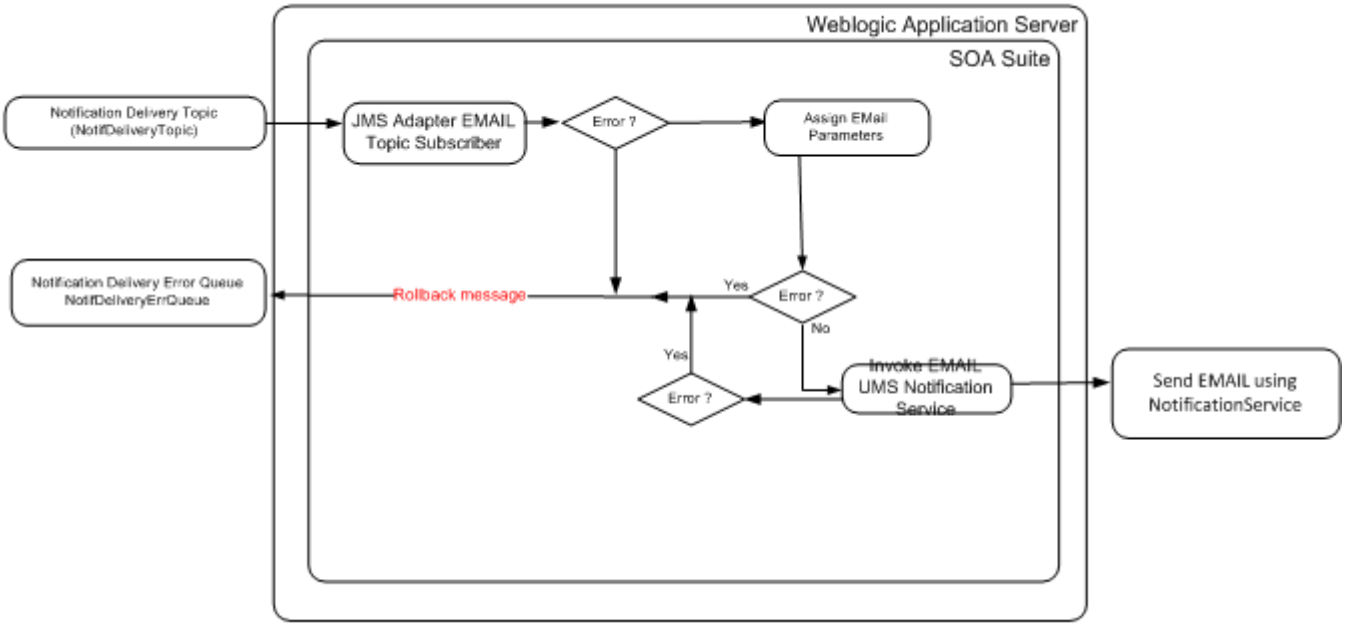
When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **Customization**

The pre invoke extension scopes is invoked before invoking the UMS Notification service.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCEmailConnector	This BPEL Process consumes Email messages (durable subscribers) from NotifDeliveryTopic and relays those messages to Universal Messaging Server (UMS)

JMS Adapter Services

Name	Description
OUNCNotifDeliveryTopicJMSConsumer	This adapter service consumes the Email message from the JMS Queue NotifPersistenceQueue

Web Services

Name	Description
NotificationService	This web service is used to invoke the notification service to send Email using UMS

Delivery Deferred Notification Integration Flow

Business Details

Some notification delivery types may have “Do Not Disturb” settings. If these are present and the time period is active, the actual notification processing is delayed until after the Do Not Disturb period.

This BPEL process polls the database table for any deferred messages that can be delivered and puts them in NotifDeliveryTopic for the notification to be sent out.

Technical Details

Integration BPEL process does the following:

- Poll Database for Deferred Notifications

The input DB Adapter OUNCNotifDeferredPollerDBAdapter polls the database table/view NC_NOTIF_DEFERRED and NC_NOTIF_DND_AVAILABLE for deferred notifications that can be sent at the current time. Records are deleted from the table once the notifications have been read.

- Relay Message
  - Message Transformation for Relay

    For each read notification, transform the notification message to a format to be put into the topic.
  - Push Notification Message to Delivery Topic

    After transformation, invoke JMS adapter to push the message into the Notification Delivery Topic (NotifDeliveryTopic)

- **Error Handling**

When a business or technical error is encountered in the Integration, a SOAP fault will be returned with a specific business or technical message code. The message codes are obtained from the configuration properties file.

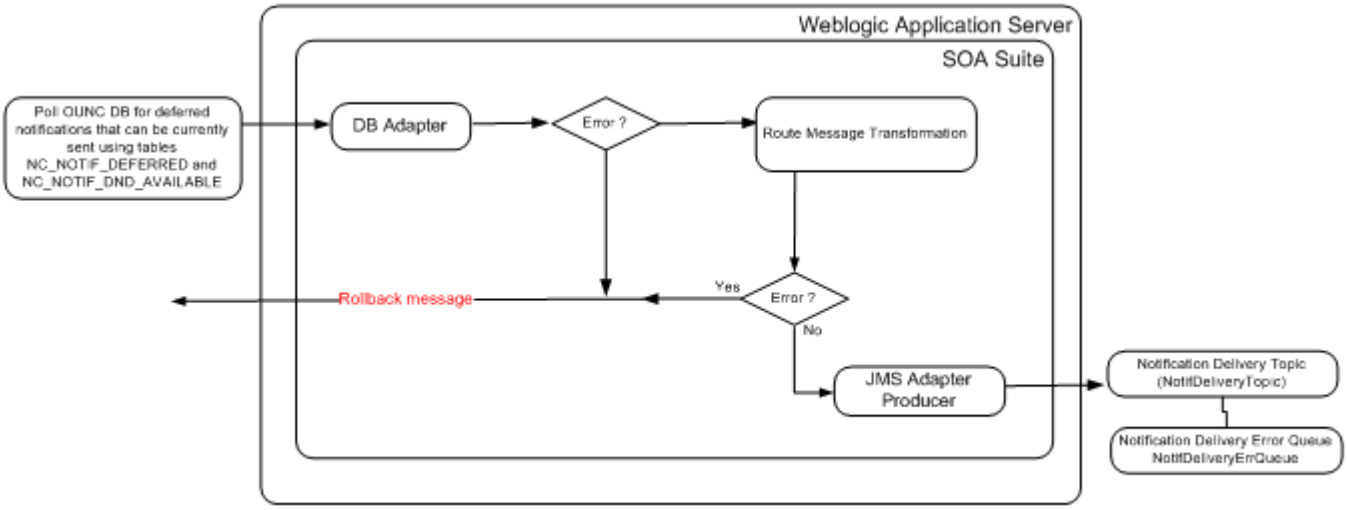
- **Customization**

The pre transformation extension scope is invoked before the message is routed.

The pre invoke extension scopes are invoked before putting the message into the Delivery Topic. This will help the implementers to change the message as required.

Note: Refer to **Customization and Extension Methodology** below for more information about customization.

Technical Flow



Integration Services

Name	Description
OUNCDeliverDeferred	This BPEL process polls the database table for any deferred messages that can be delivered and puts them in

	NotifDeliveryTopic for the notification to be sent out

DB Adapter Services

Name	Description	Operation
OUNCNotifDeferredPollerDBAdapter	This adapter service polls the database table/view NC_NOTIF_DEFERRED and NC_NOTIF_DND_AVAILABLE for deferred notifications	Poll the database table/view NC_NOTIF_DEFERRED and NC_NOTIF_DND_AVAILABLE for deferred notifications

JMS Adapter Services

Name	Description
OUNCNotificationDeferredJMSProducer	This adapter service puts the notification message that was previously deferred to be relayed to the topic Notification Delivery Topic

Send Optin Message Integration Flow

Business Details

This BPEL process is invoked as part of the Opt-in lifecycle when OUNC owns delivery channels, notification preferences and Optin lifecycle of SMS types. This process sends Opt-in SMS messages to the phone numbers based on the AUDIT_SMS notification types sent as part of the request. This process is called from the OUNCWXDeliveryChannels BPEL process when the OUCSS user enters a new SMS number as a delivery channel in the Notification Profile screen. This process is also called from OUNCProcessAuditMessage (Notifications.owner = OUNC) to send and persist the corresponding SMS messages in response to the SMS keywords (notification types) sent by the SMS user.

Technical Details

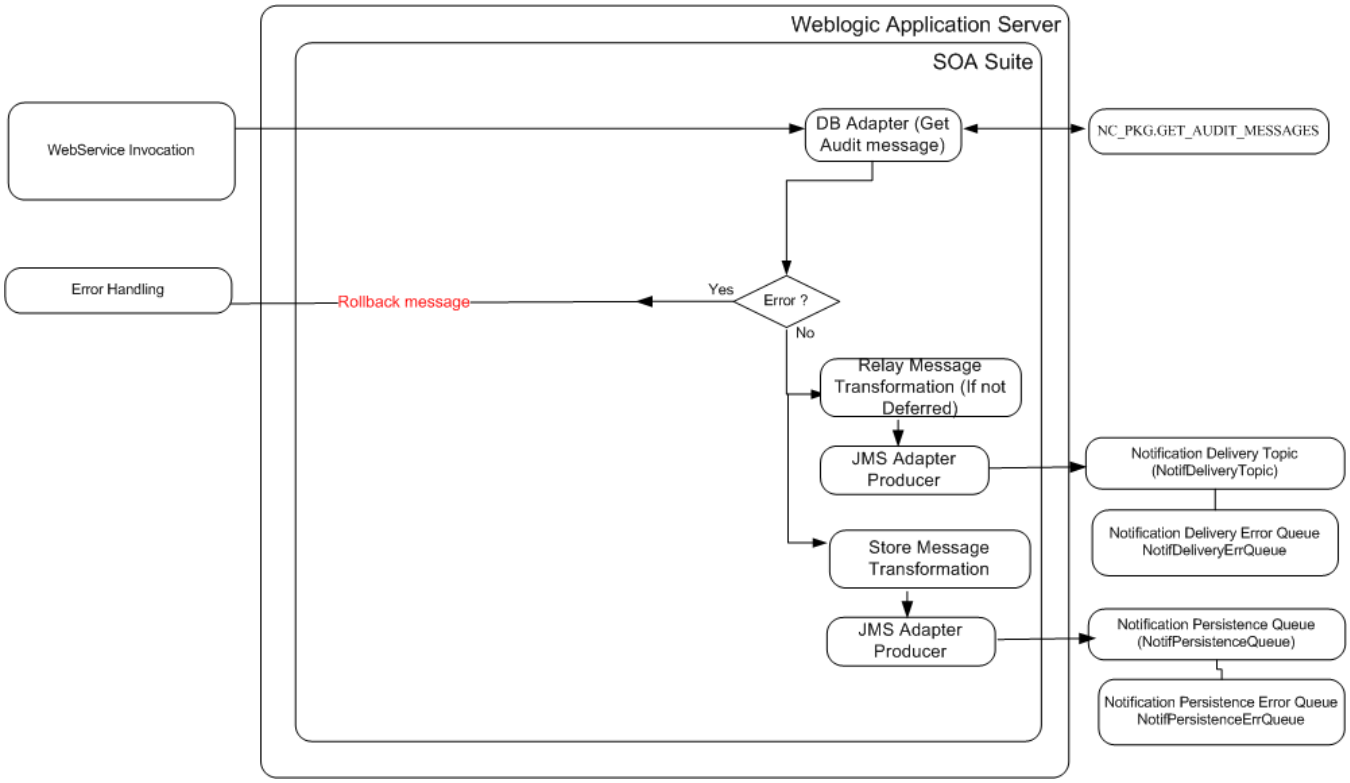
The following steps describe the flow when a request is received to send and persist an Opt-in SMS message:

1. Get the “Default.Audit.Language” property set in the ConfigurationProperties.xml file.
2. Get the SMS message and the subject that will be sent by calling the NC_PKG.GET_AUDIT_MESSAGES stored procedure through the OUNCGetAuditMessageDBAdapter DB Adapter service.
3. Transform the message to the format required by the consumer OUNCPersistNotification process.
4. Push the message into the JMS Queue (NotifPersistenceQueue) through the JMS Adapter service OUNCPersistNotification. The existing BPEL flow OUNCPersistNotification consumes the message from this queue and persists the message to the NC_NOTIFICATION table. The audit message record in the NC_NOTIFICATION table is stored with the APP_ID as ‘AUDIT_SMS’ for Opt-in messages received and sent. USER_ID and KEY1 are populated with the SMS number to which the message is sent or received.
5. Transform the message to the format required by the consumer OUNCSmppConnector process.
6. Push the message into the JMS Notification topic (NotifDeliveryTopic) through the JMS Adapter service OUNCRelayNotificationToGateway. The existing BPEL flow OUNCSmppConnector consumes the message from this topic and sends out the SMS message through UMS.

Error Handling: When a technical error is encountered in the Integration, a SOAP fault is thrown and the message is rolled back. If an error occurs in the OUNCSmppConnector BPEL process or OUNCPersistNotification BPEL process, the message is moved to the respective error queues.

- Customization:

- The pre-transformation extension scope is invoked before the main transformation is executed.
- The pre-invoke extension scope is invoked before putting the message into the queue. This helps the implementers change the message as required.
- Override custom transformations are called after the main transformation. These override custom transformations can be used to change or override values to existing elements.



Integration Services

Name	Description
OUNCSendOptInMessage	This BPEL process pushes the Optin Message to the corresponding Optin Notification type into NotifPersistenceQueue and NotifDeliveryTopic for the Optin message to be delivered to the delivery contact and also persist in OUNC NC_NOTIFICATION table

DB Adapter Services

Name	Description	Operation
OUNCGetAuditMessageDBAdapter	This adapter service invokes the plsql procedure to get the SMS messages and subject	NC_PKG.GET_AUDIT_MESSAGES

JMS Adapter Services

Name	Description
OUNCRelayNotificationToGateway	This adapter service puts the notification message to be relayed to the topic Notification Delivery Topic

OUNCPersistNotification	This adapter service puts the notification message into the persistence queue NotifPersistenceQueue

Process Audit Message Integration Flow

Business Details

This process consumes the messages in NotifReceiveQueue populated by ReceiveNotifUMSClient with the SMS message (SMS keyword configured) sent by the user. It does the required processing needed based on the message received. If any error is encountered , the message is rolled back to ReceiveNotifErrQueue.

Technical Details

The following steps details the flow when a message is consumed from NotifReceiveQueue:

When Notification.Owner = 'OUNC':

1. The processing takes place only if the Opt-in type set in ConfigurationProperties.xml is either SINGLE or DOUBLE.
2. For the SMS message received , the corresponding predefined notification type mapped to the received message is retrieved. This is done through a call to the stored procedure NC_PKG.GET_AUDIT_NOTIF_TYPE through a OUNCMsgNotifTypesDBAdapter DB Adapter.
3. If the received SMS is a valid AUDIT_SMS notification type, the received message is persisted. The message is first transformed to the format required by the consumer OUNCPersistNotification process. The message is then pushed into JMS Notification Queue (NotifPersistenceQueue) through JMS Adapter service OUNCPersistNotification. Existing BPEL flow OUNCPersistNotification consumes the message from this queue and persists the message to the NC_NOTIFICATION table. The audit message record in the NC_NOTIFICATION table will have APP_ID set to 'AUDIT_SMS'. USER_ID and KEY1 are populated with the SMS number that is sent.
4. If the received SMS is not a valid AUDIT_SMS notification type and if the 'Help.On.No.Match' is set to "Y' in the configuration properties , then 'HELP' notification type will be used. If Help.On.No.Match is set to 'N' and if the SMS received is not a valid notification type then no response is returned.
5. If the message corresponds to the CONFIRM notification type and is a DOUBLE Opt-in type, the VERIFIED column in NC_USER_DELIVERY_OPT table is set to 'Y' for all the records which have this SMS number as a delivery channel. This is done through a NC_PKG.SET_AUDIT_VERIFIED stored procedure call that is called through a OUNCAuditVerifiedDBAdapter DB Adapter service.
6. If the message corresponds to the STOP notification type, then a call to stored procedure NC_PKG.GET_USERS_FOR_DELIVERY_CHANNEL is made through the OUNCUsersForDelivChannelDBAdapter DB Adapter service. This stored procedure returns all users that have this SMS number as a delivery channel.
7. The records returned from this stored procedure call is transformed to a format as specified in preferencesData.xsd
8. For each preferenceList element obtained as a result of the transformation in step 6 that has notification preferences:
   - **A** If there are notification preferences associated with the delivery channel, the message is transformed to the format needed for the OUNCWXNotificationPreferences BPEL process and OUNCWXNotificationPreferences is called with a READ action to get all the notification preferences for that user.
   - **B** The message is then transformed for OUNCWXNotificationPreferences on UPDATE to update Notification center tables and service tasks in CCB and NMS . The IP address sent is defaulted to 127.0.0.1, since the IP address will not be available in the request.
   - **C** A stored procedure call to NC_PKG_DELETE_USER_CHANNEL is made through the OUNCDeleteDeliveryChannel DB Adapter service to delete all records corresponding to the SMS number from which the STOP message was received.

9 For each of the preferenceList elements obtained as a result of the transformation in step 6 that does not have notification preferences:

   A A stored procedure call to NC_PKG_DELETE_USER_CHANNEL through OUNCDeleteDeliveryChannel DB Adapter service to delete all the records corresponding to the SMS number from which the STOP message was received.

10 OUNCSendOptIn Message is called to send appropriate message related to the notification type (HELP,CONFIRM,STOP) and also to persist the message sent.

When Notification.Owner = 'CCB':

1 For the SMS message received , the corresponding predefined notification type mapped to the received message is retrieved. This is done through a call to the stored procedure NC_PKG.GET_AUDIT_NOTIF_TYPE through a OUNCMsgNotifTypesDBAdapter DB Adapter.

2 If the received SMS is not a valid AUDIT_SMS notification type and if the 'Help.On.No.Match' is set to "Y' in the configuration properties , then 'HELP' notification type will be used. If Help.On.No.Match is set to 'N' and if the SMS received is not a valid notification type then no response is returned.

3 Get the corresponding CCB Action as defined in the configuration.

4 Transform the CCB response message to the request format of C1-ProcessOptinReply CCB service

5 Invoke C1-ProcessOptInReply CCB service

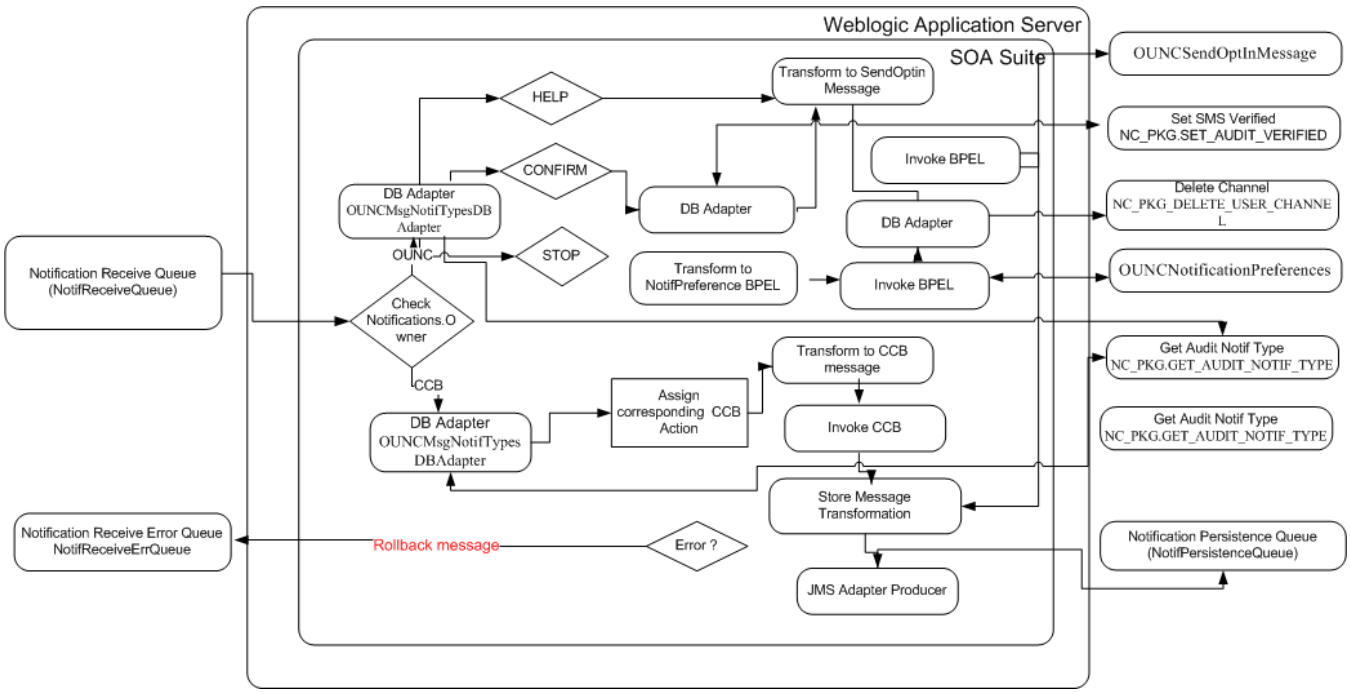
6 Transform the response to the message format of NotifInboundQueue

7 The message is then consumed by OUNCRouteNotification, and routed to NotifDeliveryTopic for message delivery and NotifPersistenceQueue to persist the message in NC_NOTIFICATION table

Error Handling: When a business or technical error is encountered in the Integration, the message is rolled back to NotifReceiveErrQueue.

Customization:

- The pre-transformation extension scope is invoked before the main transformation is executed.
- The pre-invoke extension scope is invoked before putting the message into the queue. This helps the implementers change the message as required.
- Override custom transformation are called after the main transformation. These override custom transformations can be used to change or override values to existing elements.



Integration Services

Name	Description
OUNCProcessAuditMessage	• This BPEL process consumes messages from NotifReceiveQueue and processes the SMS response of the user.

DB Adapter Services

Name	Description	Operation
OUNCMsgNotifTypesDBAdapter	This adapter service invokes the plsql procedure to get the SMS message and SMS subject.	NC_PKG.GET_AUDIT_NOTIF_TYPE
OUNCAuditVerifiedDBAdapter	This adapter service marks the SMS delivery channel as verified for the corresponding SMS number by setting VERIFIED='Y' in the NC_USER_DELIVERY_OPT table.	NC_PKG.SET_AUDIT_VERIFIED
OUNCUsersForDelivChannelDBAdapter	This adapter service returns all the users that have this SMS number as a delivery channel.	NC_PKG.GET_USERS_FOR_DELIVERY_CHANNEL
OUNCDeleteDeliveryChannel	This adapter service deletes records from NC_USER_DELIVERY_OPT table corresponding to the delivery channel ID.	NC_PKG_DELETE_USER_CHANNEL

JMS Adapter Services

Name	Description
OUNCPersistNotification	This adapter service pushes the notification message into the NotifPersistenceQueue.

Web Services

Name	Description	Operation
OUNCSendOptInMessage	This BPEL process is called to send and persist SMS messages corresponding to the notification type back to the user.	
OUNCNotificationPreferences	This BPEL process is called on STOP notification type received as SMS by the user.	
C1-ProcessOptInReply	This is a CCB web service which is called	

Receive Audit Message Integration Flow

Business Details

This process receives the SMS message from UMS and populates the NotifReceiveQueue.

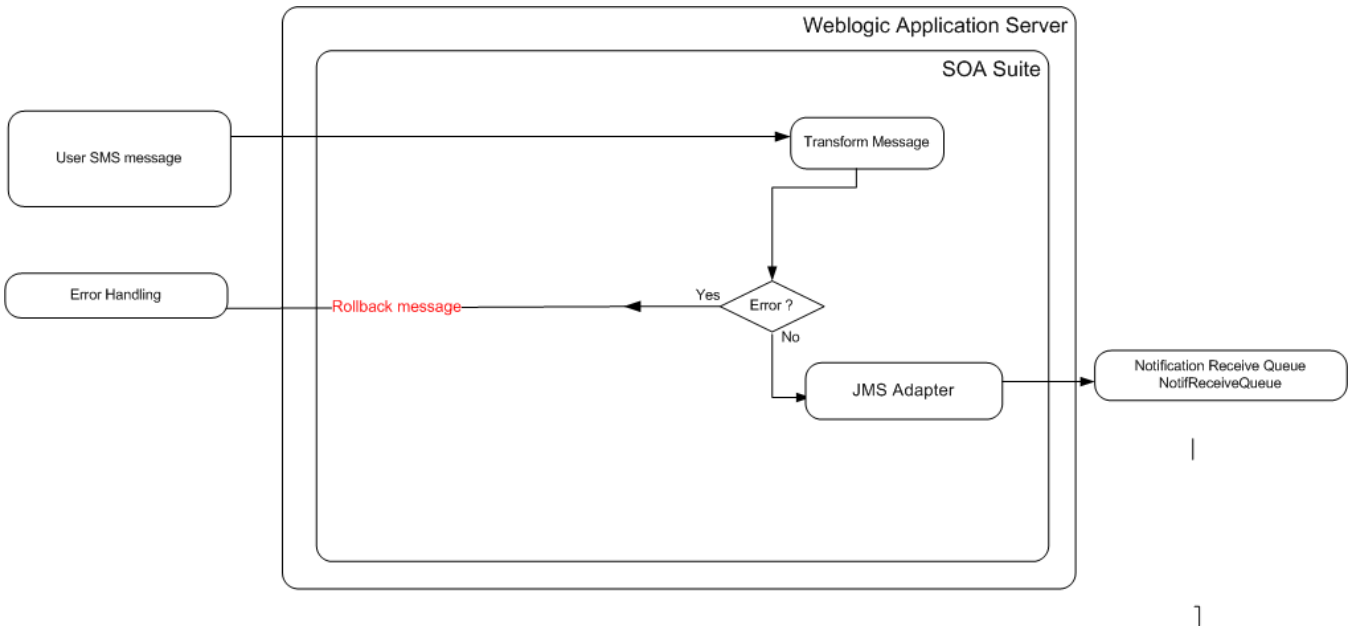
Technical Details

The following describes the flow when an SMS message is received by UMS:

1. The message is received by the UMS Adapter for SMS .
2. The message is transferred to Java callout ProcessMessage.
3. Sender and Recipient information is retrieved and sent to a BPEL process.
4. The message is transformed to the format required by OUNCProcessAuditMessage.
5. The message is pushed to NotifReceiveQueue.

Error Handling: When an error is encountered in the integration, a SOAP fault is thrown and the message is rolled back.

- Customization:
  - The pre-invoke extension scope is invoked before putting the message into the queue. This helps the implementers to change the message as required.
  - Override custom transformations are called after the main transformation. These override custom transformations can be used to change or override values to existing elements.



Integration Services

Name	Description
OUNCReceiveAuditMessage	This process receives the SMS message from UMS and populates the NotifReceiveQueue.

UMS Adapter Service

Name	Description
ReceiveUMSService	This adapter receives SMS messages from UMS

JMS Adapter Service

Name	Description
ReceiveMsgReference	This adapter service puts the transformed message into the NotifReceiveQueue, which is consumed by the OUNCProcessAuditMessage service.

Notification History Integration Flow

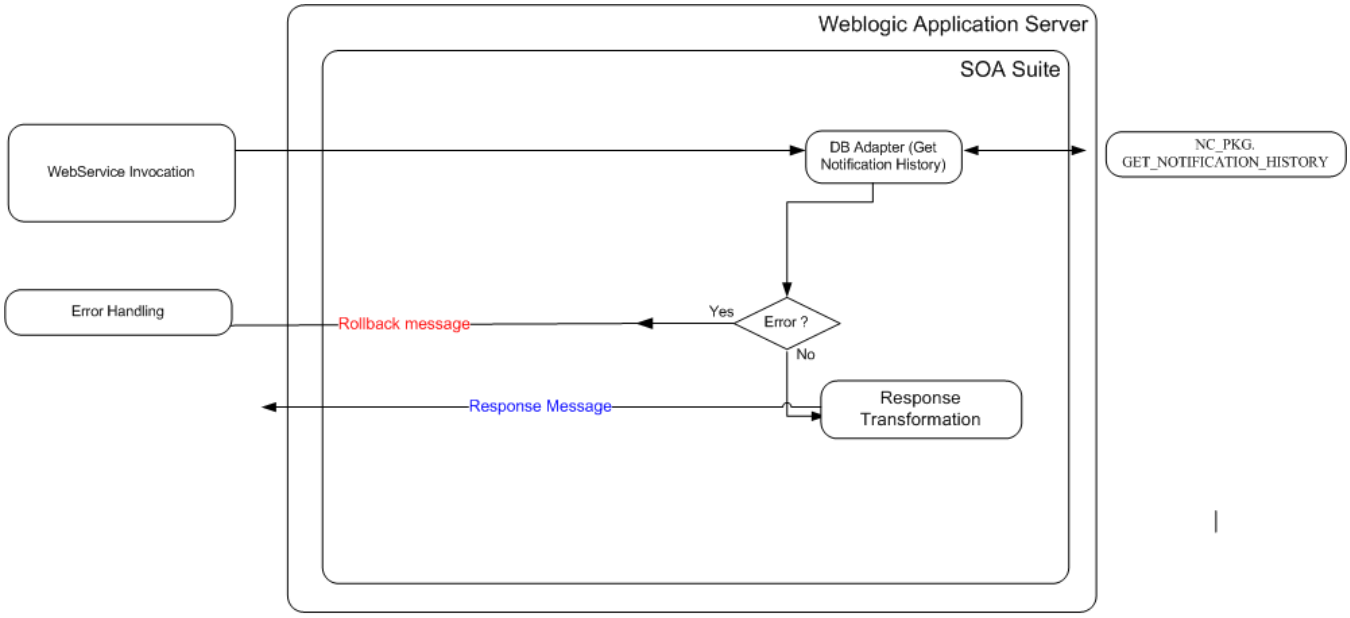
Business Details

This process retrieves history records from NC_NOTIFICATION table for a given accounted ,personId ,startDate and endDate.

TechnicalDetails

The input parameters to OUNCGetNotificationHistoryDbAdapter is assigned and the db adapter is called to retrieve records from NC_NOTIFICATION for a given accounted ,personId ,startDate and endDate.

The response from the OUNCGetNotificationHistoryDbAdapter is transformed to the reply format. Max.History.Records configuration property determines the maximum number of records that are retrieved. If the total records exceeds Max.History.Records limit then the response will populate only “excessRowCount” element with the total number of records matching the search criteria



Integration Services

Name	Description
CCBOUNCNotificationHistory	This process retrieves the history records from NC_NOTIFICATION table

DB Adapter Services

Name	Description	Operation
OUNCGetNotificationHistoryDbAdapter	This db adapter is calls plsql stored procedure to retrieve records from NC_NOTIFICATION for a given accounted ,personId ,startDate and endDate	NC_PKG. GET_NOTIFICATION_HISTORY

Notification Engine Integration Queues

Name	Description	Type
NotifInboundQueue	It is the queue a notificaiton message is put into once it is received from the edge application from a web service call for CCB and a db call for NMS	Queue
NotifInboundErrQueue	Error queue for NotificationInboundQueue	Queue
NotifDeliveryTopic	Queue that containss SMS or EMAIL notification messages that need to be delivered via UMS	Topic
NotifDeliveryErrQueue	Error queue for the Notification Delivery Queue	Queue
NotifPersistenceQueue	Queue that contains messages that need to be persisted to the Notification database.	Queue
NotifPersistenceErrQueue	Error queue for Notification Persistence Queue	Queue
NotifReceiveQueue	Queue to receive user responses	Queue
NotifReceiveErrQueue	Error queue for NotifReceiveQueue	Queue

Notification Integration Product Configuration

The following sections describe the configuration needed in the integration to meet the requirements for this integration.

Configuration steps include setting the following:

Task	Remarks
Setting Notification Types	Setting up Notification Types
Setting Configuration Properties	Update the OUNC ConfigurationProperties.xml file.

Setting System Properties	Set the Module Configurations properties that are shared by multiple integration flows and Service Configurations properties that are used by a specific BPEL process.
Domain Value Maps	Set the Domain value maps (DVMs) to map codes and other static values across applications.
Error Handling	Set up error notifications for the Notifications Engine.

Setting Notification Types

The valid NMS notification types used by the application are defined in the Notification Center Database. The tables that hold these values are:

- NC_NOTIF_TYPES
- NC_NOTIF_TYPES_L
- NC_APP_NOTIF_TYPES

These notification types are automatically loaded to the database during install. These are the available notification types that are shipped with the product.

Note: Notification Type **ERT – Estimated Restore Type** is a new NMS Notification Type. It is available only if patch 19022578 is installed.

EdgeApp	Language	Notification Type (Default/Shipped Value)	Description
NMS	ENG	OUT	Outage
NMS	ENG	RST	Power Restored
NMS	ENG	ERT	Estimated Restore Time

Note: If the CSS system supports other languages, please define an equivalent notification type description for each notification type and language supported.

Example: If CSS supports English and French, you need to define the outage description in the OUNC tables for each supported language. Add corresponding values to the database.

EdgeApp	Language	Notification Type (Default/Shipped Value)	Description
NMS	FRA	OUT	Outage French Description
	ENG	OUT	Outage

The following database tables are used to keep language related types definitions (see example for ERT type):

Table	Field=Value
NC_NOTIF_TYPES	NOTIF_TYPE = ERT OWNER_FLG = NC VERSION = 1
NC_NOTIF_TYPES_L	NOTIF_TYPE = ERT LOCALE = FRA DESCR = Estimated Restore Time in French OWNER_FLG = NC VERSION = 1

Overriding SMS Notification Type Mapping for Optin

The NC_LOOKUP, NC_LOOKUP_L, NC_LOOKUP_VAL, and NC_LOOKUP_VAL_L tables are used to map the SMS notification types corresponding to 'CONFIRM', 'STOP', 'STOPALL', and 'HELP'. By default, 'CONFIRM', 'STOP', 'STOPALL', and 'HELP' notification types are mapped to 'CONFIRM', 'STOP', 'STOPALL', and 'HELP' keywords. These mappings can be overridden as described below.

Note: STOPALL notification type is used only by CCB, when Notifications.owner configuration property is set to CCB

Example: To override the CONFIRM' mapping on LOCALE='ENG':

1. Run the following SQL statement to deactivate the existing CONFIRM record in NC_LOOKUP_VAL:

```
UPDATE NC_LOOKUP_VAL
SET STATUS      ='INACTIVE'
WHERE LOOKUP_CD = 'CONFIRM'
AND OWNER_FLG   = 'NC';
```

2. Run the following SQL statement, replacing <NEW_VALUE> with the custom keyword:

```
INSERT
INTO NC_LOOKUP_VAL
  (
    LOOKUP_CD,
    LOOKUP_VALUE,
    STATUS,
    OWNER_FLG,
    VERSION
  )
  VALUES
  (
    'CONFIRM',
    '<NEW_VALUE>',
    'ACTIVE',
    'CM',
    1
  );
```

3. Insert a language record in NC_LOOKUP_VAL_L table for the CM CONFIRM value. You can use the following SQL statement, replacing <NEW_VALUE> with the CM CONFIRM value, and <CUSTOM_DESCR> with a short description of the CM value:

```
INSERT
INTO NC_LOOKUP_VAL_L
  (
    LOOKUP_CD,
    LOOKUP_VALUE,
    LOCALE,
    DESCR,
    OWNER_FLG,
    VERSION
  )
  VALUES
  (
    'CONFIRM',
    '<NEW_VALUE>',
    'ENG',
    '<CUSTOM_DESCR>',
    'CM',
    1
  );
```

4. Commit the changes.

Overriding SMS Messages Sent Out to the User for Optin

SMS messages sent out from the Notification center are stored in the NC_MESSAGE and NC_MESSAGE_L table. This is used only when Notifications.owner = OUNC. When Notifications.owner = 'CCB', CCB sends the Optin message that needs to be sent out to the user as a response to CCB's C1-ProcessOptInReply service call in OUNCProcessAuditMessage BPEL flow

Default SMS messages associated with each of the notification type can be overridden as described below.

Example: To override the CONFIRM message for the 'ENG' locale, you can run the following SQL statement, replacing the <CUSTOM_CONFIRM_MSG> with the custom CONFIRM message:

```
UPDATE NC_MESSAGE_L
SET DESCR_OVRD   = '<CUSTOM_CONFIRM_MESSAGE>'
WHERE MESSAGE_CD = 'CONFIRM'
AND LOCALE       ='ENG'
AND OWNER_FLG    ='NC';
```

Overriding the SMS Subject for Optin

SMS messages sent require a subject. These subject descriptions correspond to the SMS notification types stored in the NC_NOTIF_TYPES_L table.

```
UPDATE NC_NOTIF_TYPES_L
SET DESCR_OVRD  = '<CUSTOM_CONFIRM_SUBJECT>'
WHERE NOTIF_TYPE= 'CONFIRM'
AND LOCALE      ='ENG';
```

Setting Configuration Properties

The ConfigurationProperties.XML file contains properties which can be defaulted in the integration. ConfigurationProperties.XML is located in MDS under the directory apps/OUNC/AIAMetaData/config.

Note. Whenever the ConfigurationProperties.XML file is updated, it must be reloaded to MDS for updates to be reflected in the applications or services that use the updated properties. You can perform the reload by rebooting the SOA server.

Setting System Properties

There are two sets of configuration properties described in this section:

- Module Configurations are the properties that are shared by multiple integration flows within the Oracle Utilities Notification Center Integration Pack.
- Service Configurations are the properties that are used by a specific BPEL process.

Module Configurations

Module Name	Default / Shipped Value	Description
NC.TechnicalFault.MessageCode	SYSTEM_UNAVAILABLE_EXCP_MSG	This value should be the generic message code setup in OUNC for technical errors (e.g., when the edge apps are down). This is the message code that the integration process passes back when a technical fault is encountered. Used by all the OUNC flows
NC.Generic.ExceptionCode	999999999	This is the exception code that the integration process passes back when a fault is encountered in the integration.
SOA-INFRA.AuditLevel	ON	This property needs to be set to OFF if the Audit Level is set to OFF for the BPEL processes. If the setting is OFF, then error handling does not use the composite and component instance IDs to log the error message.
ErrorHandling.GenericEmailID		This property is used to set the administrator email ID for the error handling process to send out an email in case of a critical failure where even the Errorhandling process fails.
Default.DeliveryType.Email	EMAIL	Delivery type of Email
Default.DeliveryType.SMS	SMS	Delivery Type of Short Message
NMS.Language	ENG	NMS currently supports only one language at a time. Define the language that the NMS system is using or supporting. This is the language used when getting inbound Notifications from NMS. Default is ENG.
Default.Audit.Language	ENG	The SMS messages, subject, and notification type mappings use the language configured in this property.
Default.OptIn.Type	DOUBLE	The Opt-in property configured. Possible values can be SINGLE, DOUBLE, or NONE.
Notification.Owner	OUNC	This property indicates if CCB or OUNC is the Notifications owner, managing the preferences and delivery channels
CONFIRM.CCB.Action	CONFIRM	This is the CCB action that is sent servicescorresponding to

		'CONFIRM' notification typein OUNCProcessAuditMessage when calling C1-ProcessOptInReply CCB
STOP.CCB.Action	UNSUB	This is the CCB action that is sent servicescorresponding to 'STOP' notification typein OUNCProcessAuditMessage when calling C1-ProcessOptInReply CCB
STOPALL.CCB.Action	UNSUBALL	This is the CCB action that is sent servicescorresponding to 'STOPALL' notification typein OUNCProcessAuditMessage when calling C1-ProcessOptInReply CCB
HELP.CCB.Action	HELP	This is the CCB action that is sent servicescorresponding to 'HELP' notification typein OUNCProcessAuditMessage when C1-ProcessOptInReply CCB
Max.History.Records	100	This is the maximum records that is sent through CCBOUNCNotificationHistory BPEL flow
Help.On.No.Match	N	If the user SMS response to Optin request does not match any of the configured value SS_LOOKUP and SS_LOOKUP_VAL tables, then the HELP message will be sent
CCB.Filtered.NotifTypes		This property will indicate the CCB Optin types that has to be filtered when OUNCWXGetNotifications BPEL process is called. The values must be separated by ",". For example: <CCB.Filtered.NotifTypes>OPTIN_HELP,OPTIN_CONFIRM</CCB.Filtered.NotifTypes>

Service Configurations

Property Name	Default / Shipped Value	Description
Service Name : OUNCWXDeliveryChannels		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCWXNotificationPreferences		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
CCB.EnabledFlag	true	Default is set to true, notification preferences are enabled for CCB Notification types
NMS.EnabledFlag	true	Default is set to true, notification preferences are enabled for NMS Notification types
CCB.WXNotificationPreferences.Endpoint.URL		This value is the CCB Service Details Endpoint URL. Shipped with this value: @EdgeApplications.OUCCB.ManagedServer.protocol://@EdgeApplications.OUCCB.ManagedServer.hostname:@EdgeApplications.OUCCB.ManagedServer.portnumber/@E

		dgeApplications.OUCCB.ManagedServer.context/Webservices/ WXSetNotificationPreferences During install, the CCB edge application information will be tokenized to point to the correct CCB server being used.
Service Name : OUNCWXGetNotifications		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCNMSNotificationInbound		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCNotificationInbound		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCDeliverDeferred		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCEmailConnector		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUNCSmppConnector		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUNCPersistNotification		
Default.SystemID	OU_NC_01	Initiating system ID.

TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUNCRouteNotification		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUNCSendOptInMessage		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : OUNCProcessAuditMessage		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : OUNCReceiveAuditMessage		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
BusinessError.NotificationFlag	false	If set to true, Business error notification is sent via Email.
Service Name : CCBOUNCNotificationInbound		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.
Service Name : CCBOUNCNotificationHistory		
Default.SystemID	OU_NC_01	Initiating system ID.
TechnicalError.NotificationFlag	false	If set to true, technical error notification is sent via Email.

Notification Center Data Store

Transactional Data Tables

The following are some of the important tables that store the transactional data.

NC_USER_DELIVERY_OPT

This table stores delivery channels per user. For instance, user can define 2 Emails, 3 SMS numbers, and so on when Notifications.owner = OUNC.

NC_USER_NOTIF_PREF

This table stores user's preferences per notification type per account. Available NMS notification types are stored in the Notification Center schema and should match those exposed by the edge applications when Notifications.owner = OUNC

NC_USER_NOTIF_PREF_DELIV_OPT

Many-to-many table between notification preferences and delivery channels defined by the user when Notifications.owner = OUNC

NC_NOTIFICATION

This table stores the actual notification messages sent by the edge application.

NC_NOTIF_DEFERRED

Stores IDs of messages that need to be delivered later (e.g., Do not Disturb setting is on). Once the message can be delivered, it is placed in the regular delivery queue and erased from the table. This table is polled by a BPEL database connector for notifications that are available for delivery in a batch fashion.

Package

NC_PKG is the PLSQL package used by the DB Adapters in BPEL Processes to insert, update and delete transactional data.

Chapter 9

Customization and Extension

OUCSS core taskflows/portlets can be customized using the ADF customization methodology. ADF/JDeveloper customizations are stored in MDS. The OUCSS applications are preconfigured to allow for customization. To support the extension, CCB inbound web services expose 10 custom fields (fields1 through 10) in the "custom" node of the WSDL both in request and response payloads. These custom fields can be used to dynamically extend the taskflows/portlets without redeploying any piece of code.

Note: For simple customization, taskflows can also be customized at runtime using WebCenter Composer. But for more complex customizations, design time customization is required using JDeveloper.

Steps to Customize OUCSS Taskflows (DT)

Creating Customization Documents

1. In CCB, configure custom fields (fields 1 through 10) that are intended to be extended for a given module/webservice.
2. In the OUCSS Package zip file downloaded from OTN, find the ExtendOUCSSPortal.zip. Extract the zip. OUCSS taskflows can be customized using **ExtendOUCSSPortal** workspace. Open ExtendOUCSSPortal workspace in JDeveloper.

   Note: For more ExtendOUCSSPortal workspace, refer to the "Extending the OUCSS Portal" section in *Customizing and Extending the OUCSS Custom Portal Whitepaper*, available for download in the Oracle Utilities Customer Self Service section of the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://docs.oracle.com/cd/E72219_01/documentation.html)
3. In the OUCSS Package zip file downloaded from OTN, find the OUCSS_Extension.war or OUCSS_Commercial_Extension.war containing the taskflow of the module that needs to be customized.
4. For example, if you need to extend Account Summary module, then copy **oracle.ugbu.ss.billing.accountsummary.model.jar** and **oracle.ugbu.ss.billing.accountsummary.view.jar** to a CM folder (e.g., `C:\OUCSS\Portal\CM`) on the machine on which JDeveloper is running.

Note: You can extend more than one module at the same time by copying all the related ADF libraries in the same folder to facilitate customization.

5 In the Resource Palette, create a File System connection to the CM folder containing the ADF Libraries.

6 Choose the OUCSSPortalADFLibrary project from ExtendOUCSSPortal, then, from the resource palette file connection, right-click on the ADF Libraries and select/click on **Add to Project**. Repeat Add to Project for each module jar.

7 Make sure that the corresponding data control (e.g., AccountSummaryService) is listed in the Data Control panel.

8 In the JDeveloper Application Navigator, choose to show libraries.

9 From the list of Libraries displayed, find library named “ADF Library”. All the OUCSS jars added using “Add to Project” will be found under this library.

10 Expand ADF Library to find and open the desired jsff (UI) file. For this sample, we will use **summary.jsff.**

11 In order to customize/extend the content, switch to JDeveloper’s Customization Mode. To change the mode,

12 Select **Tools > Switch Roles >** Customization **Developer** in JDeveloper 12c.

13 JDeveloper will restart (on Windows) or advise you to restart JDeveloper (on Linux). Restart JDeveloper.

14 After JDeveloper restart with Customization mode, a **Customization Context** window should show on bottom right corner in JDeveloper. Make sure that "Edit with following Customization Context” radio is enabled/selected.

15 Confirm that Edit is enabled and Tip Layer/Customization Context is configured as site/webcenter. For WebCenter Portal with a standard customization configuration using the SiteCC customization class, use a cust-layer-value webcenter instead of site.

16 In customization mode, edit the page/jsff in design mode or using Structure pane. You cannot modify the source code manually but when you drag and drop data control entries into the jsff, JDeveloper will create the required customization files to record the delta of the updates. In our example, the file generated will be `summary.jsff.xml`.

17 To customize the page

Alter the properties of existing components using the Property Inspector.

Add new components by Drag and drop from Component Pallete as per the requirement.

Add a Custom Field (1 to 10) from the Data Control by dragging it to the location in the jsff where it needs to be rendered.

Optionally, customers can select other fields (which are not under custom) that are available in data control but not part of the out-of-box UI.

18 Ensure that you are extending using the corresponding Data Control of the module of the jsff.

19 All customization delta file(s) generated by JDeveloper that needs to be uploaded to MDS can be found under <<ApplicationFolder>>\ OUCSSPortalADFLibrary\libraryCustomizations folder.

20 You can open these files in JDev and manually edit them to make changes.

21 If any updates are erroneously applied, delete the *.jsff.xml and *.pageDef.xml file(s) from <<ApplicationFolder>>\ OUCSSPortalADFLibrary\libraryCustomizations\ folder and start over.

Applying the Customization

Customization document created above are required to be imported into the MDS of the application to take effect. Customizations can be imported either using JDev or WLST.

Import Using JDeveloper

To import the customizations documents using JDeveloper,

1. From the Application menu of ExtendOUCSSPortal, select Deploy and choose OUCSSCustomization.
2. Choose either “Deploy the MAR” to create a .MAR file or “Export to a Deployed Application” to directly import the customizations to the target portal/application MDS.
3. Confirm that the deployment was successful.
4. Login to the application and verify the changes. (Generally no restart is necessary).

Import Using WLST

To import the customizations documents WLST:

1. Run the WLST and connect to the server running the WebCenter Portal application as Admin user.
2. Run the WLST command importMetadata(application='<<Target_Application_Name>>', server='<<Target_Managed_Server>>', fromLocation='<<ApplicationFolder>>\ OUCSSPortalADFLibrary\libraryCustomizations', docs='/**').

   Example:

```
importMetadata(application='webcenter',
server='WC_Portal',fromLocation='/OUCSS/OUCSSPortalADFLibrary/libraryCustomizations',
docs='/**')
```

3. Login to the application and verify the changes. (Generally no restart is necessary).

Steps to Customize an Edge Application Web Service

As previously noted, the OUCSS producer application is pre-configured to allow for customization. To facilitate the extension, the CCB web services expose 10 custom fields (field1 through field 10) in the “custom” node of the WSDL.

These custom fields can be used to dynamically extend the taskflows/portlets without redeploying any piece of code. These fields will need to be manipulated by the edge application (e.g., CCB) to either populate the custom field with data, or to make use of custom data returned from OUCSS.

Create a Custom Service Script

When creating a new custom service script, the schema for the script must be the same as the original delivered product service script.

This new service script should, at some point, execute the original service script. This will maintain service upgradability.

1. Add steps to the new service script to populate/use the custom data fields as desired. These steps may be before and/or after the execution of the original service script. Processing is only limited by the facilities provided by the service script toolset.
2. You can plug-in a new script into an existing process or refer to it from the Master Configuration in the edge application (if applicable).

3 You also can create a new inbound service to execute the new custom service script and invoke the new custom web service from OUCSS

Invoke Custom Web Service

In case your implementation needs to use a new custom service in one of the edge applications (CCB or MDM), you can redirect OUCSS to call the new custom service instead of the base productized service.

Note: The following procedure describes how to call a custom IWS service. A custom IWS service can be called by changing the end point URL in CSS or BPEL.

To be able to invoke the custom inbound web service from the OUCSS application:

1 Update OUCSS configuration by changing the wsdl and endpoint URL of the OUCSS Service Name with the new custom service url in the ADF connections page of the Portal application. Refer to Chapter 2 OUCSS Implementation – OUCSS Web Services to see the list of OUCSS web services.

Refer to your *Customer Care and Billing Implementation Guide* and *Meter Data Management Implementation Guide* for further information regarding ServiceScripts and Inbound Web Services.

Customize and Extend OUCSS Portal

OUCSS Portal can be customized and extended with custom code and taskflows. For more about how to customize and extend OUCSS Portal, refer to the *Customizing and Extending the OUCSS Custom Portal Whitepaper*, available for download in the Oracle Utilities Customer Self Service section of the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://docs.oracle.com/cd/E72219_01/documentation.html).

Chapter 10

Monitoring and Troubleshooting

Monitoring Oracle Utilities Self Service

The Self Service application involves different and distributed systems, and the root cause of issues is sometimes difficult to identify. Monitoring of key elements can help isolate issues and make them easier to address.

Monitor OUCSS Portal using Enterprise Manager

1. Login to WebLogic enterprise manager as WLS Admin.
2. From the Domain Menu (e.g., Farm_<<domain_name>> on the left, expand Target Navigation.
3. Expand WebCenter -> Portal -> Service and lick on WebCenter Portal (WC_Portal) to load the WebCenter Portal application summary page.
4. Monitor the “Response and Load” and “Recent CPU and Memory Usage” graphs to get an overall status of how the application is performing.
5. To drill down more, click on Performance Summary from WebCenter Portal -> Monitoring Menu. This will load a page with more statistics and other graphs to active Sessions, Request Processing Time, Request (per min) etc.
6. To monitor how each page in Portal is performing, click on Application Deployment menu at the top and select “WebCenter Portal -> Monitoring -> Overall Page Metrics”. This will load a page to list processing times taken to load each Portal page.

Monitor Using Oracle WebLogic Logs

WebLogic logs can be monitored to get more information on exceptions and application status.

Logs can be monitored either using Oracle Enterprise Manager or by directly accessing the physical machine on which the managed servers are running. Logs monitored from EM are more interactive and allows search capabilities which makes it easier to diagnose an issue quickly.

Command line administrators can also directly use the logs on the physical machine.

Monitor Logs Using Oracle EM

OUCSS Portal Logs

1. Login to WebLogic Enterprise Manager as WLS Admin.
2. From the Target Navigation expand WebCenter -> Portal -> Server and click on WebCenter Portal (WC_Portal).
3. From WebCenter Portal menu on top, Select Logs -> View Log Messages to load the Log Messages page.
4. Select the criteria from the form, for e.g., set Date Range to 5 hours and click on “Search”.
5. Select any row in the table showing all log entries to load the details in the bottom preview pane.

   Optionally click on the Log File name to refine more on logs from the log file selected.

Monitor Logs from Physical Machine

Logs related to WebCenter Portal are recorded in a log file with name ***<<Managed_Server_Name>>.log*** and ***<<Managed_Server_Name>>-diagnostics.log*** under domain home. To access this logs directly from the physical machine:

1. Login to the server on which the managed servers are running. Make sure the user has permissions to the domain home.
2. Change directory to <<Domain_Home>>/servers.
3. To access Portal logs, go to WC_Portal/logs folder and to access OUCSS Inbound Services application logs, go to respective <<ManagedServerName>>/logs.

   If, for example, the domain home is /u01/oracle/product/webcenter/user_projects/domains/portal_domain, then:

   - Portal Logs (WC_Portal.log and WC_Portal-diagnostics.log) will be found under : /u01/oracle/product/webcenter/user_projects/domains/portal_domain/servers/WC_Portal/logs.

Monitoring Document References

- Monitor Oracle Fusion Middleware - https://docs.oracle.com/middleware/1221/core/ASADM/monitor.htm#ASADM206

Monitoring Oracle Utilities Customer Care and Billing

Oracle Utilities Customer Care and Billing Error Logs

Errors related to the CCB services are stored in the CCB_ENVIRONMENT_NAME/logs/system folder (e.g., V231_CCB_PERF_LIN_ORA_WLS/logs/system).

Communications to the Oracle Utilities Customer Care and Billing system is done via IWS. User will find all incoming requests and responses in IWS.trc file.

Note: For more information about errors and notifications see the Oracle Utilities Customer Care and Billing documentation.

Troubleshooting OUCSS (Taskflows-based Solution)

Symptom	Possible Cause	Corrective Action	Comments/Reference
Unable to Login as WSSAdmin	The password of WSSAdmin could be changed.	Login to WebLogic console as WLS Administrator and change the password of WSSAdmin	
WSSAdmin unable to see or acess OUCSS Portals	Tier-1/Group Space Security associated with WSSAdmin user might not be proper.	Login to WebLogic console as WLS Administrator and Check the out-of-box Enterprise groups are available in LDAP. Check WSSAdmin is member of WSSAdminGroup	
When logging to WebCenter Portal, user see Portal is Unavailable error page. The Portal is currently offline. Only the Portal moderator can access the Portal.	Portal(s) are offline for maintenance.	Login to WebCenter Spaces as WebCenter Administrator and bring the Portal online	See http://docs.oracle.com/middleware/12211/wcp/build/GUID-5671507F-11C3-4708-8B88-5E81091502A5.htm#GUID-A83645DF-B03A-4F80-B89B-D7A9C22B385E
When logging to WebCenter Portal user is informed that Portal is no longer available The Portal may have been deleted or renamed.	OUCSS Portal might be deleted/closed or renamed.	Login to WebCenter Portal as WebCenter Administrator and go to Administrator -> Portals page. Make sure OUSS Admin, OUSS Portal and Selft Service portals are listed and active. If the Portal is available but not active, make it active. If the Portal is not seen in the list, then it needs to be imported again.	

WSSAdmin unable to see Admin menu	Tier-1 Portal Security associated with WSSAdmin user might not be proper.	Login to WebLogic console as WLS Administrator and check that the out-of-box Groups are available in LDAP. Also check that WSSAdmin is a member of WSSAdminGroup	Check OUCSS Implementation Guide (OUCSS Security) section to understand more on Security setup of OUCSS Portal.
WSSCSR Admin is not able to see Account Detail taskflow.	Tier-1 Portal security associated with WSSCSR group is not proper	Login to WebLogic console or LDAP as Administrator and check the following: The out-of-box OUCSS Enterprise Groups are available in LDAP; WSSCSR is a member of WSSCSRGroup.	Check OUCSS Implementation Guide (OUCSS Security) section to understand more on Security setup of OUCSS Portal.
Taskflows in Portal pages fail to load with 'Unable to process your request at this time. Please try again later or contact the administrator." error.	Any Web Service related error causes this. Web Service connection to CCB, BPEL or NMS is either down or not configured properly.	Check if the CCB and NMS services are Up and running. Login to Oracle EM as WLS Administrator and go to the ADF Connections page of Portal application. Check all the connections are configured properly and pointing to right server. Check that the CSF Keys are created and have right credentials.	Sometimes, either the connections are not configured properly or configured to a wrong server (e.g., test CCB instance instead of production CCB servers). Enable oucss.debug.enable property to true from OUCSS Admin > Configuration Options screen to show detailed debug message related to the cause.
After logging to OUCSS Portal and clicking on "Administrator" link on the top user see UnAuthorized.	Administrator page is secured and only accessible by Administrator or members of WSSAdminGroup.	Login to OUCSS Portal as Administrator or WSSAdmin.	This is expected behavior for non-admin users.
Offers (Promotion or Rates) taskflows do not render any data or throw exception.	Either the offer service is not returning any rows for the Offer Set and Locale. Or Offer Web Service connection is not configured properly.	Test the service to make sure values are returned for the given Offer Set Code and Locale. For Banner Promotion make sure the image used is accessible. Login to Oracle EM and load the ADF connections page of OUCSS Portal application. Check the Offer Service connection has right configuration.	Check the "Offers and Promotions" section in OUCSS Implementation guide to know more about Offers.
Service Management taskflows not showing up properly.	Check the logs, if the trains are not defined properly, you will see exception related to it.	Login to OUCSS Portal as WSSAdmin and go to Admin -> Train and check the trains configured for the page that is throwing exception.	

Outage Map does not show up	Either the MapViewer server is down or the connection is not configured properly	Start the MapViewer managed server if it's not running or check the MapViewer URL configured in EM.
Portal does not load.	Page template configured as default does not exist or throws exception.	Check if the page template configured in SS_Resource exists and loads without error.
On clicking a link (not configured to home page or logout page), home page or page configured as logout page loads.	The page of the link does not exist or user does not have permission to view the page.	Check for type in link URL or if the page exists. Check if the logged user has access to the link.

Monitoring and Troubleshooting Integrated and Direct BPEL Flows

This section describes how to:

- Monitor from Oracle Utilities Customer Care and Billing
- Monitor from Oracle Utilities Meter Data Management
- Monitor from Oracle Utilities Network Management System
- Monitor from the Integration Layer
- Troubleshooting

Monitoring from Oracle Utilities Customer Care and Billing

Oracle Utilities Customer Care and Billing Error Logs

Errors related to the online integration invocation from CCB are stored in the CCB_ENVIRONMENT_NAME/logs/system folder (e.g., V231_CCB_PERF_BLD10_LIN_ORA_WLS/logs/system).

Note: For more information about errors and notifications see the Oracle Utilities Customer Care and Billing documentation.

Notifications for CCB-MDM Integrated Flows

Errors encountered in the Integration are communicated back to the Initiating application.

When Oracle Utilities Customer Care and Billing sends a request message out to Oracle Meter Data Management (MDM), it expects a response back.

When integration encounters an exception while processing the message or MDM sends an exception or fault back to the integration, integration will return a SOAP fault back to CCB. This will cause the outbound message to go to error status.

Connection Errors

Information can be found in the log file described above.

Monitoring from Oracle Utilities Meter Data Management

Errors related to the online integration invocation from Oracle Utilities Meter Data Management (MDM) are stored in the MDM_ENVIRONMENT_NAME/logs/system folder.

For example: V201_MDM_LIN_ORA_WLS/logs/system

Monitoring from Oracle Utilities Network Management System

Errors, which occur during execution of PL/SQL package, are reported to the integration layer. They are not logged within Oracle Utilities Network Management System.

Generic IVR Adapter has dedicated log file where errors are recorded. Name of the log file typically begins with 'IVRAdapter'.

For more information about troubleshooting Oracle Utilities Network Management System see the "Troubleshooting and Support" chapter in the Oracle Utilities Network Management System Configuration Guide.

Monitoring from the Integration

You can monitor from the integration by either:

- Monitoring the composite instances using WebLogic SOA Enterprise Manager
- Monitoring the WebLogic logs

Monitoring From WebLogic SOA Enterprise Manager

Check Process Instance

1. Login into the WebLogic SOA server Enterprise Manager
2. In the left menu navigate to the following partition:
   - For CCB-MDM Integrated Flows: SOA > soa-infra > CCB2-MDM2
   - For CSS Direct Flows: SOA > soa-infra > OUCSS
3. All the composite processes deployed for the CCB-MDM for Self Service Integration are available under the CCB2-MDM2 partition.
4. All the composite processes deployed for the CSS Direct Integration are available under the OUCSS partition .
5. Select the appropriate process to list all the instances for the processes sorted by time of execution.

6 The instances also have the request ID as part of the display name.

7 Click the appropriate process instance and it will display the flow for the process.

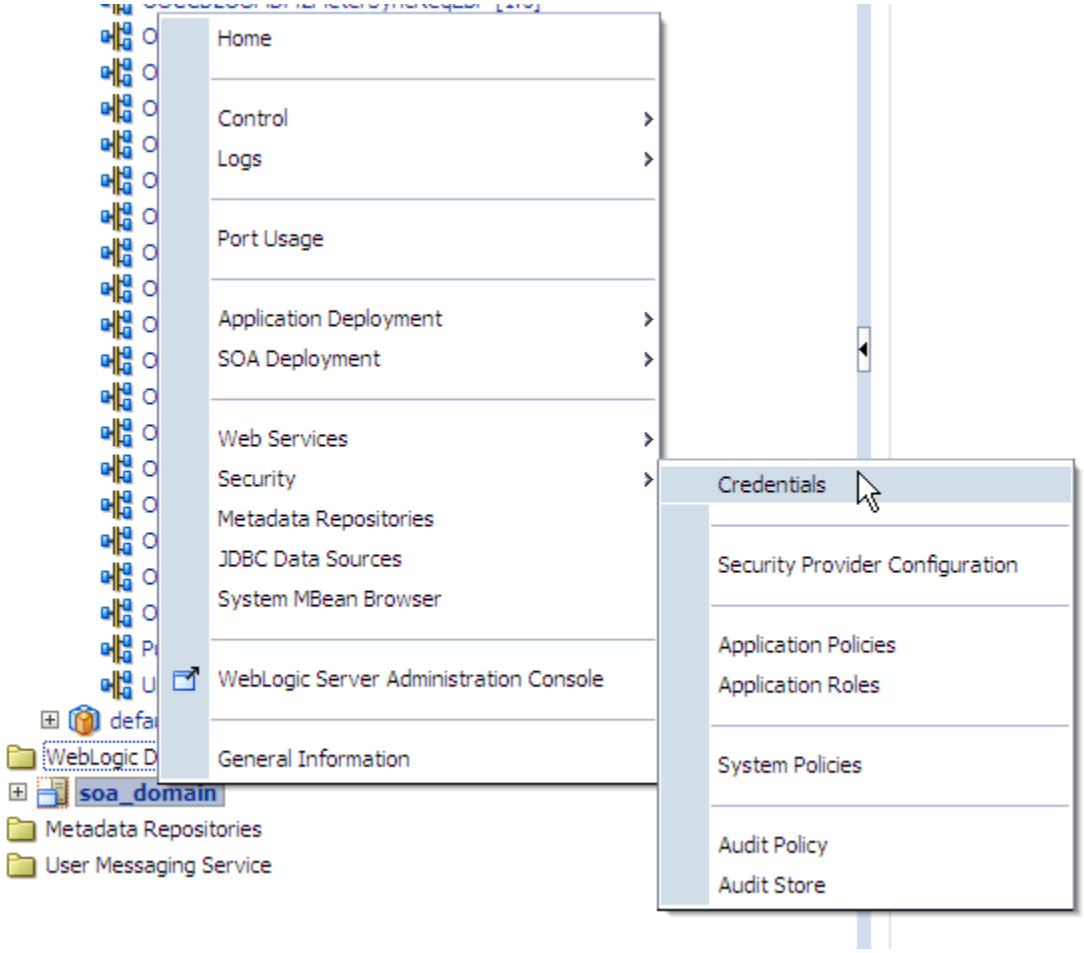
The composite flow lists all the activities in the process instance.

Check Credential Key (csf-key) Attached to Policy

1 Login into the WebLogic SOA server Enterprise Manager

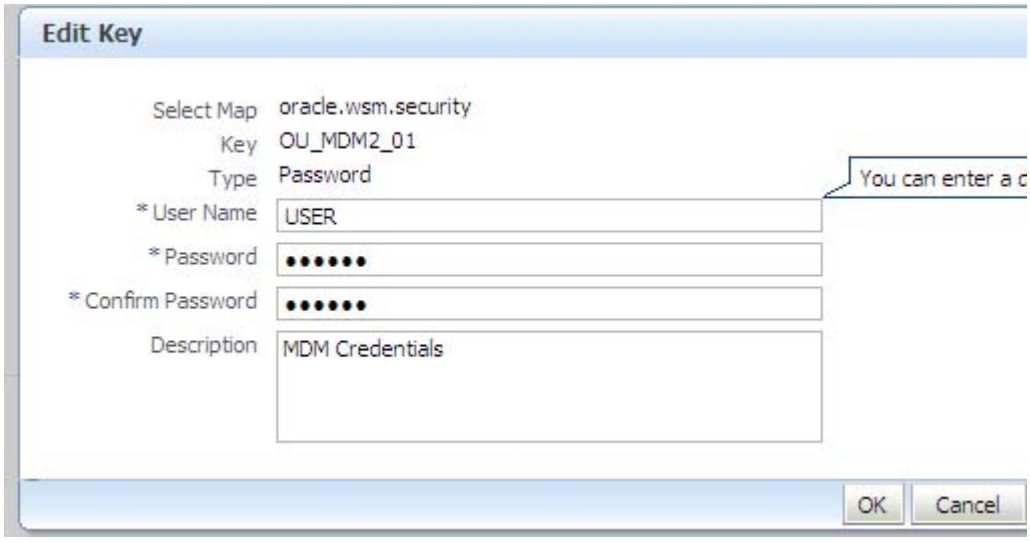
2 In the left menu navigate to WebLogic Domain > soa_domain.

3 Right click on soa_domain, click Security, and then click Credentials.



4 Expand oracle.wsm.security and OU_MDM2_01 and OU_CCB_01 keys should be defined there.

5 Edit the OU_MDM2_01 key and check that the User Name and Password defined are correct. This key is used to login to MDM application.



6 Edit the OU_CCB_01 key and check that the User Name and Password defined are correct. This key is used to login to CCB application.

Steps to Follow to Check the WebLogic Logs

1 Login into the machine where the SOA Server is installed.

2 The SOA logs are stored in: <WebLogic installation folder>/user_projects/domains/<SOA Domain name>/servers/<SOA Server name>/logs

For example: /slot/ems1234/oracle/Middleware/user_projects/domains/soa_domain/servers/soa_server1/logs

Data Purge

To maintain maximum system integrity the Oracle Fusion Middleware database should be purged periodically. Refer to note 815896.1 on support.oracle.com for information on how to complete this task.

Troubleshooting

At times, the integration processes might experience errors or issues with connection, processing, or sending or receiving messages. Following are the common scenarios which help you to troubleshoot error, if any, and find possible solutions.

Error 1: Source application sends out a message but the message does not reach the integration service. No instances found in SOA Enterprise Manager.

To resolve this error:

- **If the source application is CCB:**
  - Check the CCB logs to see if any errors are encountered while trying to send the message out. Refer to Oracle Utilities Customer Care and Billing Error Logs for more information on where to find the logs.
  - Check CCB's IWS Configuration to ensure they are configured correctly. Refer to the Setting Up Oracle Utilities Customer Care and Billing – IWS Configuration for more information.
- **If the source application is CSS:**
  - Check the CSS logs to see if any errors are encountered while trying to send the message out. Refer to Monitoring Oracle Utilities Self Service Section for more information.
  - Check if the BPEL processes are running. Refer to the for more information.
  - If WebLogic SOA Enterprise Manager is not accessible or the BPEL processes cannot be seen found in the WebLogic SOA Enterprise Manger, restart the SOA managed server.
  - If WebLogic SOA Enterprise Manager is accessible but the BPEL process is not active, activate or start up the process from the WebLogic SOA Enterprise Manager.

Error 2: Source Application sends out a request message but the message does not reach the target application or encountered an error while processing in the target application.

To resolve this error:

- Check the instance of the BPEL process ran, check if the message has faulted or encountered an error. Refer to the "Check Process Instance" section of the Monitoring From WebLogic SOA Enterprise Manager topic for more information.
- From WebLogic SOA Enterprise Manager, check the appropriate process instance flow trace to see the error details.
- Check the fault message coming from the target application and resolve the issue.
- Check the logs. Refer to Steps to Follow to Check the WebLogic Logs for more information.
- For CSS-MDM Integrated Flows

If the error encountered by the BPEL process is a runtime error stating that the CCB or MDM endpoint URL is not accessible, check the following:

- Check that the Target application is up.
- Check if the CCB or MDM web service called by the BPEL process has a policy and csf-key attached to it has the corrrect user and password. Refer to Check Credential Key for more information.
- In the Configuration Properties file, make sure the Target Application's web service endpoint URL is pointing to the correct URL.
- For CCB-MDM Integrated Flows
- If the error encountered by the BPEL process is a runtime error stating that the MDM endpoint URL is not accessible, check the following:
- Check that the MDM application is up.

- Check if the MDM web service called by the BPEL process has a policy and csf-key attached to it has the corrrect user and password. Refer to Check Credential Key for more information.
- Make sure in the MDM wsdl in MDS, the service's address location is pointing to the correct MDM URL.

Sample:

```
<wsdl:service name="WX-GetUsageOverviewService">
   <wsdl:documentation>WX-GetUsageOverview version 3: Get Usage
Overview</wsdl:documentation>
   <wsdl:port name="WX-GetUsageOverviewPort" binding="IWSxsd:WX-
GetUsageOverviewSoapBinding">
      <soap:address location="https://mdm server:9999/ouaf/Webservices/WX-
GetUsageOverview"/>
   </wsdl:port>
</wsdl:service>
```

Note: The MDM wsdl is in $PRODUCT_HOME/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/wsdl

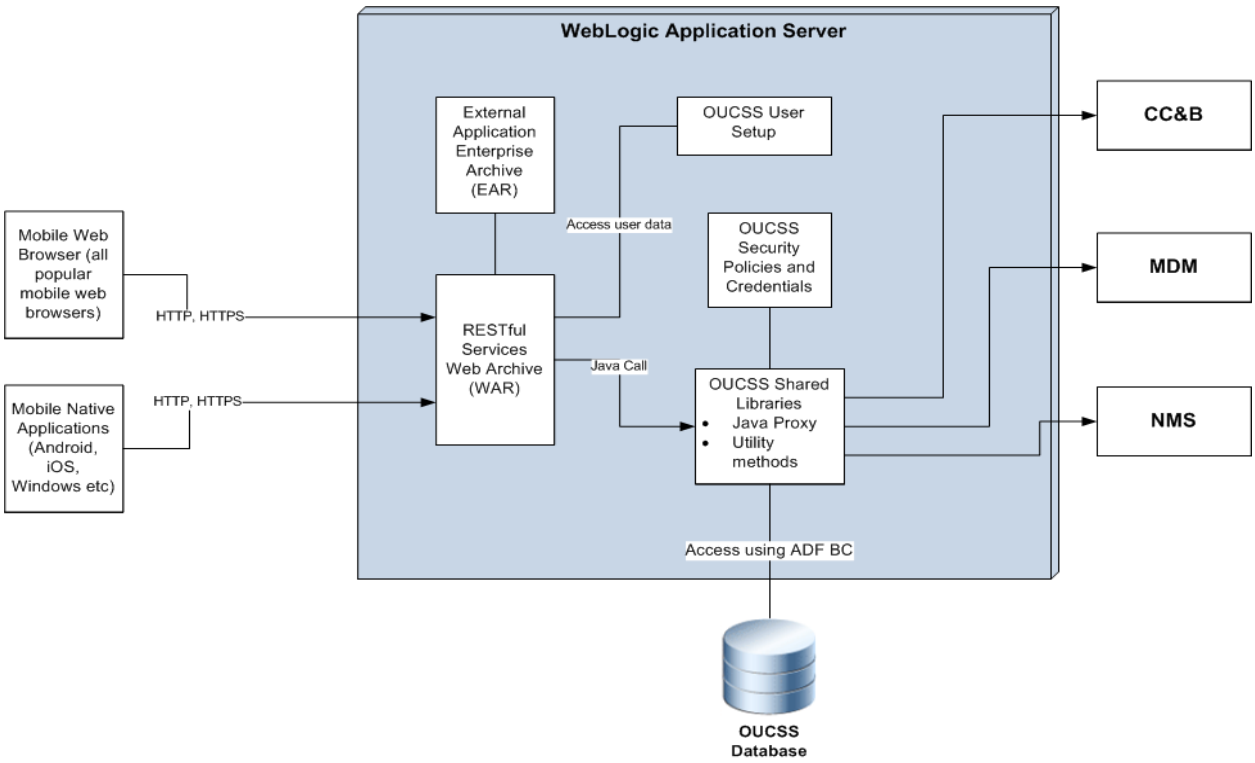
Chapter 11

REST Services

Overview

OUCSS REST Services are Restful (Representation State Transfer) Services created to be consumed by the Mobile applications and provide the OUCSS features on a Mobile platform. Most of the REST Services in turn utilize the SOAP IWS/BPEL services to retrieve data from the Edge Applications. A web service proxy is created for each of the SOAP IWS/BPEL web service. A few REST Services are created to retrieve data from OUCSS Admin database using ADF BC.

OUCSS REST services uses Jersey JAX-RS and Jackson libraries (part of jersey_bundle 1.9 shared library) shipped along with WebLogic Install.OUCSS REST Services produce either JSON or XML output based on the media type set for “Accept” header of the HTTP request.



Deployment

OUCSS REST Services are deployed as OUCSSRest.war as part of OUCSSInboundServices.ear. Connections to the edge applications are defined in the connections.xml. The web service connection names are as defined in the Core Portal application. This application refers to the following artifacts.

Artifact Name	Description	Type
com.oracle.ugbu.ss.lib	com.oracle.ugbu.ss.lib is OUCSS core shared library. Common functionality provided by this library is utilized. This artifact point to physical artifact as <PRODUCT_HOME>/Install/application/OUCSS_Extension.war	Library
extend.oucss.portal	This library contains any extensions and customizations implemented by the user. All customizations and extensions will be part OUCSSRestExtensionLibrary project of ExtendOUCSSPortal project	Library
com.oracle.ugbu.ss.rest.lib	This library contains the core implementations for REST Services. This library is part of <PRODUCT_HOME>/Install/application/ OUCSS_Rest_Extension.war This library contains jars for the Web Proxies -- representing the web services. Each web service has a corresponding web proxy jar REST Service resources - java beans for each REST service to process HTTP operations. Shared – API's and common implementations for REST Services oracle.ugbu.ss.rest.api.jar,oracle.ugbu.ss.rest.shared.jar	Library
jax-rs-2.0	jax-rs-2.0 which is deployed as part of installation process	Library

OUCSS REST Services

Base Services

Base Services which communicates with the Edge Applications to retrieve information for the core functionality. These services can be accessed as

```
Eg: http://<<host>>:<<port>>/<<context>>/api/<<REST ServiceName>>
```

OUCSS REST Service Name	REST Service Bean class (Package:com.oracle.ugbu.ss.rest.model.service.*)	Web Proxy jar	OUCSS WebService Connection Name Module Name
InvitePersonListService	InvitePersonListServiceBean	oracle.ugbu.ss.base.invitepersonlist.proxy.jar	SSInvitePersonList Login
VerifyAccountService	VerifyAccountServiceBean	oracle.ugbu.ss.base.verifyaccount.proxy.jar	SSVerifyAccount Login
ViewAccountService	ViewAccountServiceBean	oracle.ugbu.ss.viewaccount.proxy.jar	SSViewAccountService Login
SSViewAccountList	SSViewAccountListBean	oracle.ugbu.ss.base.accountlist.proxy.jar	SSViewAccountList Login
AccountSummaryService	AccountSummaryServiceBean	oracle.ugbu.ss.billing.account.summary.proxy.jar	AccountSummaryService AccountSummary
SSAutoPaySetupUpdateService	AutoPaySetupServiceBean	oracle.ugbu.ss.billing.autopaysetup.proxy.jar	SSAutoPaySetupUpdateService AccountManagement, AutoPay
BillingHistoryService	BillingHistoryServiceBean	oracle.ugbu.ss.billing.billinghistory.proxy.jar	BillingHistoryService BillingHistory
ElectronicBillingService	ElectronicBillingServiceBean	oracle.ugbu.ss.billing.electronic.proxy.jar	WSSEBillUpdateService AccountManagement Electronic Billing
MailingAddressInfoService	MailingAddressInfoServiceBean	oracle.ugbu.ss.base.mailingaddressinfo.proxy.jar	SSMaintainMailingAddressUpdateService AccountManagement, Account AddressInfo
PhoneInfoService	PhoneInfoServiceBean	oracle.ugbu.ss.base.phoneinfo.proxy.jar	WSSAccountPhoneInfoUpdateService AccountManagement, AccountPhoneInfo
ViewBillService	ViewBillServiceBean	oracle.ugbu.ss.billing.viewbill.proxy.jar	ViewBillService BillingHistory

ScalarMeterService	ScalarMeterServiceBean	oracle.ugbu.ss.customer.scalarmeter.proxy.jar	SSCreateScalarMeterAddService Scalar Meter
AlertsService	AlertsServiceBean	oracle.ugbu.ss.base.alerts.proxy.jar	AlertsService Alerts
ConsumptionSummaryService	ConsumptionSummaryServiceBean	oracle.ugbu.ss.customer.consumptionsummary.proxy.jar	SSConsumptionSummaryService ConsumptionSummary
GetRatedSAsService	GetRatedSAsServiceBean	oracle.ugbu.ss.billing.getratedsas.proxy.jar	SSGetRatedSAsService RateAnalysis
RateAnalysisService	RateAnalysisServiceBean	oracle.ugbu.ss.billing.rateanalysis.proxy.jar	SSRateAnalysisService RateAnalysis
OneTimePaymentService	OneTimePaymentServiceBean	oracle.ugbu.ss.payment.onetime.proxy.jar	SSOneTimePaymentService Payment/OneTime
BillNotificationUpdateService	BillingNotificationServiceBean	oracle.ugbu.ss.billing.billingnotification.proxy.jar	BillNotificationUpdateService BillNotification
ServiceChargesToDateService	ServiceChargesToDateServiceBean	oracle.ugbu.ss.customer.servicecharges.proxy.jar	SSServiceChargesToDate ServiceChargeToDate
UsageDetailsService	UsageDetailsServiceBean	oracle.ugbu.ss.usage.details.proxy.jar	UsageDetailService UsageDetail
SSUsageOverviewService	UsageOverviewServiceBean	oracle.ugbu.ss.customer.usageoverview.proxy.jar	SSUsageOverviewService UsageOverview
ServiceManagementService	ServiceManagementServiceBean	oracle.ugbu.ss.customer.servicemgmt.proxy.jar	CustomerMgmtService ServiceMgmt
PremiseSearchService	PremiseSearchServiceBean	oracle.ugbu.ss.customer.service.premisesearch.proxy.jar	PremiseSearchService ServiceMgmt
PayArrangementService	PayArrangementServiceBean	oracle.ugbu.ss.payment.payarrangement.proxy.jar	PaymentArrangement PayArrangement
GreenButtonService	GreenButtonServiceBean	oracle.ugbu.ss.customer.greenbutton.proxy.jar	GreenButtonService Usage Download
LookUpService	LookUpServiceBean	oracle.ugbu.ss.admin.getselfservicedropdowns.proxy.jar	SSLookupService Admin/Lookup
LabelService	LabelServiceBean	oracle.ugbu.ss.admin.selfservicelabels.proxy.jar	SSLabelService Admin/Labels
OutageSummaryService	OutageSummaryServiceBean	oracle.ugbu.ss.outage.summary.proxy.jar	OutageSummaryService Outage Map (Outage Map and Outage Table screens)
OutageTroubleCallService	OutageTroubleCallServiceBean	oracle.ugbu.ss.outage.troublecall.proxy.jar	SSTroubleCallService SSReportOutage

OutageScreenInfoService	OutageScreenInfoServiceBean	oracle.ugbu.ss.outage.screen.info.proxy.jar	SSAccountInfoTroubleCodesService SSReportOutage
OutageDetailService	OutageDetailServiceBean	oracle.ugbu.ss.outage.detail.proxy.jar	OutageMyDetailService OutageMap (My Outage Details screen)
OutagePublicDetailService	OutagePublicDetailServiceBean	oracle.ugbu.ss.outage.public.detail.proxy.jar	OutagePublicDetailService OutageMap
ContextInfoService	ContextInfoServiceBean	oracle.ugbu.ss.base.contextinfo.proxy.jar	SSContextInfoService AccountContextInfo
AccountSearchService	AccountSearchServiceBean	oracle.ugbu.ss.base.accountsearch.proxy.jar	SSAccountSearchService CSRAccountSearch
SSBudgetDetailService	BudgetDetailsServiceBean	oracle.ugbu.ss.billing.budgetdetails.proxy.jar	SSBudgetDetailServicBudgetBilling
SSBudgetRequestUpdateService	BudgetRequestServiceBean	oracle.ugbu.ss.billing.budgetrequest.proxy.jar	SSBudgetRequestUpdateService BudgetBilling
SSFinancialHistory	FinancialHistoryServiceBean	oracle.ugbu.ss.billing.financialhistory.proxy.jar	SSFinancialHistory Financial History
PPBBalanceAndChargesService	PPBBalanceAndChargesServiceBean	oracle.ugbu.ss.billing.prepaid.balanceandcharges.proxy.jar	PrepaidBalanceAndChargesService PrePaid
PPBEstimatesAndCostsService	PPBEstimatesAndCostsServiceBean	oracle.ugbu.ss.billing.prepaid.estimatesandcost.proxy.jar	PrepaidEstimatesAndCostService PrePaid
ScalarUsageDetailService	ScalarUsageDetailServiceBean	oracle.ugbu.ss.customer.scalarusagedetail.proxy.jar	ScalarUsageDetailService Scalar Usage Details
UploadProcessService	UploadProcessServiceBean	oracle.ugbu.ss.uploadprocessservice.proxy.jar	OUCSSUploadProcessService FormsMgmt
ReadDocumentService	ReadDocumentServiceBean	oracle.ugbu.ss.readprocessservice.proxy.jar	OUCSSReadProcessService FormsMgmt
FormsListService	FormsListServiceBean	oracle.ugbu.ss.customer.formslist.proxy.jar	SSFormsListService FormsMgmt
FormsManagementService	FormsManagementServiceBean	oracle.ugbu.ss.customer.formsmgmt.proxy.jar	SSFormsManagementUpdateService FormsMgmt
SSRetrieveAccountDocumentsService	RetrieveAccountDocumentsService	oracle.ugbu.ss.retrieveaccountdocuments.proxy.jar	WXRetrieveAccountDocumentsService SSReadAccountDocuments
SSReadAccountDocumentsService	ReadAccountDocumentsService	oracle.ugbu.ss.readaccountdocuments.proxy.jar	OUCSSReadDocumentService SSReadAccountDocuments
NotificationPreferencesService	NotificationPreferencesServiceBean	oracle.ugbu.ss.setnotificationprefere	SSNotifPreferences

		nces.proxy.jar	Notification Preferences
SSDeliveryChannelsService	DeliveryChannelsServiceBean	oracle.ugbu.ss.deliverychannels.proxy.jar	SSDeliveryChannelsService SSDeliveryChannels
SSNotificationListService	NotificationListServiceBean	oracle.ugbu.ss.notificationlist.proxy.jar	SSNotificationListService Notification List
UserLinkService	UserLinkServiceBean	oracle.ugbu.ss.base.userlink.proxy.jar	UserLinkService
UnEnrollAccountService	UnEnrollAccountServiceBean	oracle.ugbu.ss.base.unenrollaccount.proxy.jar	UnEnrollAccountService
LangTimeZoneService	LangTimeZoneServiceBean	oracle.ugbu.ss.base.langtz.proxy.jar	LangTimeZoneService
UsageChargesProjectedService	UsageChargesProjectedServiceBean	oracle.ugbu.ss.billing.usagechargesprojected.proxy.jar	WXUsageChargesProjectedService
CancelSchedPayService	CancelSchedPayServiceBean	oracle.ugbu.ss.payment.cancelschedpay.proxy.jar	SSCancelSchedPay
AccountListService	AccountListServiceBean	oracle.ugbu.ss.base.accountlist.proxy.jar	SSViewAccountList
Commercial Web Services (Web Services that are exclusive to commercial context)			
SSCommercialBillHistoryService	SSCommercialBillHistoryServiceBean	oracle.ugbu.ss.commercial.bill.history.proxy.jar	CommercialBillHistoryService SSCommercialBillHistory
SSCommercialUsageComparisonService	SSCommercialUsageComparisonServiceBean	oracle.ugbu.ss.commercial.usage.comparison.proxy.jar	SSCommercialUsageComparisonService SSCommercialUsageComparisonService
SSCommercialUsageAggregationService	UsageAggregationServiceBean	oracle.ugbu.ss.usage.aggregation.proxy.jar	SSCommercialUsageAggregationService SSCommercialUsageAggregationService

REST Services to Access Data from OUCSS Admin Tables

These services consumes input as parameters specified in the query string. The URL to these services is of the form

```
http://<<host>>:<<port>>/OUCSSRest/api/cache/<<REST Service Name>>?searchCode=
```

REST Service Name	Description	Sample Input to the REST Service
OUCSSAccessCache	Retrieves data from Access tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/OUCSSAccessCache searchCode=ACCOUNT_HOLDER
OUCSSLOBCache	Retrieves data from Line of Business tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSLOBCache searchCode=RESIDENTIAL
OUCSSLanguageCache	Retrieves data from the Language tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSLanguageCache searchCode=en
OUCSSLookupCache	Retrieves data from Lookup tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSLookupCache searchCode=PHONE_TYPES
OUCSSPortletAccessCache	Retrieves data from Portlet Access tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSPortletAccessCache searchCode= ACCOUNT_HOLDER
OUCSSResourceCache	Retrieves data from Resource tables for a given resource_cd	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSResourceCache searchCode= PLANNED_OUTAGE_URL
OUCSSSystemPropertiesCache	Retrieves data from System Properties tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSSystemPropertiesCache searchCode= default.enroll.role
OUCSSTrainCache	Retrieves data from Train tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSTrainCache searchCode= START_SERVICE
OUCSSLabelCache	Retrieves data from Label tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/ OUCSSLabelCache searchCode= ADDRESS_CITY_LBL
OUCSSMessageCache	Retrieves data from Message tables	http://<<host>>:<<port>>/OUCSSRest/api/cache/OUCSSAccessCache searchCode= INVALID_USER_ERROR

REST Services to Retrieve Accounts for a User Per Line of Business

This REST Service retrieves accounts for a particular user for given Line of Business. User Id is the logged in user retrieved from security context.

The URL to User Services is of the form

```
http://host:port/<<context>>/api/user/OUCSSUserService?lob=<<value>>&retrieveAccountInfo=
<<booleanValue>>
E.g:
http://host:port/<<context>>/api/user/OUCSSUserService?lob=1&retrieveAccountInfo=true
```

REST Service Name	Description	Default Value
lob	LOB Context to be used to retrieve account from DB	1 (Residential)

retrieveAccountInfo	Flag to control calling Web Service to retrieve account information from CCB.	false

REST Services to Register

This REST service registers the user into OUCSS Portal application, creating an entry in default Weblogic authentication provider and OUCSS SS_USER table.

Use the following URL to register the service:

```
http://host:port/OUCSSRest/api/OUCSSRegisterService
```

REST Service Name	Description	Default Value
userId	User Id	none
emailId	Email Address	none
firstName	First name	none
lastName	Last name	none
password	Password (Out of box implementation of processes password sent in Base64 encoded form)	none

Note: All fields are required.

Sample JSON Request

```
{"userId":"jdoe","emailId":"sample@test.com,"firstName":"John","lastName":"Doe","password
":"dGVzdA=="}
```

Sample XML Request:

```
<oucssRegister>
    <userId>jdoe</userId>
    <emailId>sample@test.com</emailId>
    <firstName>John</firstName>
    <lastName>Doe</lastName>
    <password>dGVzdA==</password>
</oucssRegister>
```

REST Services to Enroll

This REST service enrolls the account to a user. Enroll REST Service does not perform account verification.VerifyAccountService REST service should be used for account verification.

Use the following URL to enroll the service:

```
http://host:port/OUCSSRest/api/OUCSSEnrollService
```

REST Service Name	Description	Default Value
lobCd	Line of Business code Eg: RESIDENTIAL	none
key1	Account ID	none
key2	Person ID	none
userId	User Id	none

Note: lobCd, key1, and userId are required fields.

Sample JSON Request:

```
{"lobCd":"RESIDENTIAL","key1":"111111111","key2":"2222222222","userId":"jdoe"}
```

Sample XML Request:

```
<oucssEnroll>
    <lobCd>RESIDENTIAL</lobCd>
    <key1>111111111</key1>
    <key2>2222222222</key2>
    <userId>jdoe</userId>
</oucssEnroll>
```

Supported Media Types

OUCSS REST Services accepts inputs in different media types in the "**Content-type**" header of the HTTP request. The media types supported are Form input (application/x-www-form-urlencoded), JSON(application/json) and XM(application/xml).Below table shows the media types supported for different REST Services.

REST Services	JSON	XML	Form Submit (Only on action= Read where applicable)
InvitePersonListService	✓	✓	✓
VerifyAccountService	✓	✓	×
ViewAccountService	✓	✓	×
SSViewAccountList	✓	✓	×
AccountSummaryService	✓	✓	✓
SSAutoPaySetupUpdateService	✓	✓	✓
BillingHistoryService	✓	✓	×
ElectronicBillingService	✓	✓	✓
MailingAddressInfoService	✓	✓	✓
PhoneInfoService	✓	✓	✓
ViewBillService	✓	✓	✓
ScalarMeterService	✓	✓	✓
AlertsService	✓	✓	✓
ConsumptionSummaryService	✓	✓	✓

GetRatedSAsService	✓	✓	✓
RateAnalysisService	✓	✓	×
OneTimePaymentService	✓	✓	✓
BillNotificationUpdateService	✓	✓	✓
ServiceChargesToDateService	✓	✓	✓
UsageDetailsService	✓	✓	✓
SSUsageOverviewService	✓	✓	✓
ServiceManagementService	✓	✓	×
PremiseSearchService	✓	✓	×
PayArrangementService	✓	✓	✓
GreenButtonService	✓	✓	×
LookUpService	✓	✓	×
LabelService	✓	✓	×
OutageSummaryService	✓	✓	✓
OutageTroubleCallService	✓	✓	×
OutageScreenInfoService	✓	✓	✓
OutageDetailService	✓	✓	✓
OutagePublicDetailService	✓	✓	✓
ContextInfoService	✓	✓	✓
AccountSearchService	✓	✓	×
SSBudgetDetailService	✓	✓	•
SSBudgetRequestUpdateService	✓	✓	✓
SSFinancialHistory	✓	✓	✓
PPBBalanceAndChargesService	✓	✓	✓
PPBEstimatesAndCostsService	✓	✓	✓
ScalarUsageDetailService	✓	✓	✓
UploadProcessService	✓	✓	×
ReadDocumentService	✓	✓	×
FormsListService	✓	✓	×
FormsManagementService	✓	✓	×
SSRetrieveAccountDocumentsService	✓	✓	×
SSReadAccountDocumentsService	✓	✓	×
NotificationPreferencesService	✓	✓	✓
SSDeliveryChannelsService	✓	✓	✓
SSNotificationListService	✓	✓	×
SSCommercialBillHistoryService	✓	✓	×
SSCommercialUsageComparisonService	✓	✓	×
SSCommercialUsageAggregationService	✓	✓	×
UserLinkService	✓	✓	×
UnEnrollAccountService	✓	✓	✓

LangTimeZoneService	✓	✓	✓
UsageChargesProjectedService	✓	✓	✓
CancelSchedPayService	✓	✓	✓
OUCSSAccessCache	×	×	✓
OUCSSLOBCache	×	×	✓
OUCSSLanguageCache	×	×	✓
OUCSSLookupCache	×	×	✓
OUCSSPortletAccessCache	×	×	✓
OUCSSResourceCache	×	×	✓
OUCSSSystemPropertiesCache	×	×	✓
OUCSSTrainCache	×	×	✓
OUCSSLabelCache	×	×	✓
OUCSSMessageCache	×	×	✓
OUCSSEnrollService	✓	✓	×
OUCSSRegisterService	✓	✓	×

Sample Inputs to REST Services

Structure of the payload to the REST Services is based on the backend SOAP web services these RestServices in turn call. Input to the REST Services is based on the element names and structure of the backend SOAP wsdl. Please refer to the respective SOAP service WSDLs to know the structure and the element names.

Form Submit Using GET (HTTP Request Header Content-Type= application/x-www-form-urlencoded)

Account Id and Person ID must be provided as key1 and key2. All other parameters are element names (first character must be in lower case).

Example:

```
http://host:port/context/api/FinancialHistoryService?key1=111111111&key2=222222222
http://host:port/context/api/OutagePublicDetailService?areaType=CITY&area=UnionTown, OH
```

JSON Input Using POST (HTTP Request Header Content-Type= application/json)

Example: ElectronicBillingService

```
{
"head": {
"action": "UPDATE",
"key1": {
"name": "ACCT_ID",
"value": "1111111111"
},
"key2": {
"name": "PER_ID",
"value": "2222222222"
}
},
"mainData": {
"billRoutingInfo": {
"receivesCopyOfBill": "true",
"billRouteType": "EMAIL",
```

```
"emailAddress": "sample@test.com"
}
},
"custom": {}
}
```

XML Input Using POST (HTTP Request Header Content-Type= application/xml)

XML input requires that a target namespace be specified in the input.

Example: ElectronicBillingService

```
<WXEBillSetup xmlns="http://oracle.com/WXEBillSetup.xsd" dateTimeTagFormat="">
  <head>
    <action>UPDATE</action>
    <key1>
      <name>ACCT_ID</name>
      <value>9175218417</value>
    </key1>
    <key2>
      <name>PER_ID</name>
      <value>9907619695</value>
    </key2>
    <emailAddress />
    <webUserId />
    <ipAddress />
  </head>
  <mainData>
    <billRoutingInfo>
      <receivesCopyOfBill>true</receivesCopyOfBill>
      <billRouteType>EMAIL</billRouteType>
      <faxNumber />
      <emailAddress>sample@test.com</emailAddress>
    </billRoutingInfo>
  </mainData>
</WXEBillSetup>
```

Error Messages

This table describes various error codes and possible causes.

Exception Condition	Fault Type	HTTP Error Code
SOAP Fault from Web Service	faultCode:10001 faultSummary:Generic message faultDetail: if debug is on then faultString contains the SOAP Fault message	503 Internal Server Error
Error in <errorReference> tag	faultCode:error Number from errorReference faultSummary: Error message from errorReference faultDetail: Null faultCategory:error Category from errorReference	503 Internal Server Error
Service Not Found	faultCode:10001 faultSummary:Generic message faultDetail:Service <<ServiceName>>	404 (Not Found)

	not found.	
Invalid Request	faultCode:10001 faultSummary:Generic message faultDetail:Localized Message from Exception thrown while processing request.	400 (Bad Request)
Mime Type Not Supported E.g: Calling with GET and Content-Type: application/json.	faultCode:10001 faultSummary:Generic message faultDetail:Mime Type not supported.	415 (Media not supported)

Security

REST Services are secured using the `oracle/multi_token_rest_service_policy` policy. Refer to https://docs.oracle.com/middleware/1221/owsm/security/owsm-predefined-policies.htm#OWSMS5487 for more information on this policy.

One of the ways to authenticate is through Basic authentication.

Sample code snippet from the OUCSSMobileWebApp to call the REST Service by setting Basic Authentication using JQuery.

```
var auth = "Basic " + btoa(username + ":" + password);

var headers = { "Accept": "application/json; charset=utf-8", "Authorization" : auth};

varr ajaxObject= {
                url : restServerURL,
                type : HttpMethod,
                async : true,
                data : args,
                dataType:  'json',
                success : onSuccessMethod,
                error : onFailureMethod,
                headers: headers
                 };

if (ajaxArgs.contentType)
   ajaxObject["contentType"] = ajaxArgs.contentType;

 $.ajax(ajaxObject);
```

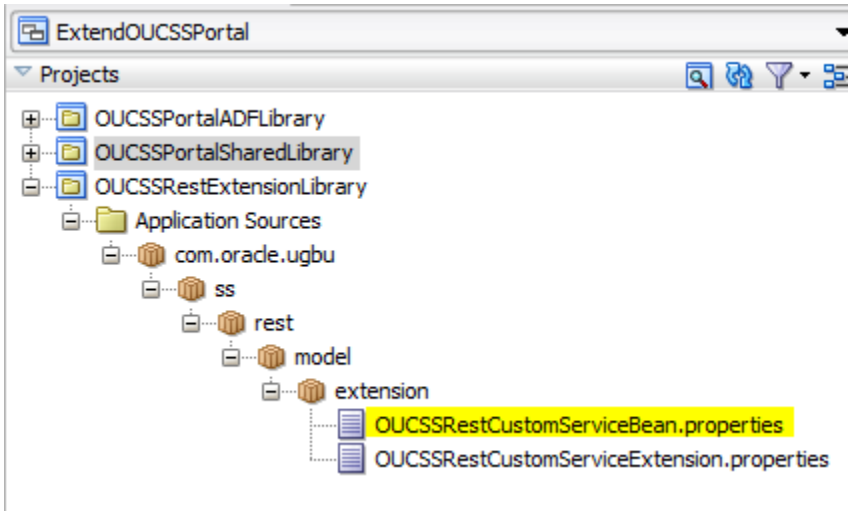
Customization

REST Services can be customized as follows:

- A Custom Service bean can be created to override the base Service Bean.
- A Custom Service bean can also implement the interface OUCSSRestWSProxyBean<T, R>.
- Mapping for the custom service should be added in the OUCSSRestCustomServiceBean.properties file under package com.oracle.ugbu.ss.rest.model.extension in the OUCSSRestExtensionLibrary.

Example:

```
FinancialHistoryService=com.oracle.ugbu.ss.rest.model.service.custom.FinancialHistoryServ
iceCustomizedBean
```



Note: OTB REST Service beans extend one of the three internal Abstract classes which implements OUCSSRestWSProxyBean Interface. Implementors must note that these are internal classes and the implementations may change in future releases.

The Abstract classes are:

- OUCSSAbstractWSProxyServiceBeanImpl<T,R> which is extended by all Base Services which are not Outage related, which in turn invoke SOAP web services to retrieve data from Edge applications.
- OUCSSAbstractOutageServiceBean which is extended by all Outage related REST services
- OUCSSAbstractADFBCServiceBeanImpl<T> which is extended by REST Services which retrieve data from OUCSS Admin tables.

Appendix A

Optional OUCSS-OUCCB BPEL Flows Web Services

The following base services invoked by Oracle Utilities Customer Self Service can be configured to use the OUCSS-OUCCB Optional BPEL service to connect to CCB.

Note: For more information on the base services invoked by Oracle Utilities Customer Self Service, see Chapter 2, "OUCSS Implementation", in the *Oracle Utilities Customer Self Service Implementation Guide*.

OUCSS Service Name Module Name	Service Description	CSS-CCB BPEL Service
SSInvitePersonList Login	It is responsible for retrieving person information related to an account.	OUCSSOUCCBWXInvitePersonListEBF
SSVerifyAccount Login	This service is used to enroll an account to a self-service user for web access. The list of verification fields per line of business are defined on the Self-Service Integration master configuration.	OUCSSOUCCBWXVerifyAccountEBF
SSViewAccountService Login	This service retrieves account information to display in the self-service application.	OUCSSOUCCBWXViewAccountEBF
SSViewAccountList Login	This service accepts a list of accounts from self-service and returns corresponding account information. It uses the information scripts defined on the Self-Service Integration master configuration. In addition, it may also invoke the account list filter script defined on the self-service	OUCSSOUCCBWXViewAccountListEBF

	master configuration to apply filter criteria to the account list.	
AccountSummaryService AccountSummary	This service retrieves account charge summary information to display in the self-service application. The service returns • The most recent bill for the account (excludes off cycle bill generated bills) • Account's current balance	OUCSSOUCCBWXAccountChargesSummaryRetrieverEBF
SSAutoPaySetupUpdateService AccountManagement, AutoPay	This service is used for retrieving and maintaining an account's auto pay details. For auto pay updates, the service merely creates an instance of the Auto Pay Setup business object defined on the Self-Service Integration master configuration.	OUCSSOUCCBWXAutoPaySetupEBF
BillingHistoryService BillingHistory	This service retrieves bills and payments for an account for a given period to display in the self-service application.	OUCSSOUCCBWXBillPayHistoryRetrieverEBF
WSSEBillUpdateService AccountManagement Electronic Billing	This service is responsible for retrieving and maintaining a customer's bill routing method, e.g., postal, email, etc. This service supports the following modes: • On READ action, it retrieves the account's current bill routing method setting • On UPDATE action, it updates the account's current bill routing method setting	OUCSSOUCCBWXEBillSetupEBF
SSMaintainMailingAddressUpdateService AccountManagement, Account AddressInfo	This service is used for retrieval and updating of customer's mailing address. This service supports the following modes: • On READ action, it retrieves the account's current mailing address • On UPDATE action, it updates the account's mailing address. Note that when a mailing address is updated, it is stored on the person correspondence information	OUCSSOUCCBWXMaintainMailingAddressInfoEBF
WSSAccountPhoneInfoUpdateService AccountManagement, AccountPhoneInfo	This service is responsible for retrieval and updating of customer's phone numbers. This service supports the following modes: • On READ action, it retrieves the customer's current phone information • On UPDATE action, it updates the customer's phone information	OUCSSOUCCBWXMaintainPhoneInfoEBF
ViewBillService BillingHistory	This service retrieves account's bill details to display in the self-service application.	OUCSSOUCCBWXBillViewEBF
SSCreateScalarMeterAddService Scalar Meter	It is responsible for retrieving and adding manual or scalar meter reads. When adding a new meter read, the service merely creates an instance of the Meter Read Creation business object defined on	OUCSSOUCCBWXCreateMeterReadEBF

	the Self-Service Integration master configuration.	
SSConsumptionSummaryService ConsumptionSummary	This service retrieves consumption information for display in the self-service application. It retrieves consumption information for service agreements that do not require MDM bill determinants.	OUCSSOUCCBWXGetConsumptionSummaryEBF
SSGetRatedSAsService RateAnalysis	This service retrieves the rated service agreements of an account. It also returns the valid rate schedules for every SA's SA Type.	OUCSSOUCCBWXGetRatedSAsEBF
SSRateAnalysisService RateAnalysis	This service receives an SA and a new rate schedule and does a comparison of the bill segments of the SA versus what the charges would have been if the SA was billed using the new rate schedule.	OUCSSOUCCBWXRateAnalysisEBF
SSOneTimePaymentService Payment/OneTime	This service is used for creation of online payments.	OUCSSOUCCBWXMakePaymentEBF
BillNotificationUpdateService BillNotification	This service is responsible for maintaining the billing notification preferences of a self-service user.	OUCSSOUCCBWXSetBillNotifyPreferenceEBF
SSServiceChargesToDate ServiceChargeToDate	This service retrieves the charges to date for a self-service user's account. The system only attempts to calculate unbilled charges to date for service agreements that require bill determinants from MDM.	OUCSSOUCCBWXUsageChargesToDateEBF
CustomerMgmtService ServiceMgmt	This inbound service is used to process start, stop and transfer service requests. The service supports the following modes: • On READ action, it invokes the start service criteria script defined on the self-service master configuration • On UPDATE action, the service merely creates an instance of the start service task type defined on the master configuration. The start, stop or transfer request is handled within the service tasks's lifecycle.	OUCSSOUCCBWXProcessStartStopRequestEBF
PremiseSearchService ServiceMgmt	This service is used to search for a premise. It invokes the existing CCB premise search page to search for a premise by the address field constituents.	OUCSSOUCCBWXPremiseSearchEBF
PaymentArrangement PayArrangement	This service is used to process a pay arrangement request. The service supports the following modes: • On READ action, it invokes the payment arrangement eligibility script defined on the self-service master configuration. The eligibility script is responsible for returning the list of available pay arrangement service task types.	OUCSSOUCCBWXProcessPayArrangementRequestEBF

	• On UPDATE action, the service merely creates an instance of the pay arrangement service task type selected by the end customer. The pay arrangement request is handled within the service tasks's lifecycle.	
SSLookupService Admin/Lookup	This service retrieves data for populating dropdown lists in the self-service application. The following is returned: • Valid credit card types as defined on the Self-Service Integration master configuration • Valid payment types as defined on the Self-Service Integration master configuration • Valid bill route types • Valid phone types	OUCSSOUCCBWXGetSelfServiceDropdownsEBF See **Note below
SSLabelService Admin/Labels	This service retrieves data for populating field labels in the self-service application.	OUCSSOUCCBWXGetSelfServiceLabelsEBF See **Note below
SSContextInfoService AccountContextInfo	This service accepts account id and person id and returns a list of the context properties for the account	OUCSSOUCCBWXContextInfoEBF
SSAccountSearchService CSRAccountSearch	This service accepts account information, together along with search information The web service will return the list of accounts as a response	OUCSSOUCCBWXAccountSearchEBF
SSBudgetDetailService BudgetBilling	The web service accepts account information The list of eligible SAs along with their recommended budget amounts will be sent back to CSS. Ineligible SAs will also be sent, identifying them as such	OUCSSOUCCBWXBudgetDetailsEBF
SSBudgetRequestUpdateService BudgetBilling	After sending an update request to the service, the service will cancel the account's budget billing plan.	OUCSSOUCCBWXProcessBudgetRequestEBF
SSFinancialHistory Financial History	The web service accepts account information and returns bills and payments. Under each bill row, more details related to the bill will be supplied (total adjustments and corrections as well as the individual bill segments reported on the bill). Also, transactions not yet reported on a bill will be displayed individually.	OUCSSOUCCBWXFinancialHistoryRetrieverEBF
PrepaidBalanceAndChargesService PrePaid	The web service is used for fetching the balance and charges associated with a particular prepaid account	OUCSSOUCCBWXRetrievePPBBalanceAndChargesEBF
PrepaidEstimatesAndCostService PrePaid	This web service is used for fetching the estimated number of days available for a particular prepaid customer before his	OUCSSOUCCBWXRetrievePPBEstimatesAndCostsEBF

	service is cut.	
ScalarUsageDetailService Scalar Usage Details	The web service accepts on read account information. It returns the list of service agreements with their corresponding scalar usage details measurement	OUCSSOUCCBWXRetrieveScalarUsageEBF
SSFormsListService FormsMgmt	This web service has one action which is READ, when this web service is invoked, the edge application will return the list of form entries for the FORMTYPE.	OUCSSOUCCBWXFormsListEBF
SSFormsManagementUpdateService FormsMgmt	On UPDATE this web service is passed with the updated Form details. If the transaction is successful a reference number is returned back	OUCSSOUCCBWXFormsManagementEBF
WXRetrieveAccountDocumentsService SSReadAccountDocuments	The web service returns a list of valid documents related to the given account and the description of the documents	OUCSSOUCCBWXRetrieveAccountDocumentsEBF
CommercialBillHistoryService SSCommercialBillHistory	This service accepts a list of accounts and returns the financial history for the accounts. The base financial history service is called. This web service is used for commercial context.	OUCSSOUCCBWXMultipleAccountFinancialHistoryEBF
SSCancelSchedPay OneTimePay	The web service cancels scheduled one-time payment	OUCSSOUCCBWXCancelSchedPayEBF
WXUsageChargesProjectedService Bill Charges Projection	This web service is responsible for getting projected service charges until next bill date and the average service charges for prior bills.	OUCSSOUCCBWXUsageChargesProjectedEBF
LangTimeZoneService LangTimeZonePref	This web service is responsible for maintaining a customer's preferred language and time zone. This service supports the following modes: - On Read action, it retrieves the person's language and time zone. - On Update action, it updates the person's language and time zone. Note that when a language and time zone is updated, it is stored on the person correspondence information.	OUCSSOUCCBWXMaintainPersonLanguageTimeZoneEBF
UserLinkService Login	This web service supports Enroll, Add, Update, and Remove actions. The actions reflect the actions of the account/person relationship. An account and person are either associated with each other or no longer associated with each other in CSS. The update action can indicate a change to person information such as email or a change to account/person information such as access role.	OUCSSOUCCBWXMaintainCSSUserAccountEBF
MailingAddressValidation	This web service is responsible for maintaining a customer's mailing address.	OUCSSOUCCBWXMaintainMailingAddressInfoEBF

AccountAddressInfo	This service supports the following modes: - On Read action, it retrieves the account's current mailing address - On Update action, it updates the account's mailing address. Note that when a mailing address is updated, it is stored on the person correspondence information accepts a list of accounts and returns the financial history for the accounts. The base financial history service is called. This web service is used for commercial context.	
SSCommPreferencesUpdate Notification Preferences	This service is used to retrieve preferences for a given account to a self-service user. The notification types and the preferences are retrieved dynamically from CCB . This is called when oucss.notification.owner configuration is set to CCB	OUCSSOUCCB WXMaintainCommPreferencesEBF

Appendix B

Configuration for OUCCB Edge Application Referenced Users

1. Create a navigation model that includes only navigation links to pages/modules supported by inbound web services in CCB 2.x.

   A new Navigation Model "OUCSS CCB 2.x Navigation Model" is created; note that modules *not* supported by CCB 2.x are not in this the navigation model.

2. Customize the following Portal pages to remove references to services and taskflows not supported in CCB 2.x.

3. Bundle the Navigation model as `OUCSS210x_CCB2xx_Customization.zip`.

4. Unzip the `OUCSS210x_CCB2xx_Customization.zip` to a folder on the machine running the OUCSS 2.2.0.x instance.

Note: Make sure the folder names in the path do *not* contain a spaces and start with a alphabetic character. Otherwise, MDS import might fail.

5. Run WLST from `<<MW_HOME>>/Oracle_WC1/common/bin/wlst.cmd` (Windows) or `wlst.sh` (Linux).

   A Connect to the admin server using the following command:

```
connect("weblogic","<<wlspassword>>","t3://<<AdminServerHost>>:<<AdminServerPort>>")
```

   B Import the Navigation Model:

```
importWebCenterResource(appName='OUCSSPortal#v2.2.0.x', server='<<PortalManagedServer>>',
fileName='<<UnzipFolder>>/MDS/OUCSS210_CCB231_Navigation_Model.ear',
resourceType='navigation')
```

   C Import Customization Metadata:

```
importMetadata(application='OUCSSPortal', server='WC_CustomPortal',
fromLocation='<<UnzipFolder>>/MDS/v210', docs='/**')
```

6. Update the **SS_RESOURCE** table and override the navigation model and landing page using the following commands and in accordance with the directory structure in the `transport.mar` file:

```
UPDATE ss_resource SET url_ovrd =
'/oracle/webcenter/siteresources/scopedMD/s8bba98ff_4cbb_40b8_beee_296c916a23ed/navigatio
n/gsr0d464579_408d_4f18_af81_fa93d05c737c/OUCSSNavigationModel.xml' WHERE resource_cd =
'PORTAL_NAV_RESIDENTIAL';

UPDATE ss_resource_l SET descr_ovrd='Override with CCB 2.3.1 Navigation Model.' WHERE
locale = 'en' AND resource_cd = 'PORTAL_NAV_RESIDENTIAL';

UPDATE ss_resource SET url_ovrd =
'/oracle/webcenter/siteresources/scopedMD/s8bba98ff_4cbb_40b8_beee_296c916a23ed/navigatio
n/gsr0d464579_408d_4f18_af81_fa93d05c737c/OUCSSNavigationModel.xml' WHERE resource_cd =
'PORTAL_NAV_COMMERCIAL';

UPDATE ss_resource_l SET descr_ovrd='Override with CCB 2.3.1 Navigation Model.' WHERE
locale = 'en' AND resource_cd = 'PORTAL_NAV_COMMERCIAL';

UPDATE ss_resource SET url_ovrd =
'/faces/oracle/ugbu/ss/portal/pages/base/AccountList.jspx' WHERE resource_cd =
'PORTAL_LAND_COMMERCIAL';

UPDATE ss_resource_l SET descr_ovrd='Override Landing Page for Commercial with single
account.' WHERE locale = 'en' AND resource_cd = 'PORTAL_LAND_COMMERCIAL';

UPDATE ss_resource SET url_ovrd =
'/faces/oracle/ugbu/ss/portal/pages/base/Dashboard.jspx' WHERE resource_cd =
'PORTAL_LAND_COMMERCIAL_SA';

UPDATE ss_resource_l SET descr_ovrd='Override Landing Page for Commercial with multiple
account.' WHERE locale = 'en' AND resource_cd = 'PORTAL_LAND_COMMERCIAL_SA';
```

7 Restart the WC_CustomPortal server.