Oracle Utilities
Customer Self Service

Implementation Guide

Release 2.1.0

E24862-08

September 2013

ORACLE

Oracle Utilities Customer Self Service Implementation Guide

Release 2.1.0

E24862-08

September 2013

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, 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 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
  - 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
  - Multiple Account management
  - 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.

Technical Overview

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

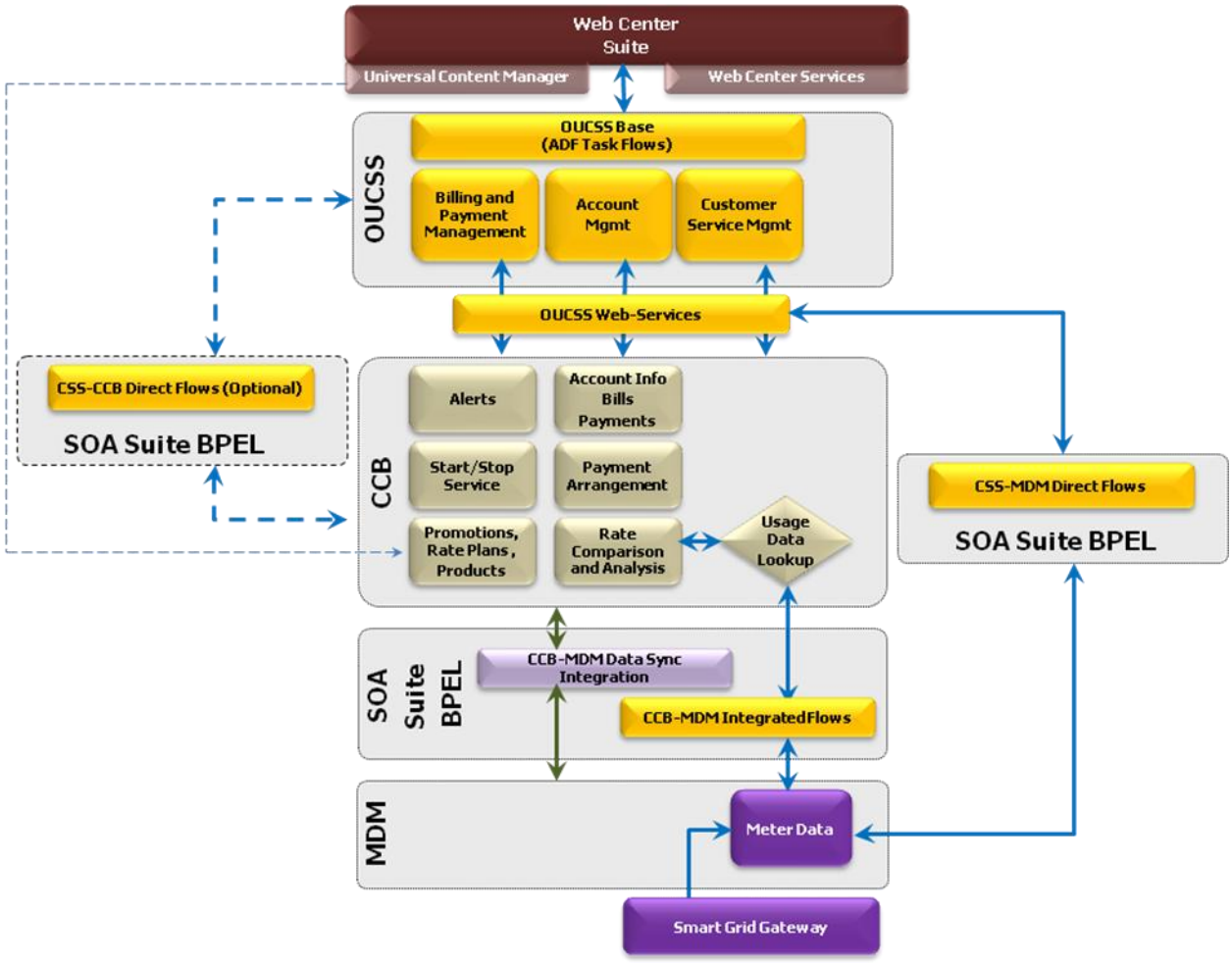
- Portal/Taskflow components are developed using Oracle Application Development Framework (ADF) 11g and are packaged as ADF shared library as well as WSPR 2.0-compliant portlets.
- Taskflows/Portlets are pre-integrated with Oracle Utilities Customer Care and Billing, Oracle Utilities Meter Data Management and Oracle Utilities Network Management System applications using a standards-based web service API and Oracle SOA Suite.
- Oracle WebCenter 11g is the recommended portal platform for consumption with the following approaches:
  - OUCSS taskflows consumed directly in WebCenter Custom Portal application.
  - OUCSS taskflows consumed as WSRP 2.0 portlets in WebCenter Custom Portal application
  - OUCSS Portal application based on Oracle WebCenter Framework (with preconfigured security, navigation model and page templates) is provided with the release package to facilitate implementation and development activities.

Security

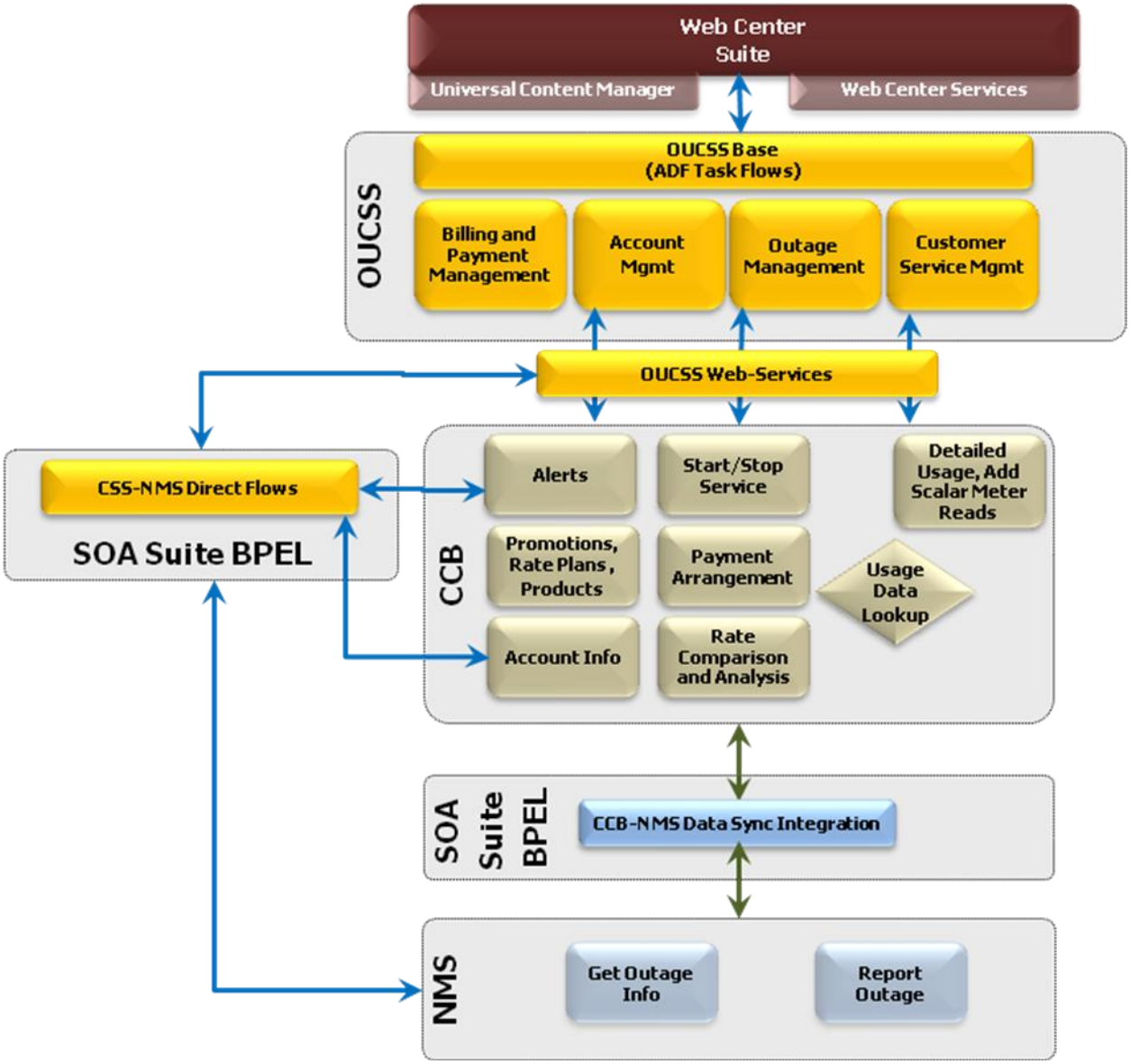
OUCSS offers Tier-1 and Tier-2 security.

- Tier-1 Security: Most of the pages in OUCSS Portal are secured and are accessed through specific roles only. For more information on this type of security see the Reference Security Roles section.
- 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*.

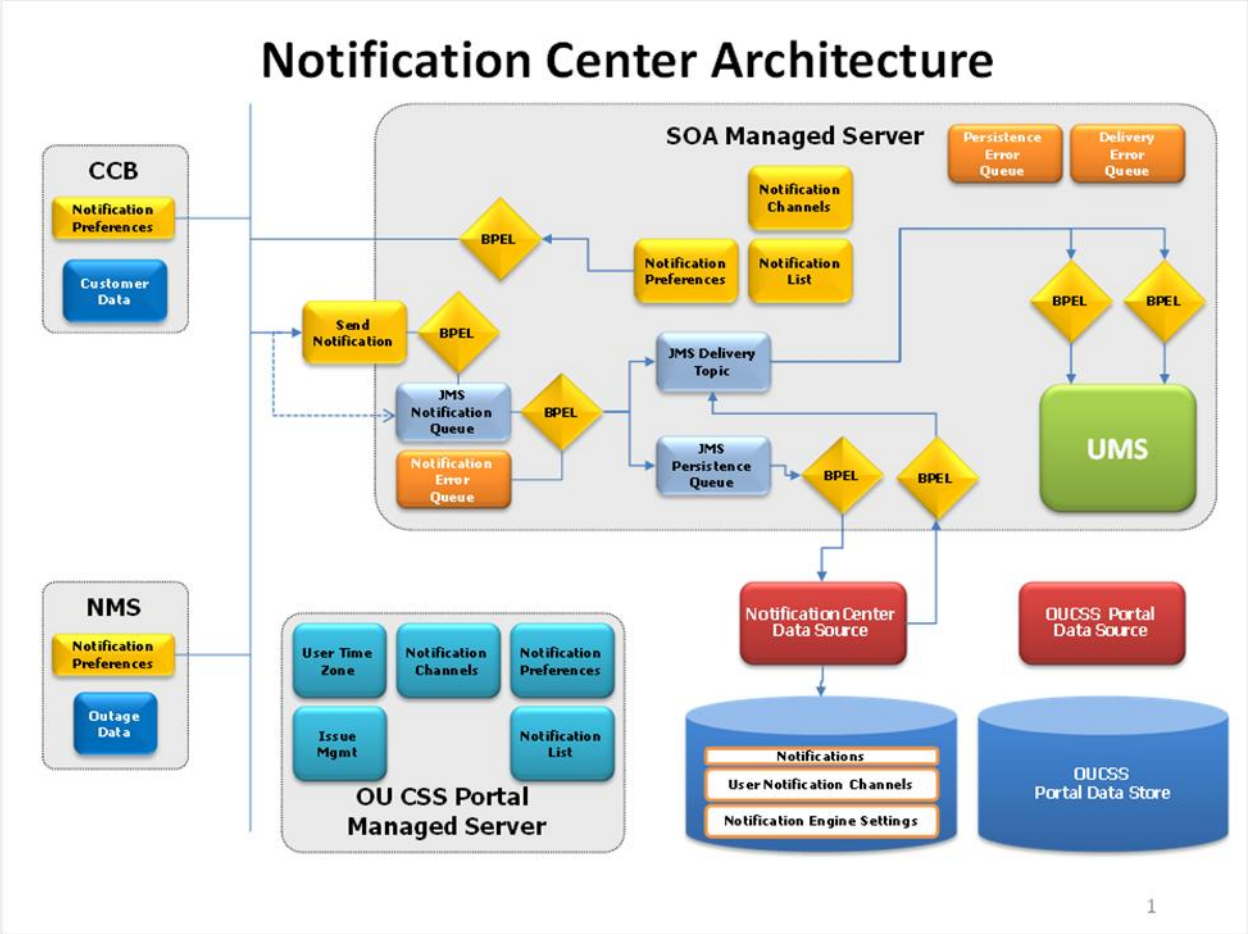
OUCSS Architecture



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
Oracle ADF Mobile Browser documentation	http://docs.oracle.com/cd/E23943_01/web.1111/e10140/toc.htm
Oracle ADF Mobile Skinning Support	http://docs.oracle.com/cd/E23943_01/web.1111/e10140/skinning.htm#CHDCGGDJ
Apache Trinidad Skinning	http://myfaces.apache.org/trinidad/devguide/skinning.html
Selectors for Skinning Trinidad Components	http://myfaces.apache.org/trinidad/skin-selectors.html
WebCenter Custom Portal Application Developer's Guide	http://docs.oracle.com/cd/E28280_01/webcenter.1111/e10148/toc.htm
Oracle Fusion Developers Guide (JDeveloper and ADF)	http://docs.oracle.com/cd/E28280_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 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://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).

Note: This document and the documentation mentioned above is 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 Customer Self Service section of the Oracle Utilities Documentation area on the Oracle Technology Network (OTN) web site (http://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.html).

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 Utilities Customer Care and Billing user documentation.
- For more information on configuring CCB services, see Chapter 5, Network Management System 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 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. The list of verification fields per line of business are defined on the Self-Service Integration master	WXVerifyAccount		

	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
BillingHistoryService BillingHistory	This service retrieves bills and payments for an account for a given period to display in the self-service application.	WXBillPayHistoryRetriever
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	WXEBillSetup
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	WXMaintainMailingAddressInfo

	account's mailing address. Note that when a mailing address is updated, it is stored on the person correspondence information			
WSSAccountPhoneInfoUpdateService AccountManagement, AccountPhoneInfo	This service is responsible for 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 BillingHistory	This service retrieves account's bill details to display in the self-service application.	WXBillView		
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		
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	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 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.		OUCSSOUMDM2DirectUsageDetailReqEBF (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.		OUCSSOUMDM2DirectUsageOverviewEBF (For more information see Direct Usage Overview Integration Flow)	•
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.	WXProcessStartStopRequest		
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	WXProcessPayArrangementRequest		

	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 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 OUCSSOUMDM2FormatGreenButton DataEBF which in turn formats the data to CSV or XML format and returns it to the calling service ,OUCSSOUMDM2GetUsageOverview EBF.		OUCSSOUMDM2Get UsageOverviewEBF OUCSSOUMDM2For matGreenButtonData EBF (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	WXGetSelfServiceDro pdowns		
SSLabelService Admin/Labels	This service retrieves data for populating field labels in the self-service application.	WXGetSelfServiceLab els		
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.		OUCSSOUNMSOuta geSummaryEBF (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.		OUCSSOUNMSTroub leCallInterfaceEBF (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 ,		OUCSSGetOutageScr eenInfoEBF (For more information see Get Outage	

	account information and service related information from CCB and trouble codes information from NMS		Screen Info Integration Flow)
OutageMyDetailService OutageMap (My Outage Details screen)	This BPEL service retrieves current and planned outages for a given account from NMS		OUCSSOUNMSOutageDetailEBF (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).		OUCSSOUNMSOutagePublicDetailEBF (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	This web service is used for fetching	WXRetrievePPBEstim	

PrePaid	the estimated number of days available for a particular prepaid customer before his service is cut.	atesAndCosts	
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	WXRetrieveScalarUsage	
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 for the preferences are retrieved dynamically for CCB and from configuration properties for NMS	WXSetNotificationPreferences	OUNCWXNotification Preferences (For more information see Notification Management OUCSS Integration Services)
SSDeliveryChannelsService	It is responsible for retrieving delivery channels information for the self-		OUNCWXDeliveryCh

SSDeliveryChannels	service user.It is the Profile page under Notifications		annels (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)
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 service is called	WXMultipleAccountFinancialHistory	
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 Application (Residential)

The reference OUCSS Portal contains pages with OUCSS taskflows. The taskflows are spread across different pages and logically grouped using a Navigation model. The Portal uses the default ADF security model to allow authentication and authorized access to valid groups and users only.

Portal Pages

The OUCSS Portal allows any user access to public pages such as Home, Register, Reset Password, etc. It also implements secured pages related to an individual's accounts.

Visible Public Pages

- Home
- Login
- Outage
- Register
- Reset Password

Hidden Public Pages

These pages are not part of the Navigation model.

- Page Not Found
- UnAuthorized Page
- Error Page

Visible Secured Pages

- Accounts
- Details
  - Dashboard
  - Financial History
  - Budget Billing (Post Paid accounts only)
  - View Bill
  - 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
- Notification
  - Inbox

  - Profile
  - Preferences
- Outages
  - Display Map
  - Display List
  - Report Public Outage
- New Customer
- User Profile
- Forms Management
  - Log an Issue
  - Form List
- Old (Supported) Portlets
  - Billing History
  - Usage Details
- Admin
  - Customer Search
  - Configuration Options
  - Resources
  - Access
  - Security
  - Edge Application
  - Line of Business
  - Portlets
  - Language
  - Labels
  - Lookup
  - Messages
  - Train
  - Offers

Hidden Secured Pages

These pages are not part of the Navigation model.

- One Time Payment
- Add Scalar Read
- Manage Address
- Manage Phone

- Manage Electronic Bill Option
- Manage Billing Notification Preference
- Manage Automatic Payment Option

OUCSS Taskflows dropped on each Page
Public Pages

Home

- No Taskflows
- Login

Reset Password

- Forgot Password.

Register

- Registration.

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).

Secure Pages

Accounts

- View Account List
- Remove My Access
- Invite
- Revoke Access
- Enroll

Dashboard (Post Paid Account)

- View Account Information
- Account Charges Summary

- Alerts
- Manage Budget Billing
- Service Charges to Date
- Usage Overview
- Scalar Consumption Summary
- Banner Promotion

Dashboard (Pre Paid Account)

- View Account Information
- Prepaid Balance and Charges
- Alerts
- Prepaid Estimates and Cost
- Service Charges to Date
- Usage Overview
- Scalar Consumption Summary
- Banner Promotion

Financial History

- View Account Information
- Financial History

Budget Billing

- View Account Information
- View Budget Billing

View Bill

- View Account Information
- View Bill

Payment Arrangement

- View Account Information
- Payment Arrangement

Compare Rates

- View Account Information
- Compare Rate Plans

Usage

- View Account Information
- Usage Detail (New with Time Selector)
- Scalar Usage Detail
- Usage Download

Personal Information

- View Account Information
- View Mailing Address
- View Phone Information
- View Billing Notification Preferences
- View Bill Delivery Method
- View Automatic Payment Option

My Outage Details

- Outage Details

Report Outage (Premise)

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

Start Service

- Start Service

Stop Service

- Stop Service

Transfer Service

- Transfer Service

New Customer

- New Customer Service

User Profile

- Change User Profile
- Change Password

Notification - Inbox

- Inbox

Notification - Profile

- Profile

Notification - Preferences

- Preference

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

Billing History (Old Supported Portlets)

- View Account Information
- Billing History

Usage Details(Old Supported Portlets)

- View Account Information
- Usage Details(old)

Secure Hidden Pages

Scalar Read

- View Account Information
- Add Meter Reading

Pay Now

- View Account Information
- Make One Time Payment

Manage Address

- View Account Information
- Update Mailing Address

Manage Phone

- View Account Information
- Update Phone Information

Manage Auto Pay

- View Account Information
- Update Automatic Payment Option

Manage Bill Notify

- View Account Information
- Update Billing Notification Preferences

Manage EBill

- View Account Information
- Update Electronic Bill Option

Form Update

- Update Form

Account Documents

- View Account Information
- Read Account Documents

Portal Resources

Portal Context

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 Generic and LOB Specific.

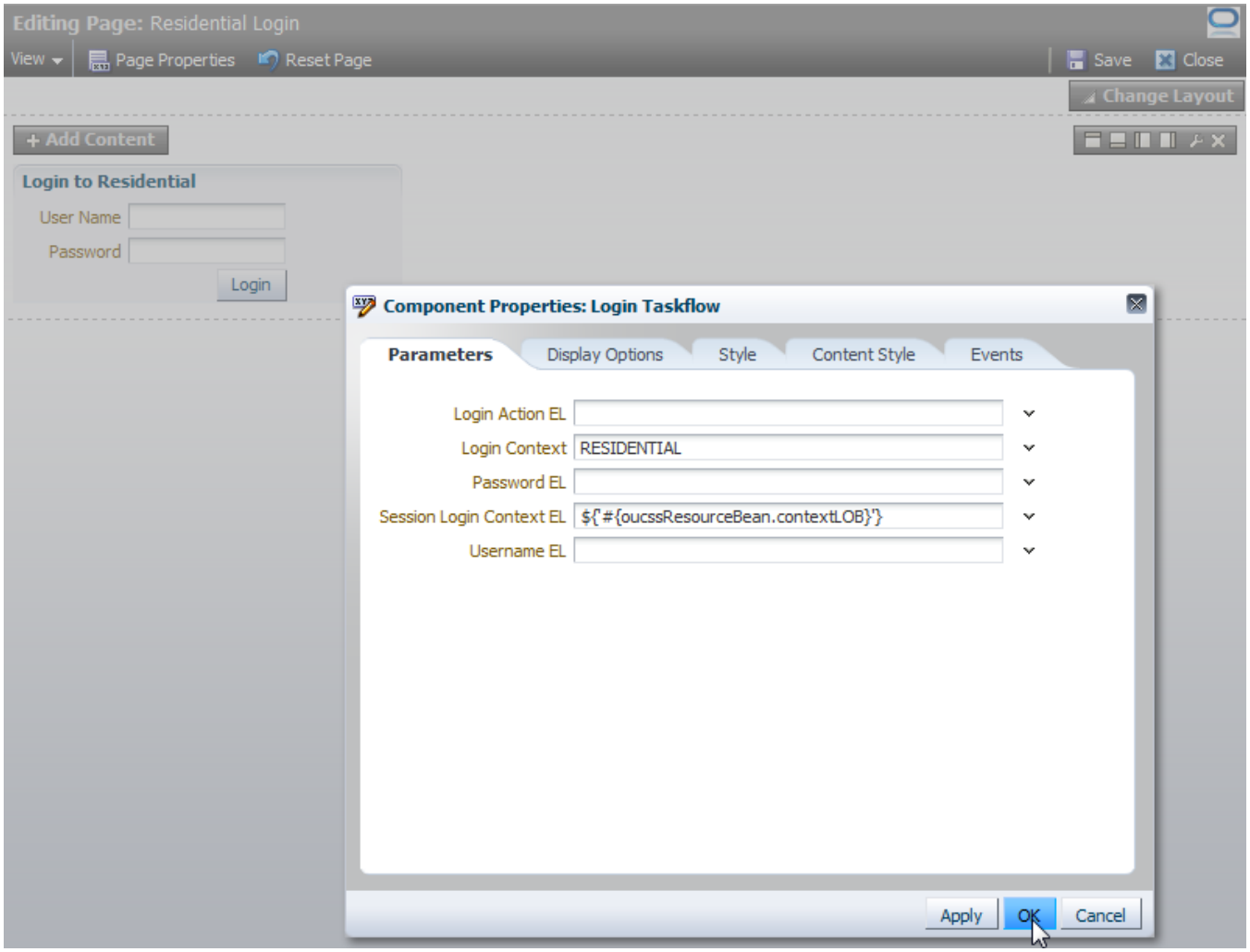
Generic Login

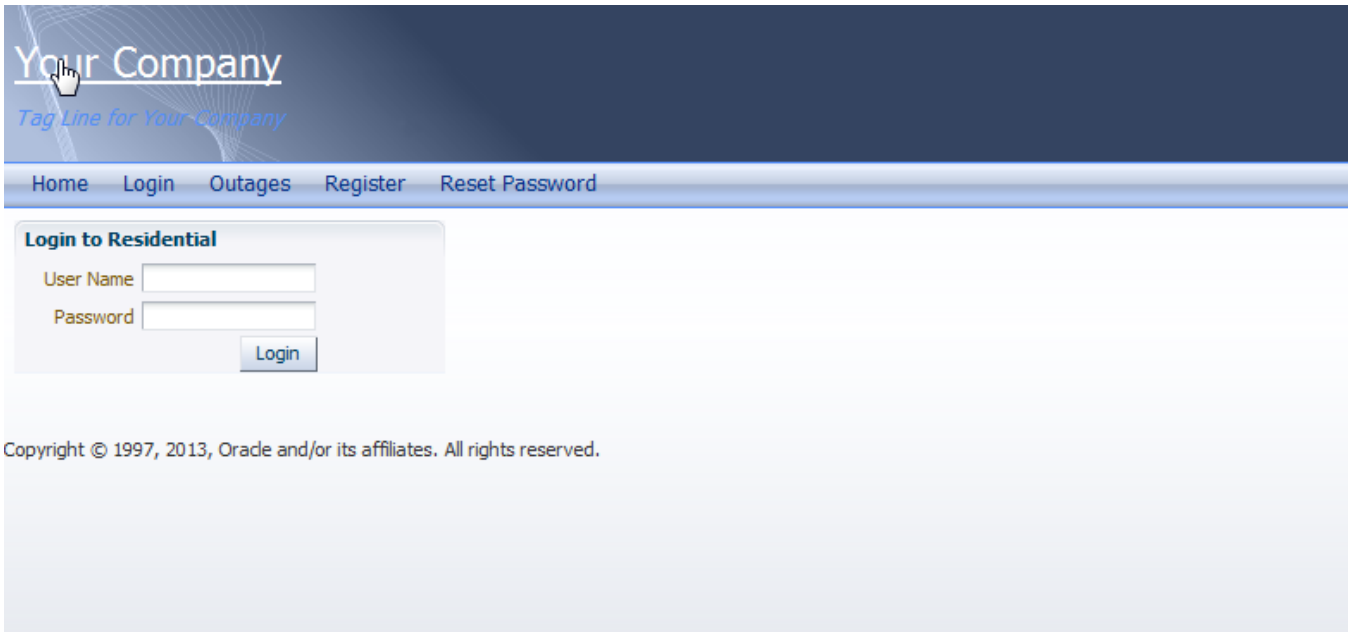
By default when Login taskflow is dropped onto a page, it is generic (LOB Input param is empty). In this mode, a list box is shown to allow users select the Context along with entering Username and password before login.

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 configured 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}'}.





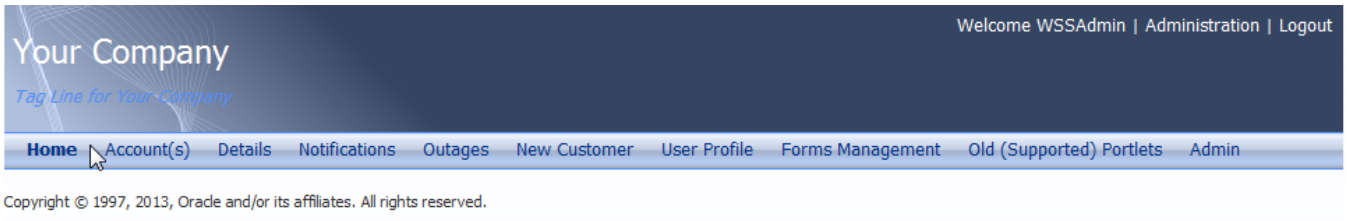
Page Template

For Residential and Public users, OUCSS Portal is configured to use the **Swooshy Page 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 User, OUCSS Portal is configured to use the Global Page Tempate.

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

Preview of Swooshy Page Template



Navigation

OUCSS Portal is configured to use separate Navigation Model for each LOB/Context supported. OTB, OUCSS Portal is configured with separate Navigaiton model for each Public, Residential and Commercial users.

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

Navigation Model for Public Users

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

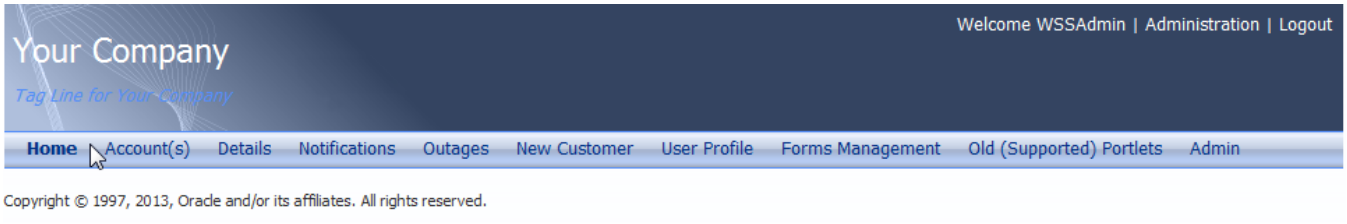
Navigation Model for Residential Users

OUCSS Portal is configured with an OUCSS Default Navigation Model for residential customers to allow access to secured pages. The Navigation model is configured to restrict links based on the criteria like valid OUCSS users (that is users who are registered using OUCSS the Registration), enrolled to one or more accounts, or user with Administrator or CSR privileges.

Navigation Model for Authenticated Users

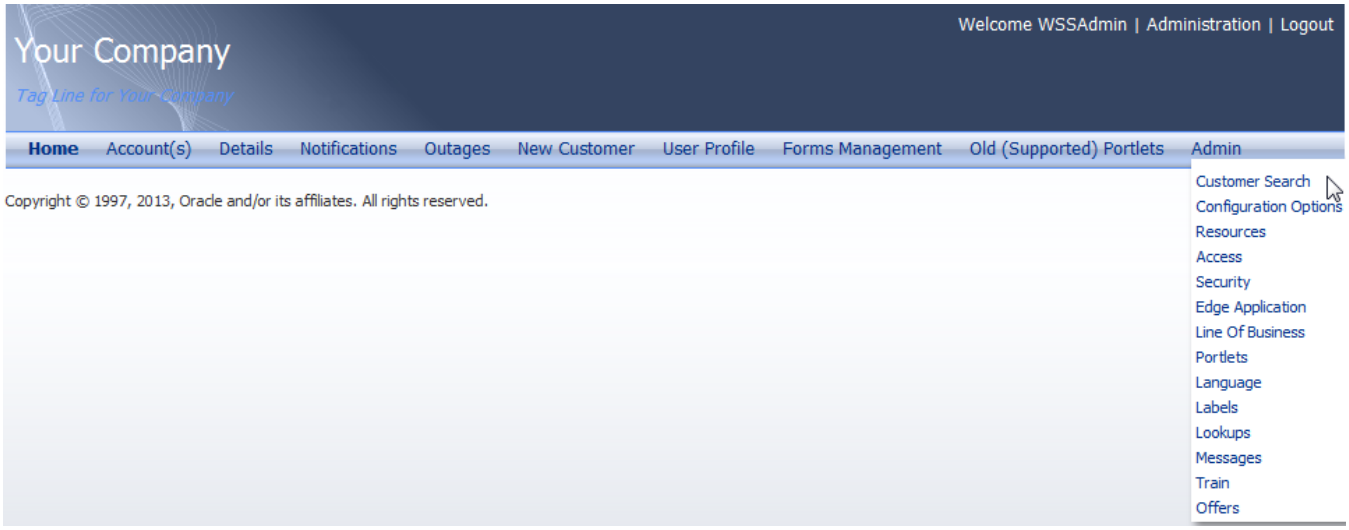
Authenticated users are users who login to Portal with a valid user name and password.

On login, the user has access to all public and secured pages. If the user is enrolled to one or more accounts, the “Details” link is visible.



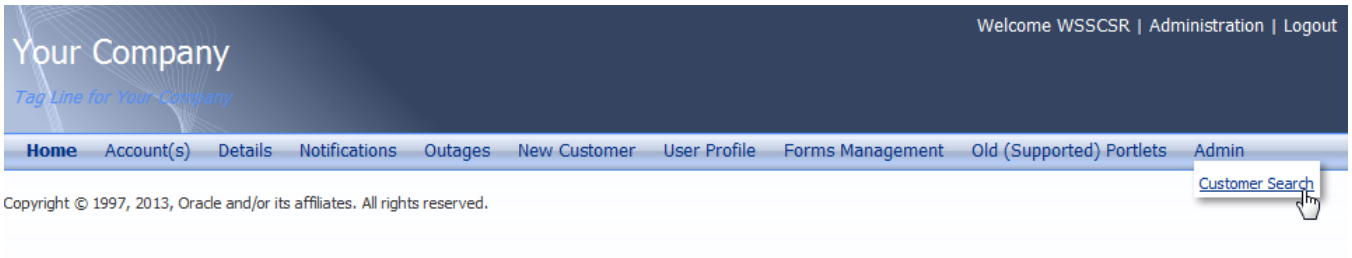
Navigation Model for OUCSS Admin Users

Users who are member of the WSSAdminGroup are OUCSS Admin. Admin is allowed access to the Admin menu, which permits configuration of the OUCSS Portal.



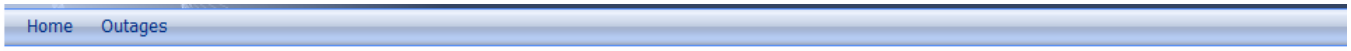
Navigation Model for WSSCSR User

A CSR is a user who is a member of the WSSCSRGroup. CSR have ability to impersonate any valid user of Portal. They get access to Admin > Customer Search page to search for any user in the system and access their account details. Refer to Customer Search for more information.



Navigation Model for Invalid User

Users who are logged in but not registered using the OUCSS Registration portlet cannot access any secured pages. These users have access only to public and non-account related pages.



Resource Catalog

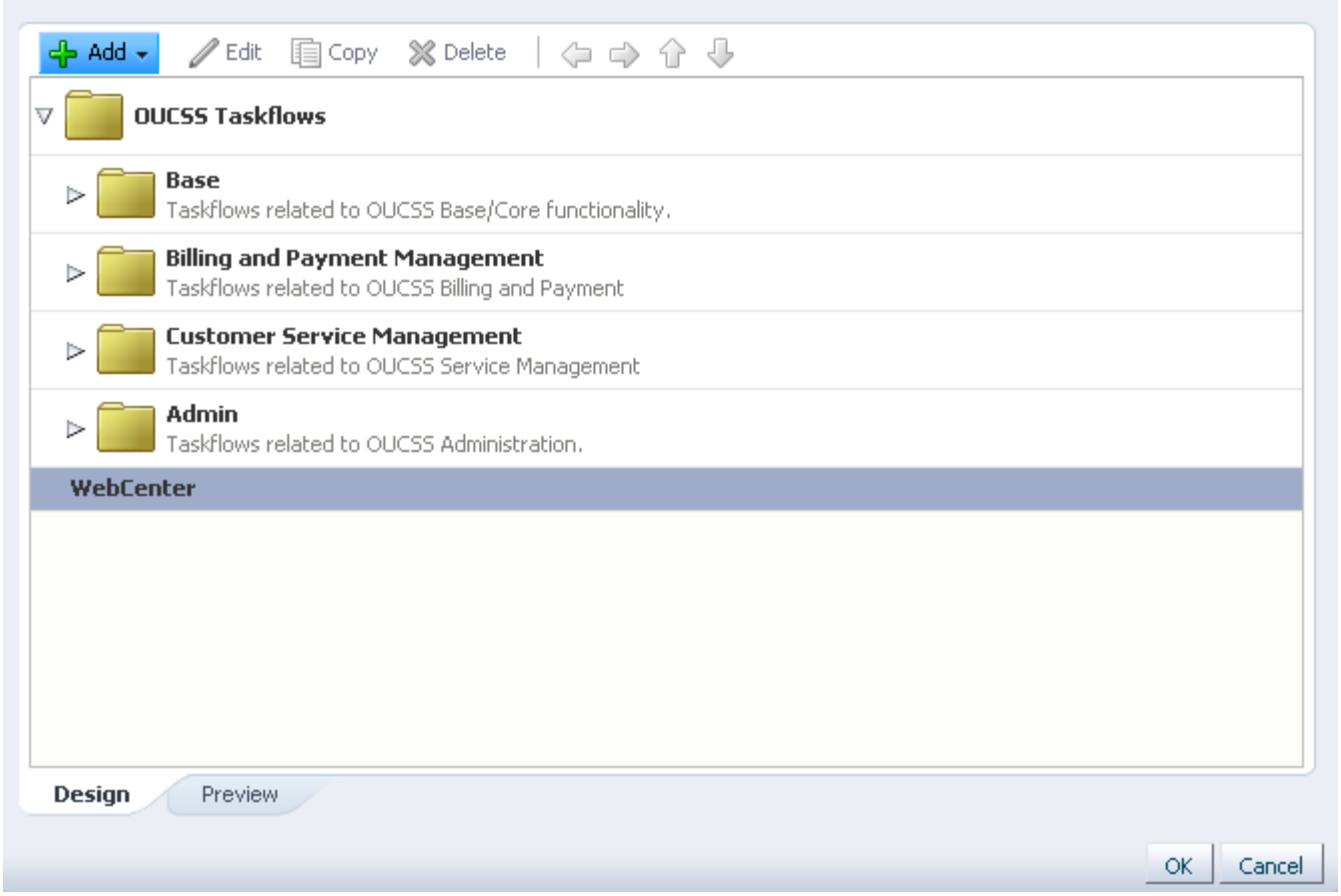
OUCSS Portal is configured to use separate Resource Catalog for each type of LOB/Context. OUCSS Admin Users can create or edit the page and drop required components/OUCSS taskflows using Resource Catalog.

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.

Resource Catalog for Residential User

OUCSS Portal is configured to use OUCSS Default Resource Catalog for Residential Context, which extends the out-of-box Default resource Catalog along with references to all OUCSS Residential Taskflows.

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



Reference Security Roles

The OUCSS 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 application.

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 OUCSS Portal . This user can manage all resources of OUCSS Portal. **WSSAdmin** user is a member of **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 Portal Application (Commercial)

OUCSS Portal supports multiple contexts based on the line of business selected during login. The commercial context has its own set of secured pages, navigation model and resource catalog.

Portal Pages

Once logged in as a commercial user, the visible secured pages that are exclusive to commercial context are

- Set Accounts
- Multi Account
- Financial History
- Usage Compare
- Usage Aggregate

Other pages available in Commercial context are derived from Residential application.

OUCSS Taskflows (Commercial Pages)

The following are the taskflows dropped on the pages that are exclusive to a commercial context.

Secure Pages

Set Management

- CommercialAccounts taskflow
- Enroll
- Invite

Financial History

- Business Context
- Commercial Bill History

Usage Compare

- Business Context
- Usage Comparison

Usage Aggregate

- Business Context

- Usage Aggregation
- Usage download

Portal Resources

Navigation

Navigation Model for Commercial Users

OUCSS Portal is configured with OUCSS Commercial Navigation Model for commercial customers to allow access to secured pages. The Navigation model is configured to display pages that are restricted to valid OUCSS users (that is users who are registered using OUCSS the Registration), enrolled to one or more accounts, or possess Administrator or CSR privileges.

Commercial Context allows access to Multi-Account modules to group accounts as sets and view and compare Usage and Financial history for multiple accounts.

Resource Catalog

Resource Catalog for Commercial User

OUCSS Portal is configured to use OUCSS Commercial Catalog for Commercial Context, which extends the out-of-box Default resource Catalog along with references to all OUCSS Commercial Taskflows.

OUCSS Commercial Taskflows are logically grouped with the modules name

The security roles would be same as for Residential application. Refer for more information Reference Security Roles

OUCSS Application Configuration (Admin)

Customer Search

The Customer Search page is accessible only to OUCSS Administrators and CSRs. This search page is accessible from **Admin** > **Customer Search** menu.

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

OUCSS User Search

The user search allows the CSR to search for the user based on 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 page using **Admin** > **Access**.

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.

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 this page using **Admin > Labels**.

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. Labeles for these questions are stored in CCB application.

To load these labels from CCB into OUCSS, go to **Admin > Labels**. On the Actions menu dropdown select Reload Labels. After a successful reload, you flush the cache by selecting Flush Cache from the Actions menu dropdown.

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 online 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 this page using **Admin > Language**.

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, a user sees the system in their default language which is defined on their browser (e.g., Internet Explorer, Firefox).

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 this page using Admin > Edge Application.

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 is the server information where the edge application is running.

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 this page using **Admin > Line of Business**.

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 this page using **Admin > Lookup**.

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. 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. 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 a existing LOB, update the "URL Override" column of the corresponding resource with new Site Template URL using Admin > Resources page.

Adding New Site Template

To add a new Site Template for a new LOB:

1. Go to Admin > Resource screen.
2. Add a new Resource of WXURL type with the Site Template URL.
3. Go to Admin > Lookup screen.
4. From the List of lookups, select the Lookup with Code PORTAL_SITE_TEMPLATE.
5. In the Lookup Value table, add a 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.

Navigation

Portal Navigation is configured using the Lookup Code PORTAL_NAVIGATION. By default Navigations are defined for Public, Residential and Commercial LOBs. The relative URL of the navigation is stored as Resources.

Updating Default Navigation

To update the default Navigation for a existing LOB, update the "URL Override" column of the corresponding resource with new Navigation URL using Admin > Resources page.

Adding New Navigation

To add a new Navigation for a new LOB:

1. Go to Admin > Resource screen.
2. Add a new Resource of WXURL type with the Navigation URL. Save the changes.
3. Go to Admin > Lookup screen.
4. From the List of lookups, select the Lookup with Code PORTAL_NAVIGATION.
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. By default catalogs are defined for Residential and Commercial LOBs. The relative URL of the navigation 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 URL using Admin > Resources page.

Adding New Catalog

To add a catalog for a newly added LOB:

1. Go to Admin > Resource screen.
2. Add a new Resource of WXURL type with the Catalog URL. Save the changes.
3. Go to Admin > Lookup screen.
4. From the List of lookups, select the Lookup with Code PORTAL_CATALOG.
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. By default landing pages are defined for Residential and Commercial LOBs. Landing pages can 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 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 Admin > Resources page.

Adding New Landing Page

To add a landing page for a newly added LOB:

1. Go to Admin > Resource screen.
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 Admin > Lookup screen.
4. From the List of lookups, select the Lookup with Code PORTAL_ LANDING_PAGE.
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.

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 online Help provided 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.

Offer Service (Web Service)

Offer Service is installed by default as part of installation as a separate 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.

1. Implement custom Offers Web Service and make sure the WSDLs are identical to the current service.
2. To update the Web Service used by Offers in OUCSS:
3. Login to Enterprise Manager.
4. Click on the deployed application (for example, for the portlets solution, click on OUCSSPortletsProducer (v2.1.0); for the OUCSS Portal solution, click on OUCSSPortal (v2.1.0)).
5. From the Application Deployment menu select ADF > Configure ADF Connections.
6. Select the Offers Service connection and click Edit.
7. Update the WSDL and Service Name of the connection to a new service.
8. Click OK, then open the Advanced Connection Configurations menu and select the port to update the End Point URL.
9. Select the Configurations tab.
10. Update the Endpoint Address in the General section, and then click Apply to commit the changes.

11 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 functionality to 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, then 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, then an email will be sent to the user. User will have to register with the Self-Service application in order to view the accounts. The process will be similar to 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.

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/ OUCSS_AccountEnroll/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
- All the customers provided in the mainData section will be enrolled to all the accounts. Email will be sent to each customer with the list of accounts.
- 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 Admin group on the Top menu, then choose Security.

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:commandButton 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 this page using **Admin > Portlets**.

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 this page using **Admin > Message**.

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 this page using **Admin > Trains.**

- **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 **Admin > System**. After changes have been made through user interface the system must be restarted for 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:

Configuration Property	Description	Value
installation.owner.flag	This is the current owner of the application. For your implementation, this property must be set to CM (Customer Modification).	CM
edgeapplication.ccb.datasource	This is the edge application code (e.g., C1, M1, etc., from where the data is pulled).	C1
lookups.webservice.status	This is the status of the lookup values when they are loaded into OUCSS. Allowed values are **ACTIVE** or **INACTIVE**.	ACTIVE
webcenter.login.url	This is the URL link to login to webcenter.	http://<Specify Server Name>:<Specify Port Number>/OUCSSPortal
webcenter.register.url	This is the URL link to login to webcenter together with a registration key.	http://<Specify Server Name>:<Specify Port Number>/OUCSSPortal/faces/oracle/ugbu/ss/portal/pages/base/AccountList.jspx?regKe

		y=
mail.session.jndi.name	JNDI reference to the Mail Session to send emails.	Mail/OUCSS
validate.regex.username	Regular expression based on Java Regular Expression to validate the username	[a-zA-Z0-9_]*
validate.regex.email	Regular expression based on Java Regular Expression to validate the email	[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,4}
validate.length.password.min	Minimum length of password in integer	6
validate.length.password.max	Maximum length of password in integer	12
validate.regex.password	Regular expression based on Java Regular Expression to validate the password.	[a-zA-Z0-9]*
oucss.default.locale	Default locale that is to be used if a valid supported locale is not found in user request.	en
rollback.on.failed.email	Indicates whether user's registration is rolled back should the exception during sending a confirmation email occur. The valid values for the flag are : "Y" or "N"	Y
account.list.max.rows	This property specifies what number of associated accounts 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 a number of accounts per page presented in Account List, if the number of associated accounts exceed the value of this parameter paging will be enabled	10
default.newcust.requestmode	This is the default request mode for New Customer Service. Current value is set to "C1ST".	C1ST
default.customerclass.parm	This is the default customer class used by the New Customer Service to retrieve necessary information to be rendered in the New Customer Screen. Current value is "R" since in this release, only Residential customers are catered.	R
default.personbusiness.parm	This is the default value used by the New Customer Service to retrieve necessary information to be rendered in the New Customer Screen. Current value is "P" for person.	P
mobile.brow.branding.image.bg	This property controls the background color for branding image for OUCSS Mobile browser solution	#FF0000
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
max.rows.premise.search	Set this property to the max number of rows that can result in Premise search. The system will show error if the premise search in Start Service and New Customer service exceeds the value set here.	10
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).
outage.map.color.theme.colors	This property is configured to set the number of buckets to	#00FF00;#EEEE00;#FF7F00; #FF0000

	aggregate the Outages as well as the respective color of each bucket.	(Green, Yellow, Orange and Red)
outage.map.srid	This property is SRID of the Coordinate system used by the Base Map and Theme configured above.	Value configured in InstallProperties.xml /8307
outage.map.color.theme.loc	This property is the Area Column from the Color Theme configured above.	Value configured in InstallProperties.xml
outage.map.color.theme	This property is used to color code Outages in the Outage Summary.	Value configured in InstallProperties.xml
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
oucss.comm.context.limit	The number of accounts that will be used to compare/load commercial modules (Bill History, Usage etc) from the Business context	10
oucss.comm.context.pagesize	The maximum number of accounts that can be displayed before a scroll bar appears	10
notification.list.page.size	The default number of notifications that will be displayed in the Notifications Inbox portlet.	15
notif.list.chars.preview	The default number of characters to be displayed in the Notifications Inbox for a given notification message, beyond which there is a means of toggling the contents under a header between being disclosed(shown) or undisclosed(hidden).	140
set.account.page.size	Pagination limit for accounts in Set. Pagination controls (Next and Previous buttons) will be displayed if the set contains number of accounts more than the size configured.	10
csr.account.access	This configuration will be used to define the Access Role to CSR User when accessing an account. This should be a value as defined on the SS_ACCESS_ROLE table. (i.e. ACCOUNT_HOLDER, GUEST, CSR_ACCESS, etc.) By default the value will be ""ACCOUNT_HOLDER"""	ACCOUNT_HOLDER
csr.search.results.fetchsize	This is added to control the size of the search results grid thus limiting the initial display of records. It will have a default value of 300, the same as the maximum number of records that the CCB can return. User can define less than 300 but if ever user defines a value greater than 300, the application will still display a maximum of 300 records	300
default.enroll.role	The default access role with Enroll capability	ACCOUNT_HOLDER
oucss.reset.password.pattern	Password pattern during reset operation.	0123456789abcdefghijklmnopqrstuvwxyz$# *_ABCDEFGHIJKLMNOPQRSTUVWXYZ
outage.map.base.startingX	The default X value for the Outage Map	-81.7
outage.map.base.startingY	The default Y value for the Outage Map	40.69
outage.map.base.zoom	The zoom level in the Outage Map Screen	4
attachment.list.max.row	The maximum number of attachments allowed on the Forms	5

	Management module	
forms.list.page.size	This configuration controls the number of Forms to be displayed in the Forms List page. Default value is set to 10 forms per page. Admin/implementer can define value greater than the default value of 10. If the value defined is invalid, that is the value is not a valid number (empty/null, alpha), the application will default it to 10. If the value defined is less than 1 (0 or negative numbers), the application will send the value at is (since this is still a valid number) resulting to no data found.	10
oucss.debug.enable	Prints out extra information during exceptional case when set to true.	FALSE
def.day.mode.range.in.month	This configuration will be use by Usage Detail to set a default range for each view mode. Default value is 3 months.	3
def.hour.mode.range.in.days	This configuration will be use by Usage Detail to set a default range for each view mode. Default value is 7 months.	
def.month.mode.range.in.year	This configuration will be use by Usage Detail to set a default range for each view mode. Default value is 1 year.	1
max.day.mode.range.in.years	This configuration will be use by Usage Detail to set a maximum range that the system can process for each view mode. Default value is 1 year	1
max.hour.mode.range.in.days	This configuration will be use by Usage Detail to set a maximum range that the system can process for each view mode. Default value is 30 days.	30
max.month.mode.range.in.years	This configuration will be use by Usage Detail to set a maximum range that the system can process for each view mode. Default value is 4 years.	4
scalar.usage.graph.color	The default color of the Scalar Usage Graph	#660033
notif.validate.regex.phone	Notification Profile phone validation pattern	^(?:\+?1[-.]?)?\(?(\d{3})\)?[-.]?(\d{3})[-.]?(\d{4})$

Resources

Open this page using **Admin** > **Resources**.

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 Admin > Resources screen. By default, these resources are not created as part of product release. Implementation can choose to configure the Page Not Found, Un Authorized and Server Error pages using the following codes.

RESOURCE_CD	Purpose
PORTAL_PAGE_NOT_FOUND_PAGE	Use this code to configure the Page Not Found page.
PORTAL_UNAUTHORIZED_PAGE	Use this code to configure the Un Authorized page.
PORTAL_SERVER_ERROR_PAGE	Use this code to configure the Server Error page.

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.

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 online 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 online 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:

- Forms
- Additional Customer Requests

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, including the following:

- List of alerts that should be returned to the self-service application
- Valid payment and credit card types used when setting a customer up on auto pay
- Verification information required when a self-service customer enrolls an account
- Scripts used to construct information strings displayed in the self-service application
- Self-service task types used for the various requests supported e.g., meter read creation, one-time payment, automatic payment setup, bill ready notification, etc.
- Information needed to interact with an external system, e.g., requesting usage from MDM, sending email notifications to a self-service user, etc.

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.

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 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**.
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'.
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.
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.

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.
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.
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.

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

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

- Setup an XAI inbound service to create the task instance (see the XAI 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

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.
- 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.

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

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 creaion, 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 inputThe system will attempt to retrieve usage information from MDM for each of the account's service agreements that require bill determinantsThe 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 determinantsThis 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 dateThe 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 configurationThis 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 'from' address on the email.

XAI Configuration

XAI Sender

To create a Real-time XAI sender configured to communicate with the integration layer:

XAI Sender for Usage Request Integration Service

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

1. Navigate to Admin Menu, XAI Sender.
2. Enter a unique XAI 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*
   - XAI Sender = (XAI 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*

- XAI Sender = (XAI 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*
- XAI Sender = (XAI 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.

XAI Inbound Services

The following are the base XAI Inbound 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 configurationThe master configuration contains pertinent information needed for a MDM and self-service integration, including the following:

Multiple Accounts Aggregation/Comparison Details

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.

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.

Usage Calculation Supported Usage Groups List

In order for MDM to return usage-to-date information for a given usage subscription or return calculated usages for rate analysis, the usage subscription’s usage group must be defined in this list.

Supported Scalar Usage Groups List

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.

Processing Service 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 Task Types

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 XAI inbound service to create the task instance (see the XAI 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

Create Scalar Read Service Task Type

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.

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 3.1.1 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.

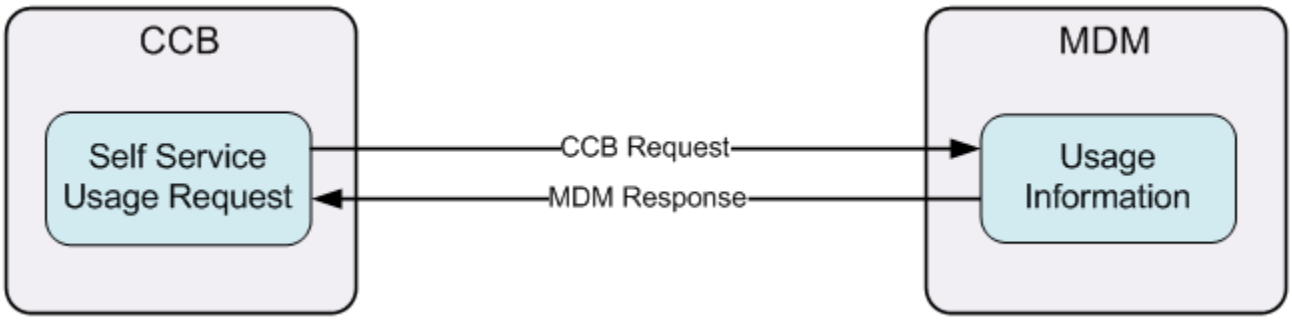
Supported Business Processes

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.
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.

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



Understanding the CCB-MDM Integrated Flows

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 information.

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

One BPEL process manages each integration flow and the BPEL process is exposed as a web service. The BPEL Process handles the following:

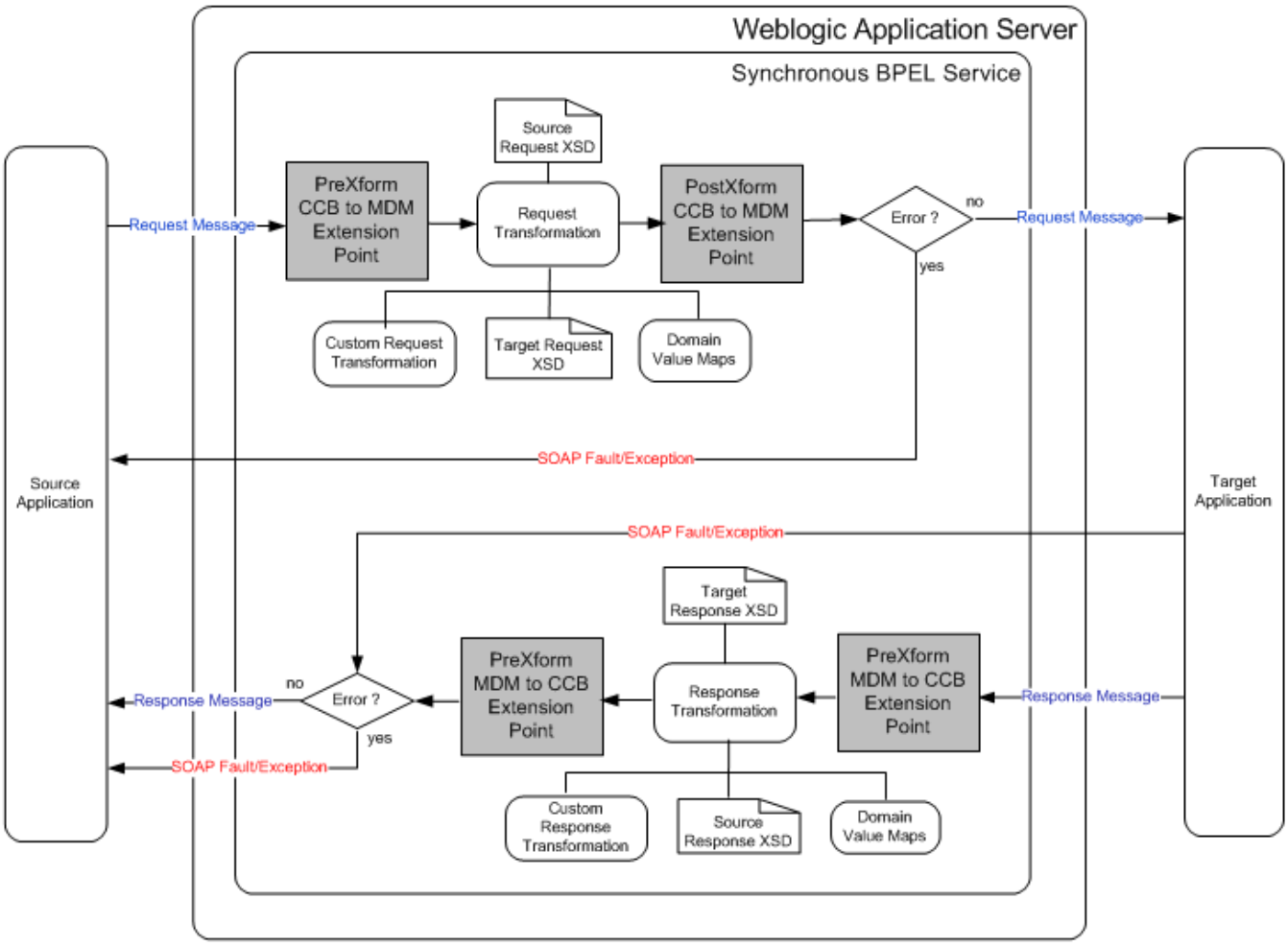
- Transform the request message coming from the source application (CCB) to the target application's (MDM) format and invoke the MDM service synchronously.
- Receives the response message coming from the target application (MDM), transforms the response message to the source application (CCB) format and sends back the response to CCB.
- Handles message extensions.
  - If the extension point flag (Extension.PreXformCCB2toMDM2) is enabled, it will invoke the PreXform CCB to MDM Custom Extension Service.
  - If the extension point flag (Extension.PostXformCCB2toMDM2) is enabled, it will invoke the PostXform CCB to MDM Custom Extension Service.
  - If the extension point flag (Extension.PreXformMDM2toCCB2) is enabled, it will invoke the PreXform MDM to CCB Custom Extension Service.

- If the extension point flag (Extension.PostXformMDM2toCCB2) is enabled, it will invoke the PostXform MDM 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.

- Any exception encountered by the integration will send back a SOAP Fault to CCB. This includes technical errors (e.g., connectivity errors) and transformation errors.
- Any exception or faults that the integration receives from MDM will be sent back to CCB.

Technical Flow

The following diagram describes the technical flow of the CCB-MDM integrated flow:



Technical Flow Diagram

Integration Points

Self Service Usage Request Integration Flow

Business Details

This process is a real-time synchronous interface from Oracle Utilities Customer Care and Billing (CCB) to get usage from Oracle Utilities Meter data management (MDM) to be used for Rate Analysis and Usage Charges To Date Calculation.

Oracle Utilities Customer Care and Billing (CCB) sends out a usage request to Oracle Utilities Meter Data Management (MDM) for rate analysis and calculation of usage charges to date if the Service Agreement (SA) requires bill determinants

- When CCB does Rate Analysis for a given service agreement (SA), it first determines the most recent 12 months charges for the given SA. If the SA requires bill determinants from MDM, it sends a real time usage request to MDM to get the usage for the 12 month periods provided for both the base rate schedule and the comparison rate schedule. The request contains a list of bill segment periods for the given SA, base rate schedule and any rate version break periods and another list of bill segment periods for the given SA and the comparison rate schedule and any rate version break periods. Oracle Utilities Meter Data Management (MDM) calculates bill determinants (usage) based on the available read data and usage subscription configuration. These Bill determinants are returned to Oracle Utilities Customer Care and Billing (CCB) and used for rate calculation.
- When CCB calculates the Usage Charges to Date for a specific SA or specific account, which can have more than one SA, and the SA(s) requires bill determinants from MDM, it sends a real time usage request to MDM to get the usage for the period provided for the SA(s). The request can contain a SA or a list of SAs for a given bill segment period. Oracle Utilities Meter Data Management (MDM) calculates bill determinants (usage) based on the available read data and usage subscription configuration. These Bill determinants are returned to Oracle Utilities Customer Care and Billing and used for calculation of usage charges to date.

Technical Details

Oracle Utilities Customer Care and Billing initiates a Usage Request when Rate Analysis Service or Usage To Date service requires bill determinants and Oracle Utilities Meter Data Management returns bill determinants for each request.

- CCB will send the usage request information in form of xml messages which will be transformed by integration and sent to MDM. MDM will respond back with the usage information which will be transformed by integration and sent back to CCB.
- DVMs are used for some of the data transformation. No new DVMs are introduced, only existing DVMs used for CCB-MDM2 integration are used.
- For missing DVM lookup values, where the value coming from an element in the CCB message is not found in the DVM table, integration will always pass the value as is to MDM and vice versa for messages coming from MDM to CCB. It will not throw any error to CCB. The exception flag defined in the configuration properties are not used in this flow.
- Integration passes the language value coming from the CCB request message to MDM by assigning the language in the SOAP Header when the MDM XAI Inbound service, ***D2_CalculateUsageMultipleRequests***, is invoked. This language that CCB passed to MDM is the CSS user's language. This is use to get the appropriate descriptions in MDM as part of the response message going back to CSS given the CSS user's language.
- CCB's request message contains a list of usage request and MDM respond with a list of usage back to CCB. (refer to Usage Request Mapping for more details)

  - For Rate Analysis, a list of bill segment periods for the given SA, the base rate schedule, and any rate version break periods and another list of bill segment periods for the given SA and the comparison rate schedule and any rate version break.
  - For the Usage To Date Calculation, a SA for a given bill segment period or a list of SAs for a given bill segment period.
  - While processing a record in the usage request list and MDM encounters an application error, MDM populates the error in the exception information of the MDM response and move on to the next record on the list. When integration gets the response back, it will map the exception information from the MDM response to the error information in the CCB response message.
  - If the SA Id of a record in the usage request list is not supported in MDM, MDM will skip the record and marked it as skipped in the response message and proceed to the next record.
- If integration encounters an exception (e.g., connectivity error, transformation error) while processing the message, integration will return a SOAP fault back to CCB.
- If MDM sends an exception or fault back to the integration, integration will return a SOAP fault back to CCB.
- No email notifications for Business and Technical errors will be sent out from the integration service.

Integration Services

Name	Description
OUCCB2OUMDM2SSUsageReqEBF	CCB-MDM Usage Request BPEL Process for Self Service Synchronous BPEL process that transform incoming CCB request message to MDM format and invoke the MDM inbound service to retrieve usage information. Transform the response coming from MDM back to CCB format.

External Service Call

Web Services

Application	XAI Service Name	Description
MDM	D2-CalculateUsageMultipleRequests	Calculate Usage Multiple Requests This inbound service is used by the self-service application for retrieving usage information for Rate Analysis and Charges to Date calculation.

Notes:

- Usage Overview Integration Flow (OUCCB2OUMDM2SSUsageOverviewReqEBF) is deprecated in the OUCSS 2.1.0 version. Use the CSS-MDM Direct Usage Overview Integration Flow instead.
- Usage Detail Integration Flow (OUCCB2OUMDM2SSUsageDetailReqEBF) is deprecated in the OUCSS 2.1.0 version. 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 3.1.1 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

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

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 3.1.1* 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.

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 3.1.1 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.

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.

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.

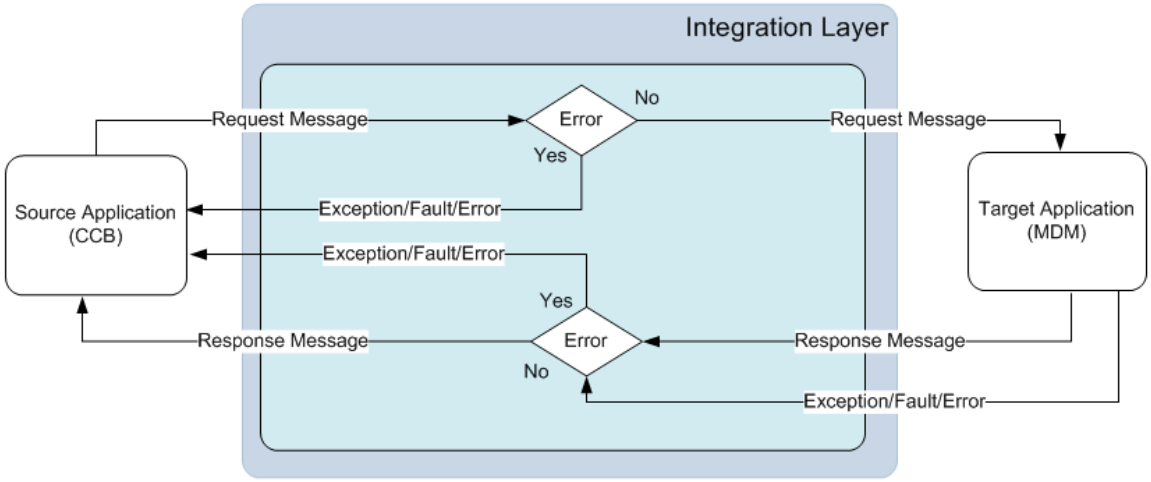
To maintain information within the domain value maps:

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 this integration are the existing DVMs for Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 3.1.1 Media Pack. No new DVMs were introduced for this integration. The lists of DVMs reused for this integration are as follows:

DVM	Integration Points	Description
OUCCB2_OUMDM2_AllowEstimate	Usage request	Transform CCB Allow Estimate to MDM Allow Estimate and vice versa.
OUCCB2_OUMDM2_BillCondition	Usage request	Transform CCB Bill Condition to MDM Bill Condition and vice versa.
OUMDM2_OUCCB2_IsEstimate	Usage request	Transform MDM Is Estimate flag to CCB Is Estimate flag and vice versa.
OUMDM2_OUCCB2_UOM	Usage request	Transform CCB UOM to MDM UOM and vice versa.
OUMDM2_OUCCB2_TOU	Usage request	Transform CCB TOU to MDM TOU and vice versa.
OUMDM2_OUCCB2_SQI	Usage request	Transform CCB SQI to MDM SQI and vice versa.
OUMDM2_OUCCB2_UsageType	Usage request	Transform the MDM usage type to CCB usage type. For example: Interval, Scalar, etc.
OUMDM2_OUCCB2_SPHowToUse	Usage request	Transform MDM SP How to Use flag to CCB SP How to Use Flag and vice versa
OUMDM2_OUCCB2_MeasuresPeakQuantity	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 3.1.1 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://sdc60025sems.us.oracle.com:8072/soa-
infra/services/default/CCBMDM2SSExtService/Service1"/>
        </port>
</service>
```

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

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, see the *Oracle Utilities Customer Care and Billing Integration to Oracle Utilities Meter Data Management Release 3.1.1 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

Usage Request Mapping

CCB Usage Request Message			MDM Usage Request Message			DVM Mapping
Element Name	**Parent Element**	**Type**	**Element Name**	**Parent Element**	**Type**	**DVM**
requestDetails		Outermost Tag	D2-CalculateUsage MultipleRequests		Outermo stTag	
mode	requestDetails	Field	mode	D2-CalculateUsage MultipleRequests	Field	
language	requestDetails	Field				
			requests	D2-CalculateUsage MultipleRequests	Group	
request	requestDetails	List	requestsList	requests	List	
			usId	requestsList	Field	
saId	request	Field	externalId	requestsList	Field	
rateSchedule	request	Field	usageGroupExternalId	requestsList	Field	

			usageGroup	requestsList	Field	
billCondition	request	Field	billCondition	requestsList	Field	
intervalProcessing	request	Group	intervalMC	requestsList	Group	
startDateTime	intervalProcessing	Field	startDateTime	intervalMC	Field	
endDateTimeFrom	intervalProcessing	Field	endDateTimeFrom	intervalMC	Field	
endDateTimeTo	intervalProcessing	Field	endDateTimeTo	intervalMC	Field	
customElements	intervalProcessing	Group				
scalarProcessing	request	Group	scalarMC	requestsList	Group	
startDateTime	scalarProcessing	Field	startDateTime	scalarMC	Field	
endDateTime	scalarProcessing	Field	endDateTimeFrom	scalarMC	Field	
			endDateTimeTo	scalarMC	Field	
billingOption	scalarProcessing	Field	endRangeOption	scalarMC	Field	OUCCB2_OUMDM2_BillingOption
minDaysOffset	scalarProcessing	Field	minDaysOffset	scalarMC	Field	
maxDaysOffset	scalarProcessing	Field	maxDaysOffset	scalarMC	Field	
allowEstimate	scalarProcessing	Field	allowEstimate	scalarMC	Field	OUCCB2_OUMDM2_AllowEstimate
estimateDate	scalarProcessing	Field	estimateDate	scalarMC	Field	
customElements	scalarProcessing	Group				
			dateBreaks	requestsList	Group	
dateBreaks	request	List	dateBreaksList	dateBreaks	List	
breakDateTime	dateBreaks	Field	dateBreak	dateBreaksList	Field	
customElements	request	Group				
			customElements	requestsList	Group	
responseDetails		Outermost Tag				
			responses	D2-CalculateUsageMultipleRequests	Group	
response	responseDetails	List	responsesList	responses	List	
saId	response	Field	externalId	responsesList	Field	
rateSchedule	response	Field	usageGroupExternalId			
usagePeriod	response	Group				
startDateTime	usagePeriod	Field	startDateTime	responsesList	Field	
endDateTime	usagePeriod	Field	endDateTime	responsesList	Field	
customElements	usagePeriod	Group				

isSkipped	response	Field	skipped	responsesList	Field	
skipReasonDescription	response	Field	skipReasonDescription	responsesList	Field	
			skipReason	responsesList	Field	
			summaryUsagePeriods	responsesList	Group	
usagePeriods	response	List	summaryUsagePeriodsList	summaryUsagePeriods	List	
startDateTime	usagePeriods	Field	startDateTime	summaryUsagePeriodsList	Field	
endtDateTime	usagePeriods	Field	endDateTime	summaryUsagePeriodsList	Field	
usageRequestType	usagePeriods	Field	usageType	summaryUsagePeriodsList	Field	OUMDM2_OUCCB2_UsageType
			SQs	summaryUsagePeriodsList	Group	
serviceQty	usagePeriods	List	SQsList	SQs	List	
seq	serviceQty	Field	sqSequence	SQsList	Field	
uom	serviceQty	Field	Uom	SQsList	Field	OUMDM2_OUCCB2_UOM
tou	serviceQty	Field	Tou	SQsList	Field	OUMDM2_OUCCB2_TOU
sqi	serviceQty	Field	Sqi	SQsList	Field	OUMDM2_OUCCB2_SQI
qty	serviceQty	Field	quantity	SQsList	Field	
customElements	usagePeriods	Group				
spUsagePeriod	response	List	summaryUsagePeriodsList	summaryUsagePeriods	List	
startDateTime	spUsagePeriod	Field	startDateTime	summaryUsagePeriodsList	Field	
endDateTime	spUsagePeriod	Field	endDateTime	summaryUsagePeriodsList	Field	
			spSQs	summaryUsagePeriodsList	Group	
serviceQty	spUsagePeriod	List	spSQsList	spSQs	List	
seq	serviceQty	Field	spSQsequence	spSQsList	Field	
spId	serviceQty	Field	spId	spSQsList	Field	
uom	serviceQty	Field	Uom	spSQsList	Field	OUMDM2_OUCCB2_UOM
tou	serviceQty	Field	Tou	spSQsList	Field	OUMDM2_OUCCB2_TOU
sqi	serviceQty	Field	Sqi	spSQsList	Field	OUMDM2_OUCCB2_SQI
qty	serviceQty	Field	quantity	spSQsList	Field	
customElements	spUsagePeriod	Group				

			customElements	summaryUsagePeriodsList	Group	
scalarProcessing	response	Group	scalarDetails	responsesList	Group	
isEstimate	scalarProcessing	Field	isEstimate	scalarDetails	Field	OUMDM2_OUCCB2_IsEstimate
reads	response	List	scalarDetailsList	scalarDetails	List	
readSeq	reads	Field	sequence	scalarDetailsList	Field	
spId	reads	Field	spId	scalarDetailsList	Field	
startReadDateTime	reads	Field	startDateTime	scalarDetailsList	Field	
endReadDateTime	reads	Field	endDateTime	scalarDetailsList	Field	
uom	reads	Field	uom	scalarDetailsList	Field	OUMDM2_OUCCB2_UOM
tou	reads	Field	tou	scalarDetailsList	Field	OUMDM2_OUCCB2_TOU
sqi	reads	Field	sqi	scalarDetailsList	Field	OUMDM2_OUCCB2_SQI
startReading	reads	Field	startMeasurement	scalarDetailsList	Field	
endReading	reads	Field	endMeasurement	scalarDetailsList	Field	
measuredQty	reads	Field	quantity	scalarDetailsList	Field	
finalQty	reads	Field	finalQuantity		Field	
finalUom	reads	Field	finalUom	scalarDetailsList	Field	
finalTou	reads	Field	finalTou	scalarDetailsList	Field	
finalSqi	reads	Field	finalSqi	scalarDetailsList	Field	
sphowToUse	reads	Field	spHowToUse	scalarDetailsList	Field	OUMDM2_OUCCB2_SPHowToUse
reghowToUse	reads	Field	mcHowToUse	scalarDetailsList	Field	
constant	reads	Field	appliedMultiplier	scalarDetailsList	Field	
measuresPeakQuantity	reads	Field	measuresPeakQuantity	scalarDetailsList	Field	OUMDM2_OUCCB2_MeasuresPeakQuantity
usePercent	reads	Field	usePercent	scalarDetailsList	Field	
customElements	reads	Group				
			customElements	scalarDetailsList	Group	
			exceptions	responsesList	Group	
exceptionInfo	response	List	exceptionsList	exceptions	List	
sequence	exceptionInfo	Field	sequence	exceptionsList	Field	
messageCategory	exceptionInfo	Field	messageCategory	exceptionsList	Field	
messageNumber	exceptionInfo	Field	messageNumber	exceptionsList	Field	
comments	exceptionInfo	Field	comments	exceptionsList	Field	
			messageParameters	exceptionsList	Group	

messageParameters	exceptionInfo	List	messageParametersList	messageParameters	List
parameterSequence	messageParameters	Field	sequence	messageParametersList	Field
messageParameterValue	messageParameters	Field	parameter	messageParametersList	Field
			parameterType	messageParametersList	Field
customElements	response	Group			
			customElements	responsesList	Group

Notes:
DVMs used in the transformation are existing DVMs used in CCB-MDM2 Integration. No new DVMs are introduced for this flow.
Language coming from CCB will be passed to MDM through the SOAP Header when invoking the MDM service.
MDM //scalarMC/endDateTimeFrom and //scalarMC/endDateTimeTo are populated accordingly:

- //scalarMC/endDateTimeFrom - concatenate CCB endDateTime and 12am

/requestDetails/request/scalarProcessing/endDateTime and -00.00.00

- //scalarMC/endDateTimeTo endDateTimeTo - concatenate CCB endDateTime and 11:59pm

/requestDetails/request/scalarProcessing/endDateTime and -23.59.59

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

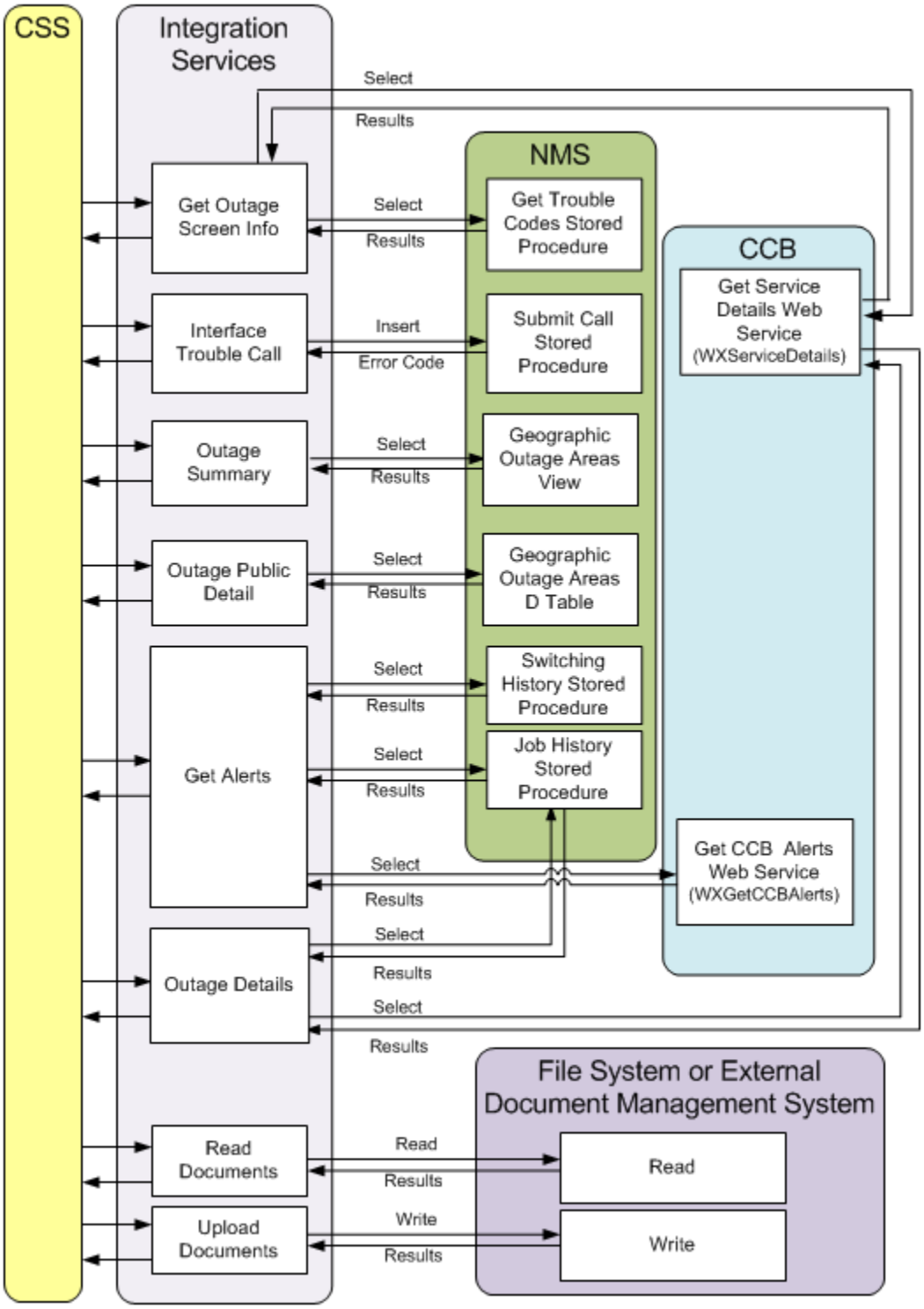
plans are generated in Oracle Utilities Network Management System, information can be provided to customers about planned outages that will impact them.

Supported Business Processes

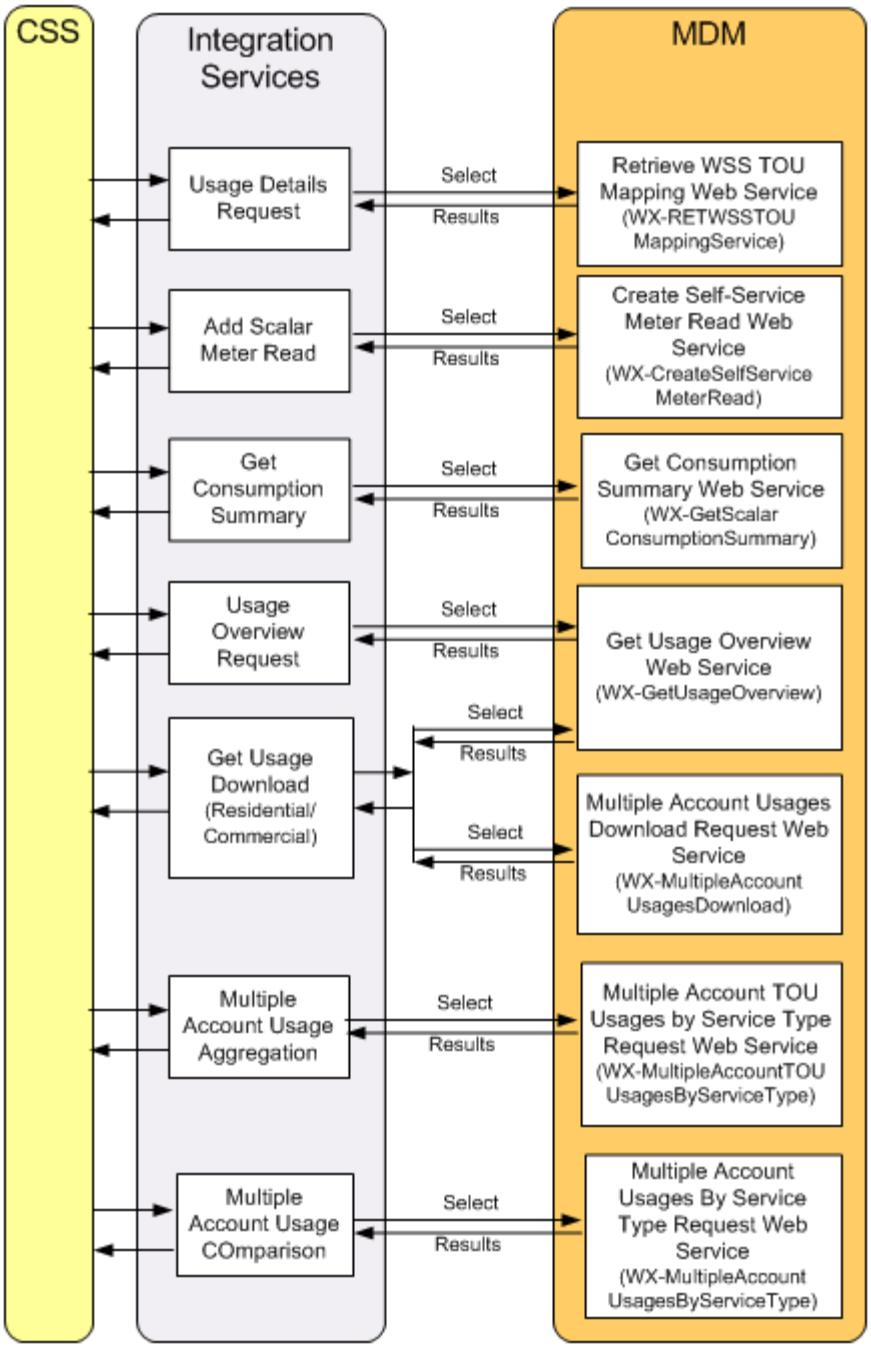
This integration between Self-Service and Meter Data Management or Network Management System is used to support the business use cases described below for the web self-service solution for Oracle Utilities.

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 Web Self-Service (WSS) 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:

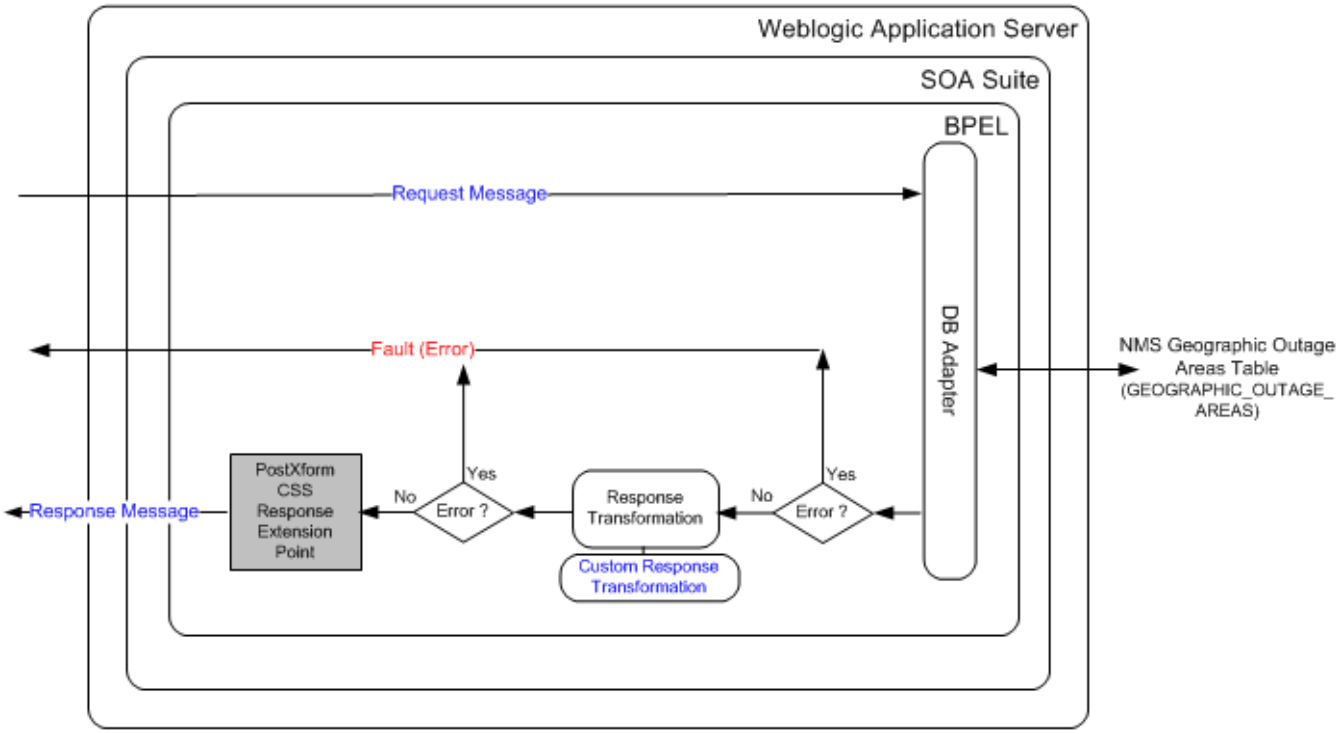
- Request Message Transformation

- 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

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

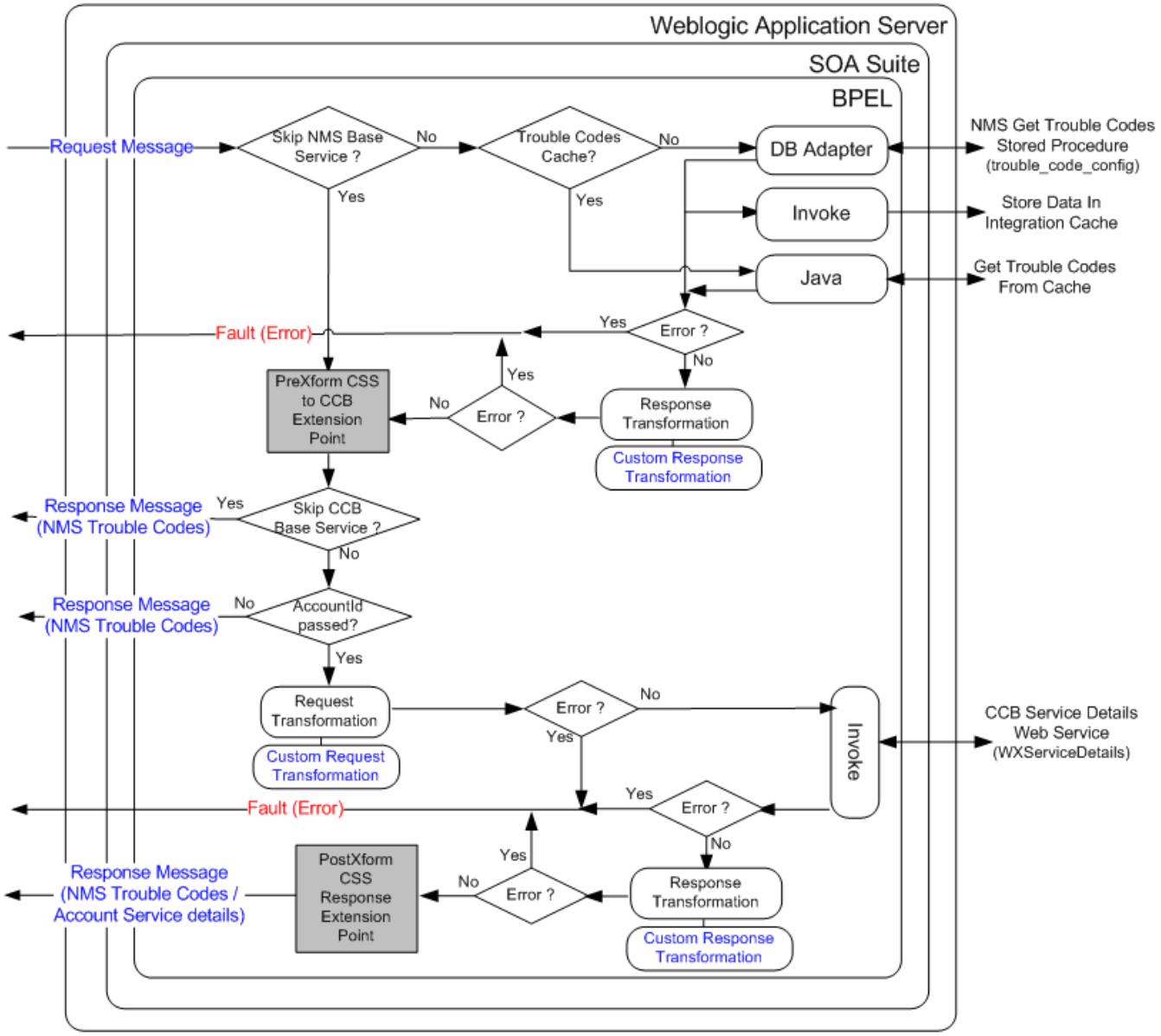
When the integration BPEL service receives a request from CSS, it will do the following:

- 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	XAI 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: o Premise 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:

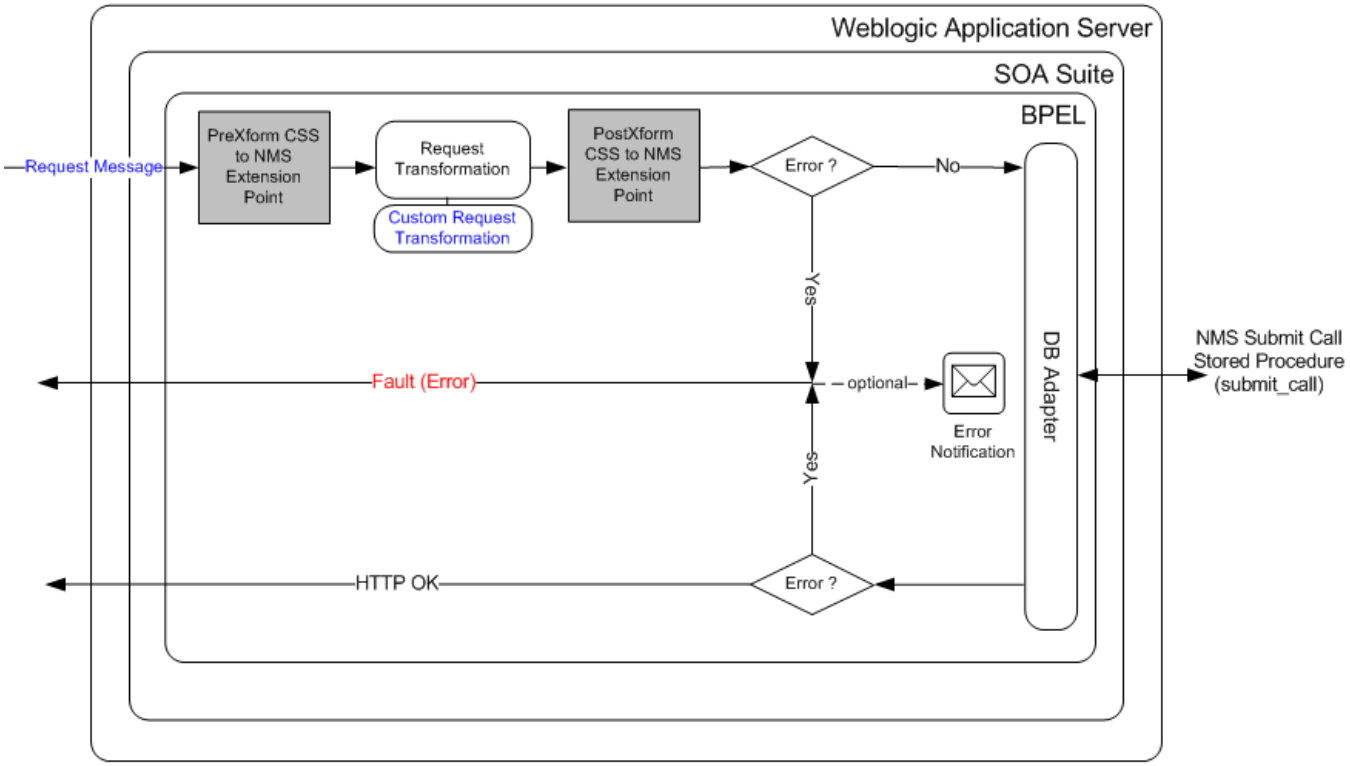
- Request Message Transformation
  - Transforming the CSS input message to the equivalent NMS input fields that will be used by the trouble calls stored procedure
- Insert Trouble Calls (Outage)
  - The Integration BPEL process will use the DB Adapter to interact with the NMS Database to invoke the NMS 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)***

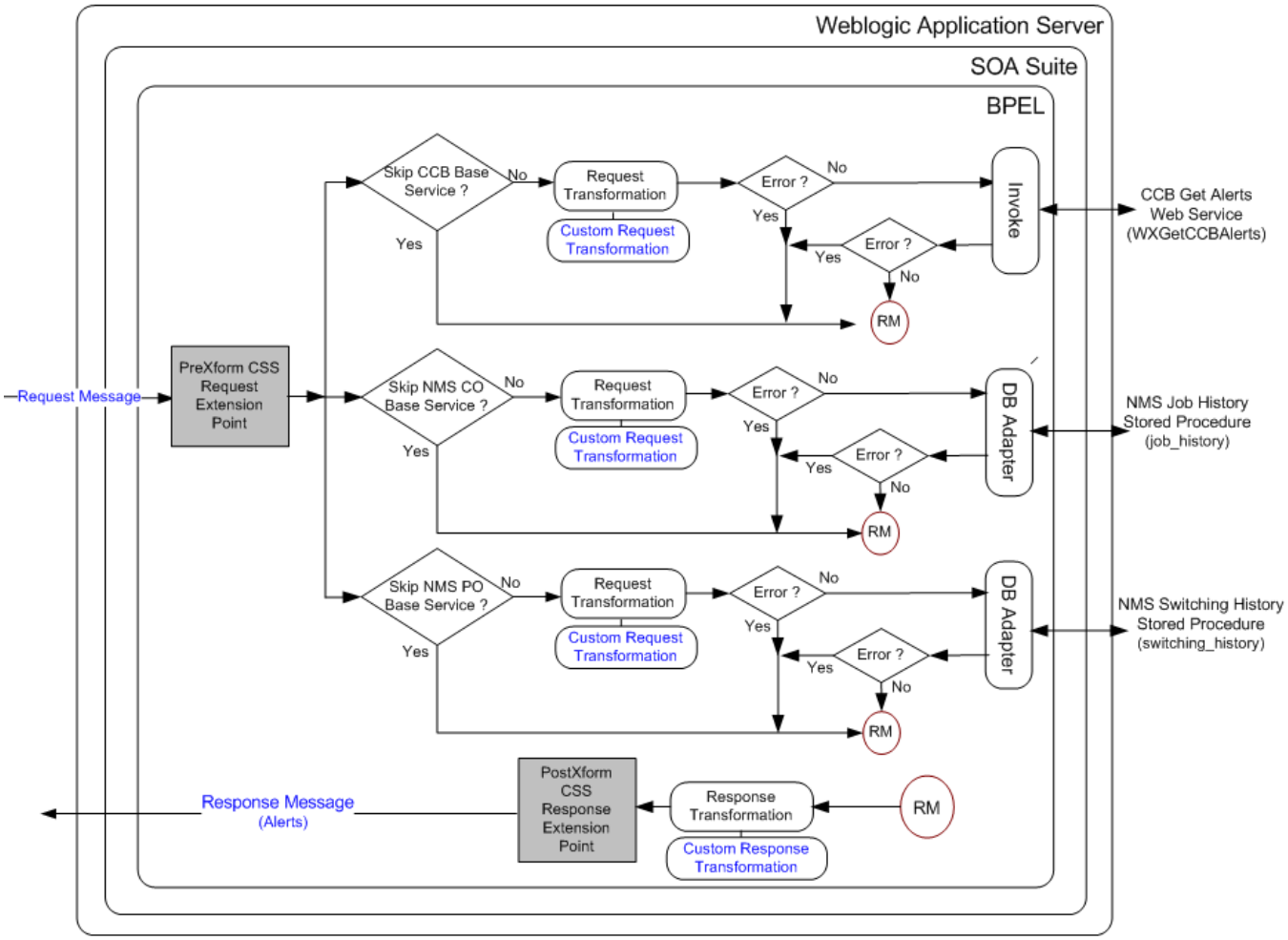
- NMS Pending Outage Alert Icon Code ***(NMS.PendingOutageAlertIconCode***)
- NMS Pending Outage Alert Link Code (***NMS.PendingOutageAlertLinkCode***)
- NMS Pending Outage Alert Type (***NMS.PendingOutageAlertType***)

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	XAI 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:

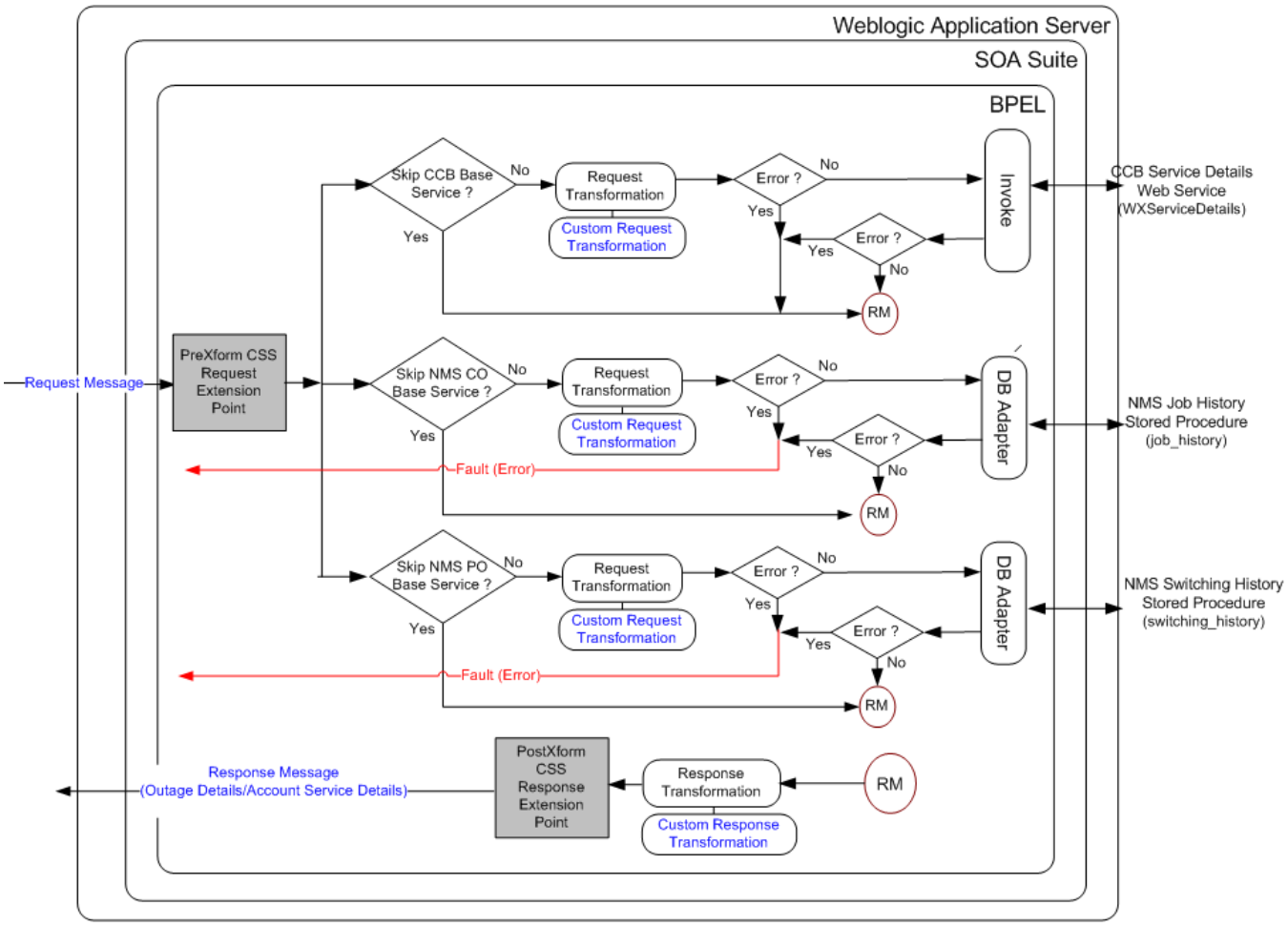
- Check Skip CCB Base Service Flag
  - The Skip CCB Base Service Flag (***CCB.SkipBaseServiceFlag***) is defined in the ConfigurationProperties xml file. The value is defaulted to false so the CCB Get Service Details web service can be invoke to get the account and service information of the account 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	XAI 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.

Outage Public Detail Integration Flow

Business Details

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).

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

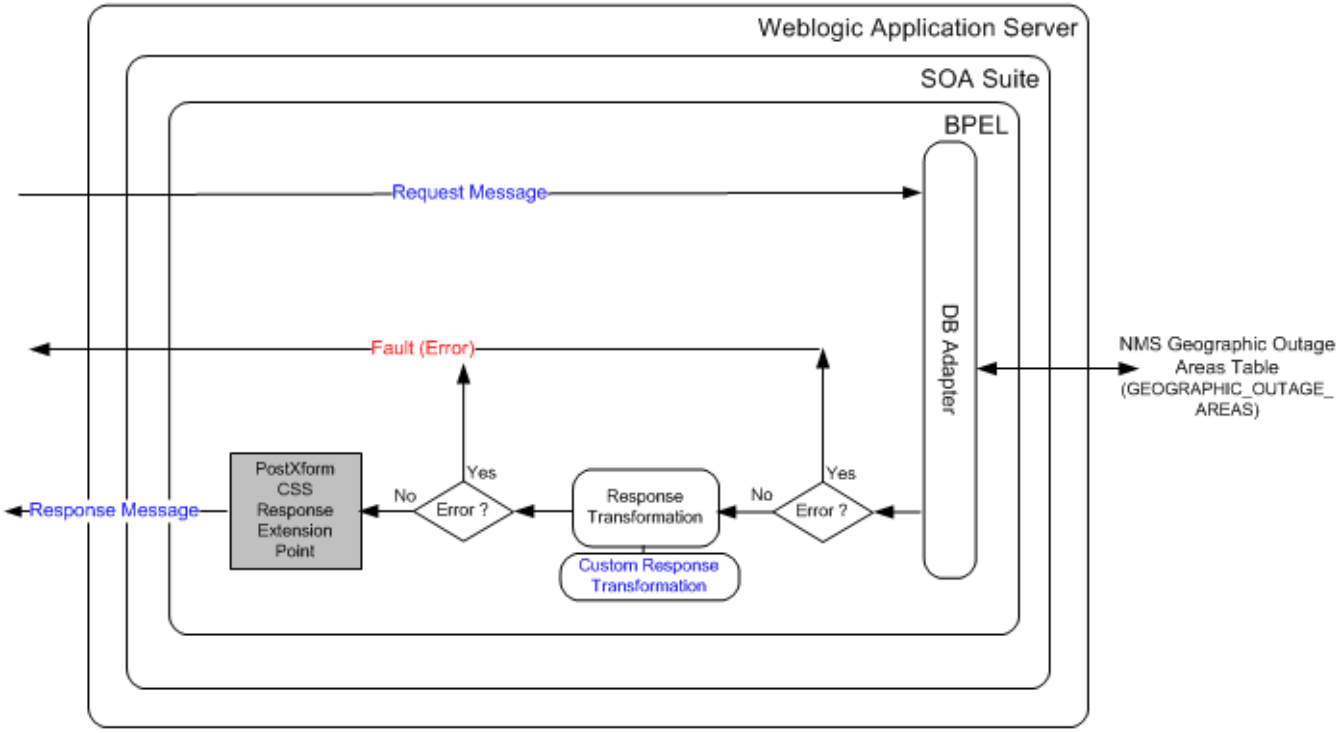
Technical Details

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_D materialized View 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
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

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

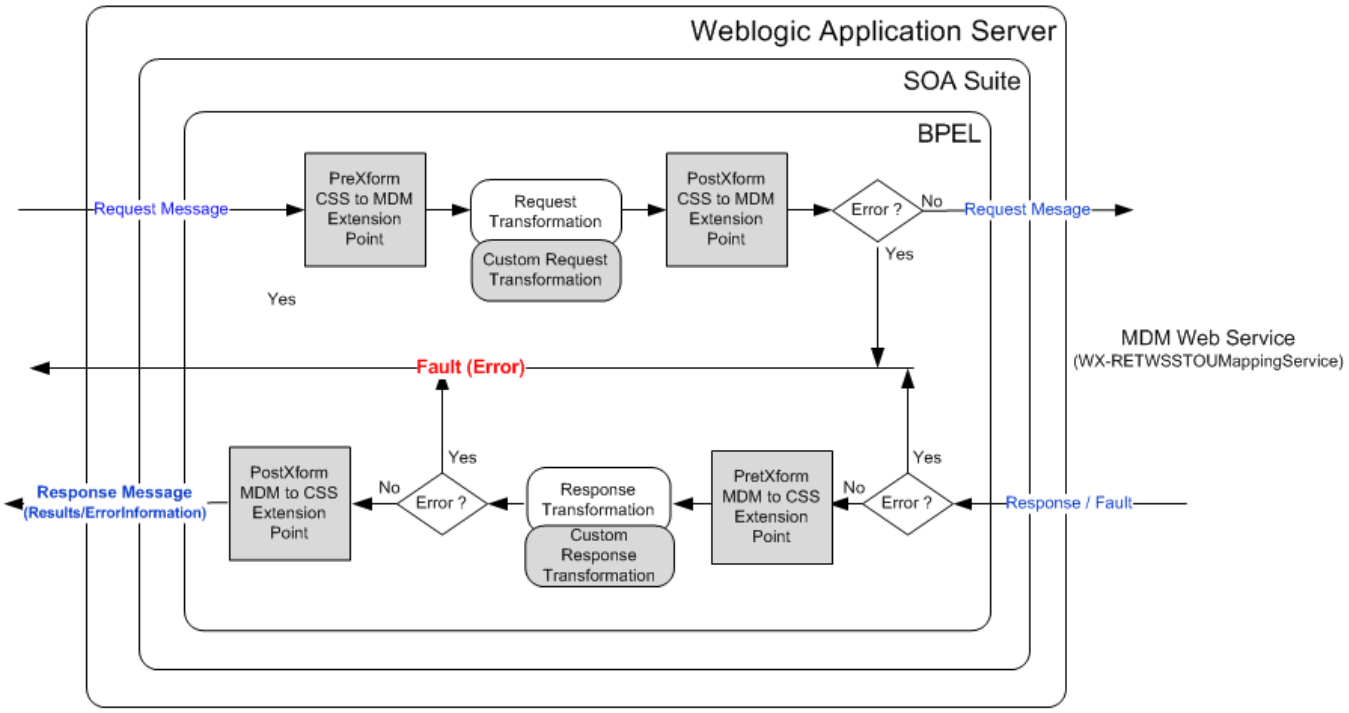
Technical Details

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	XAI 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

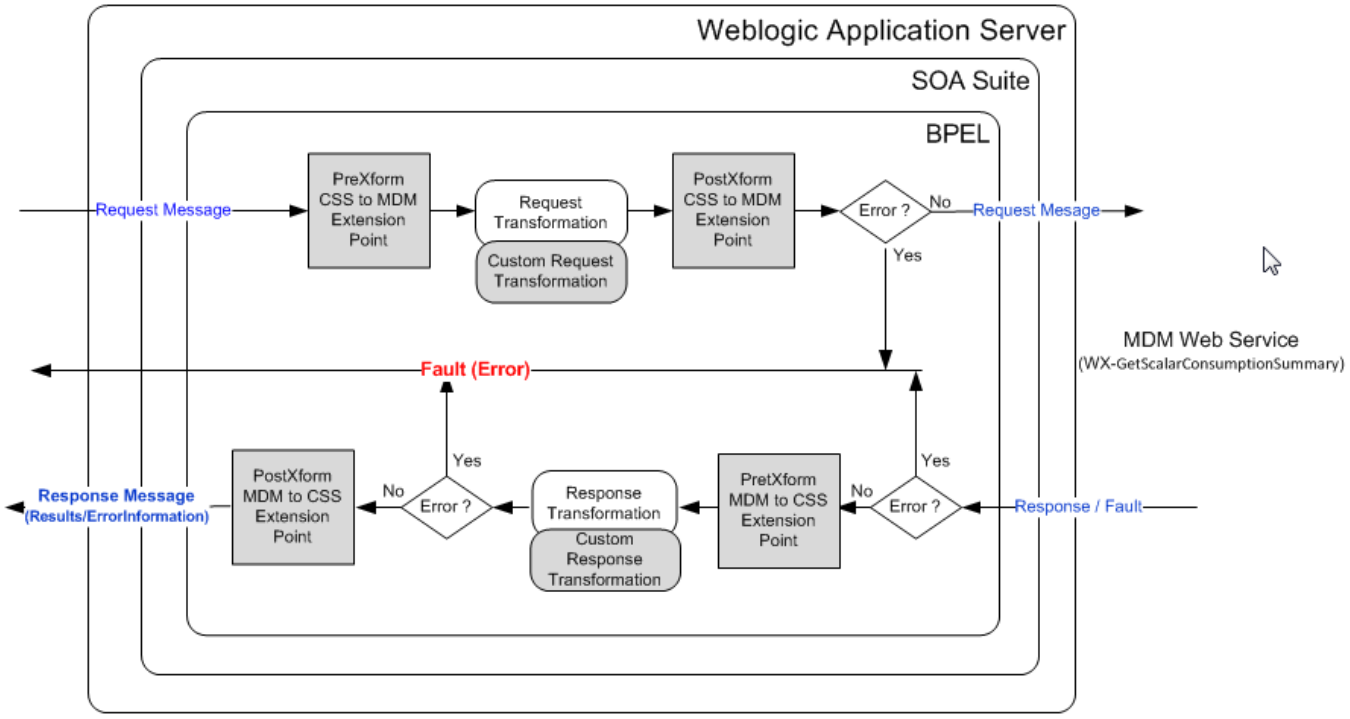
Technical Details

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	XAI 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.

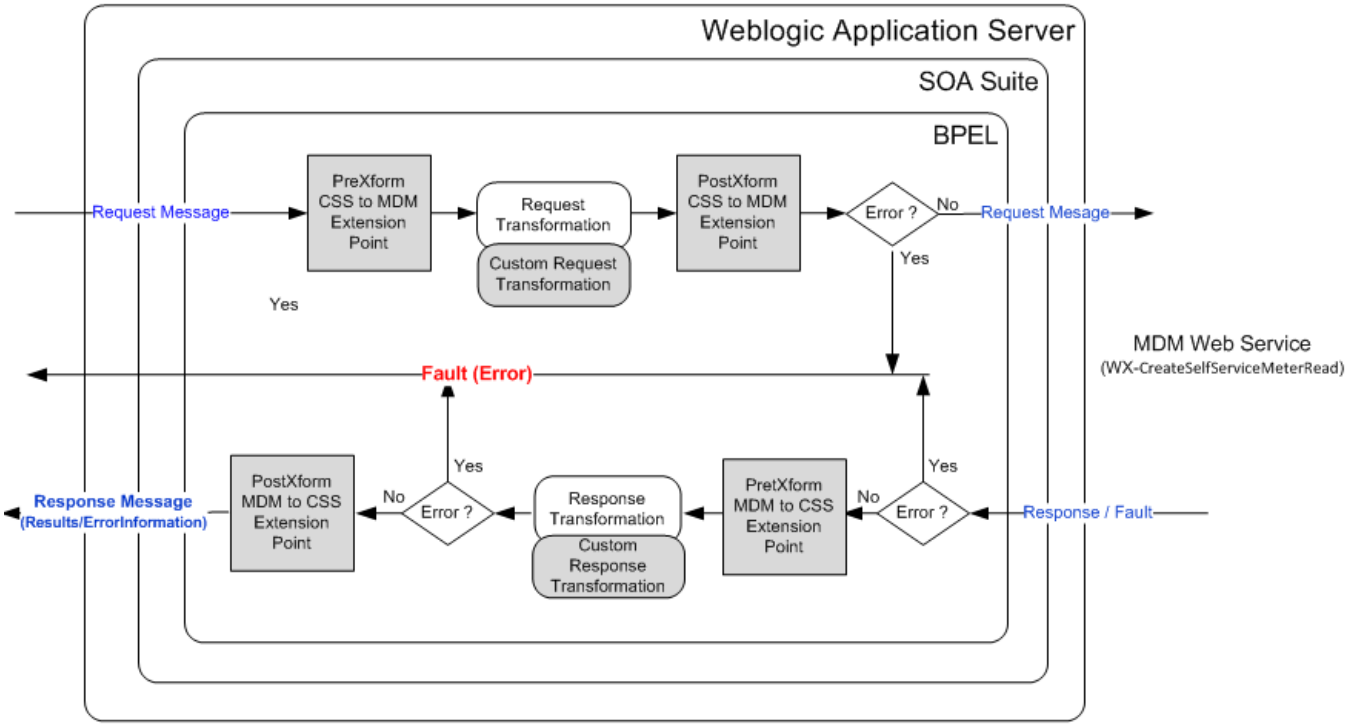
Technical Details

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	XAI 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.

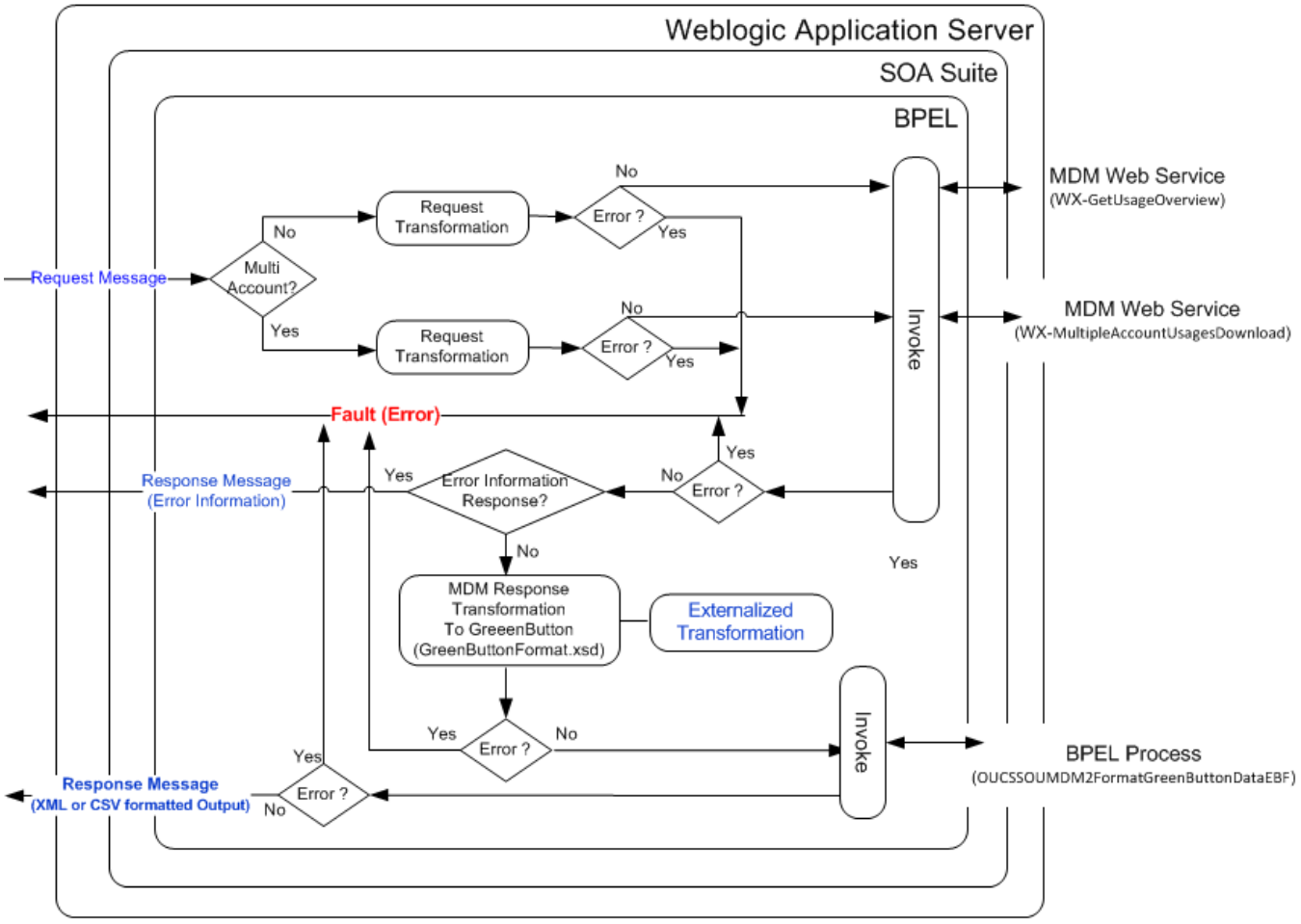
Technical Details

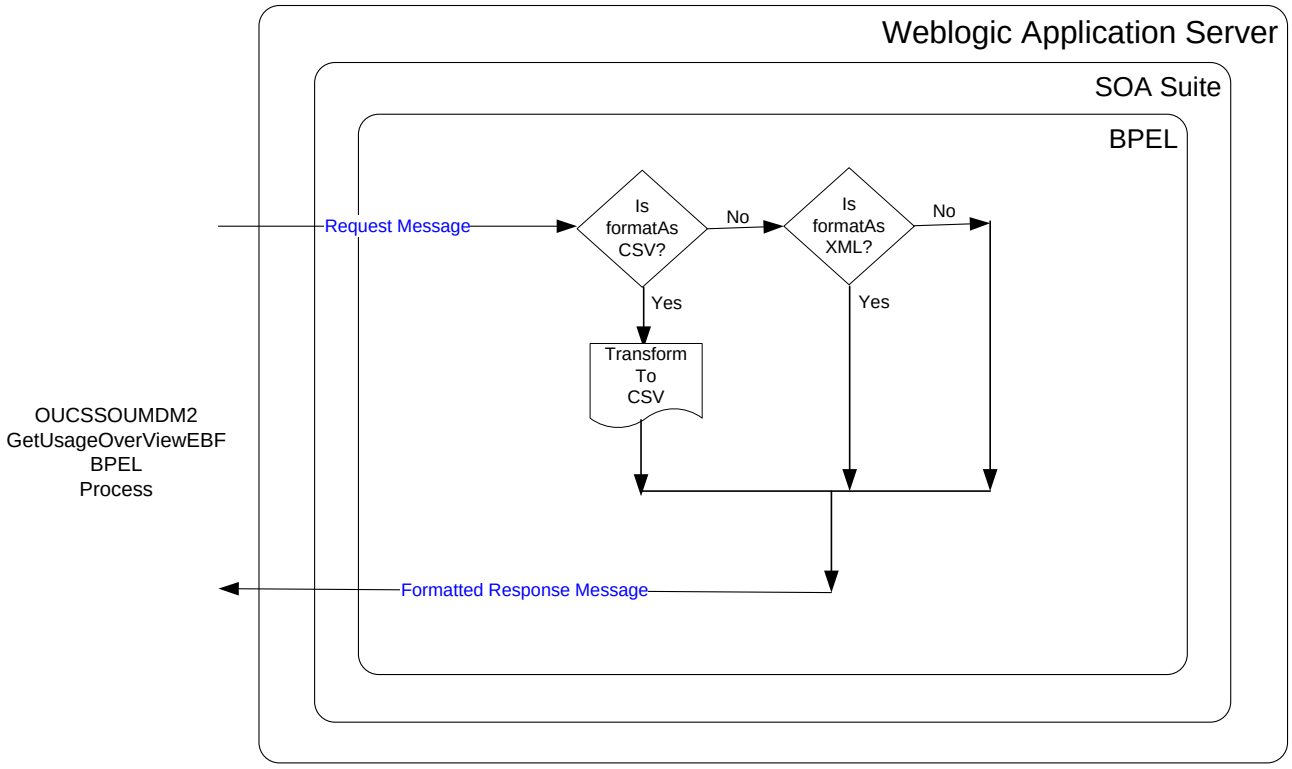
When the integration BPEL service receives a request from CSS, it will do the following:

- 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	XAI 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

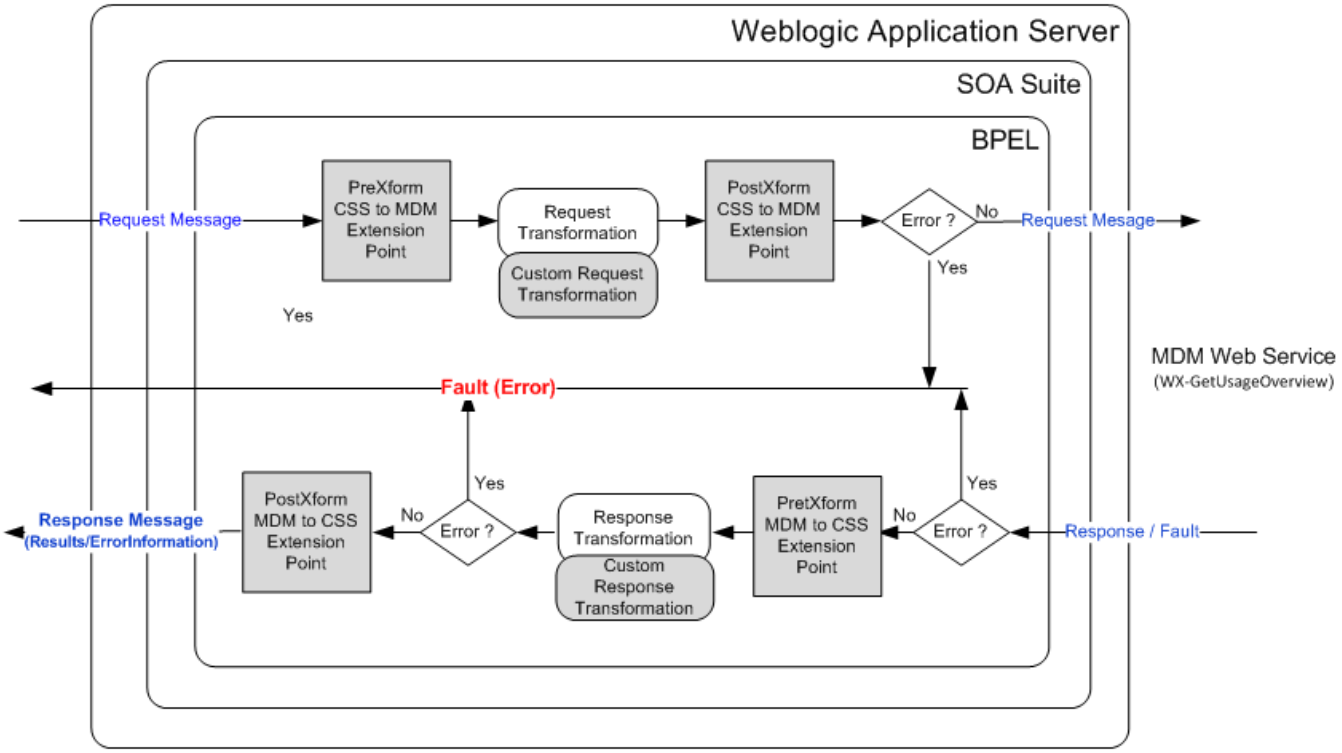
Technical Details

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	XAI 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.

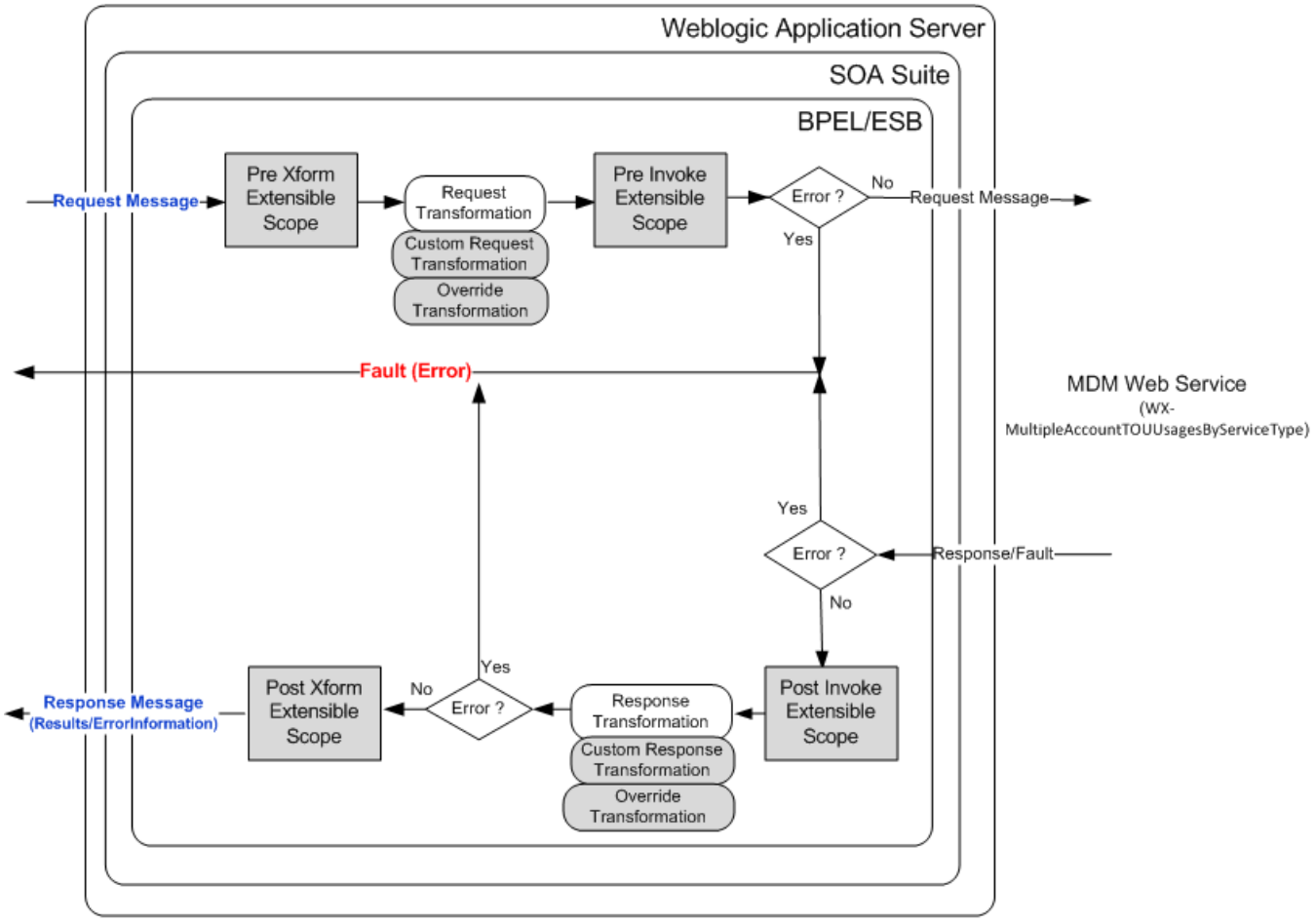
Technical Details

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	XAI 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.

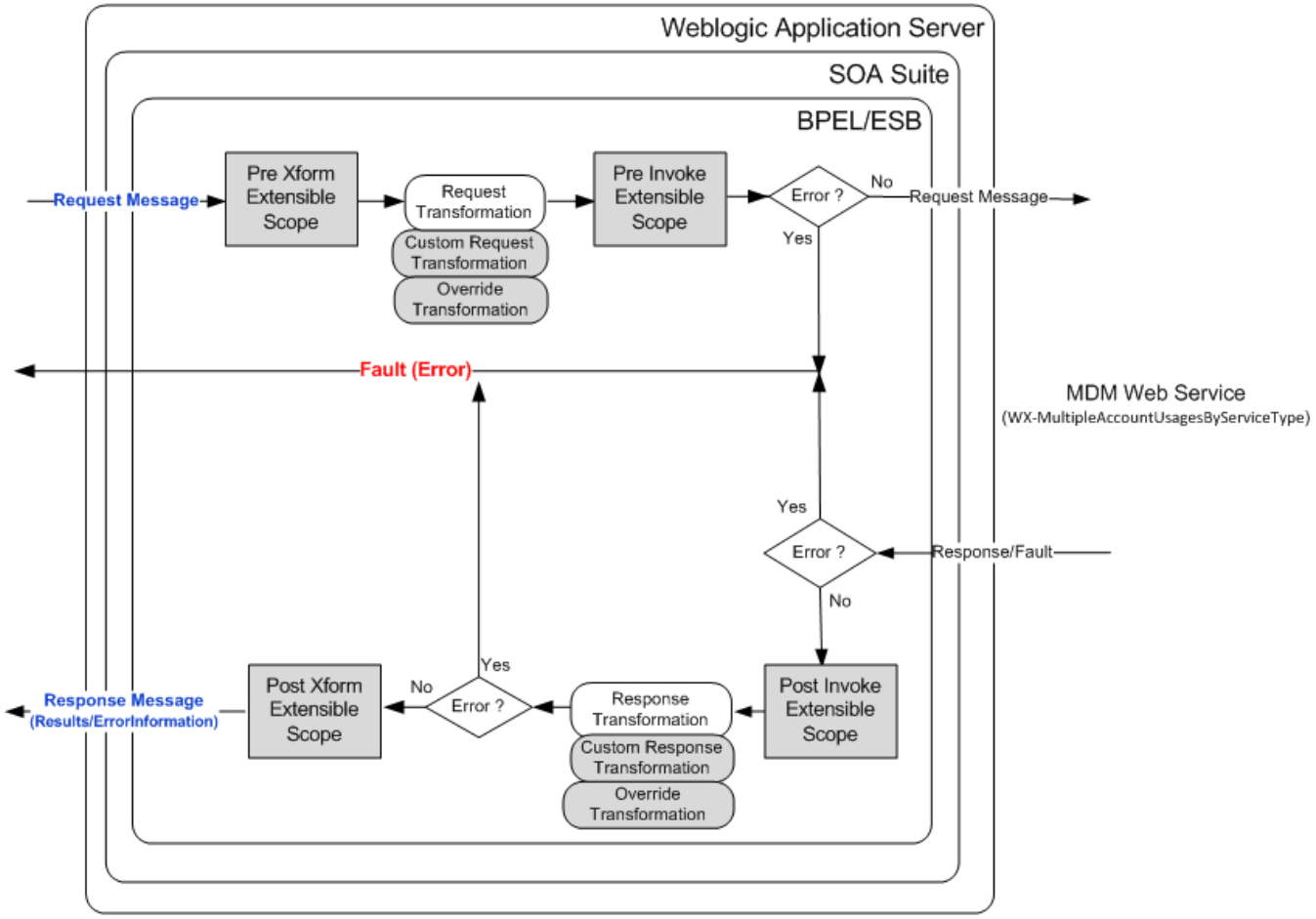
Technical Details

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	XAI 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.

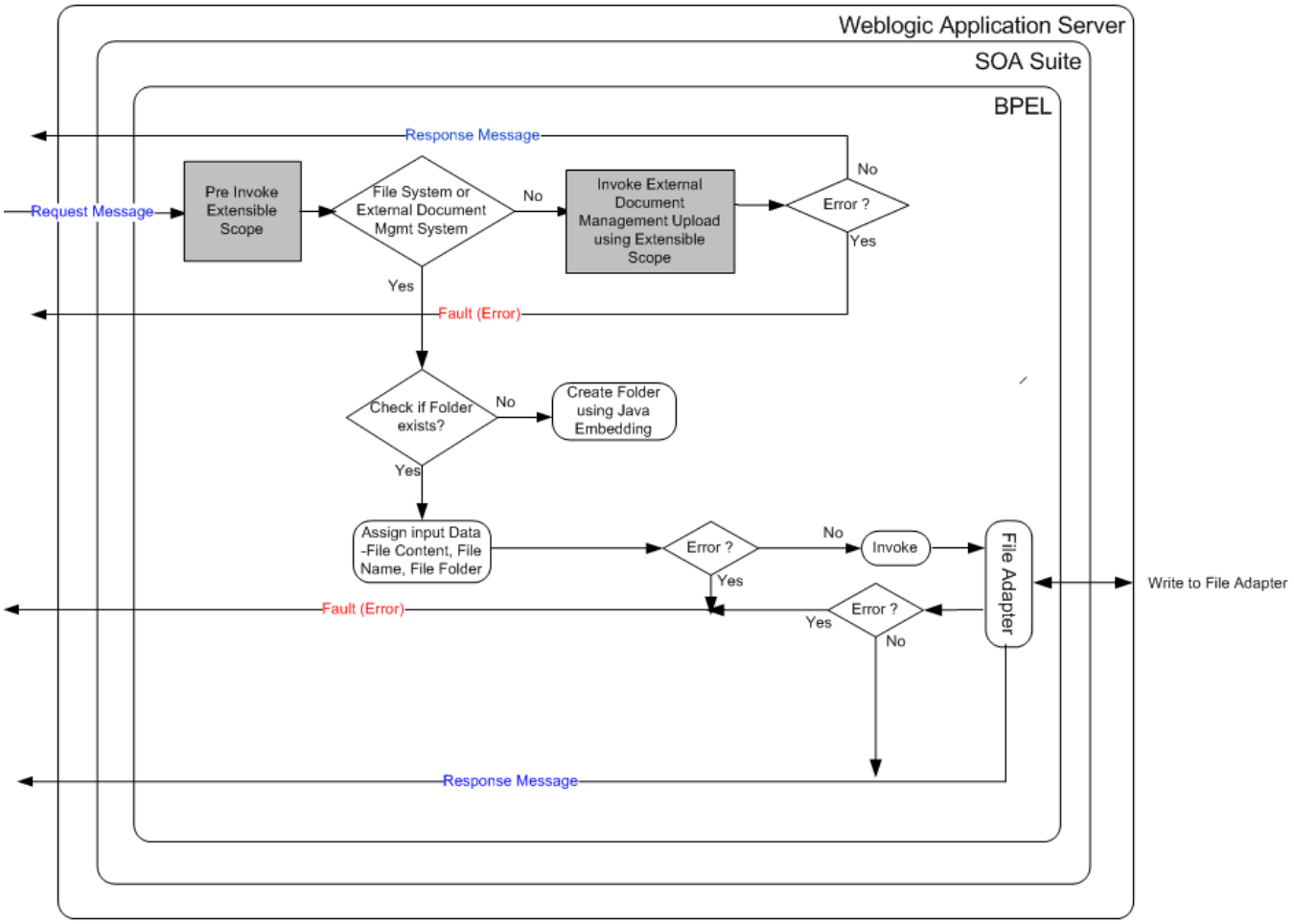
Technical Details

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.

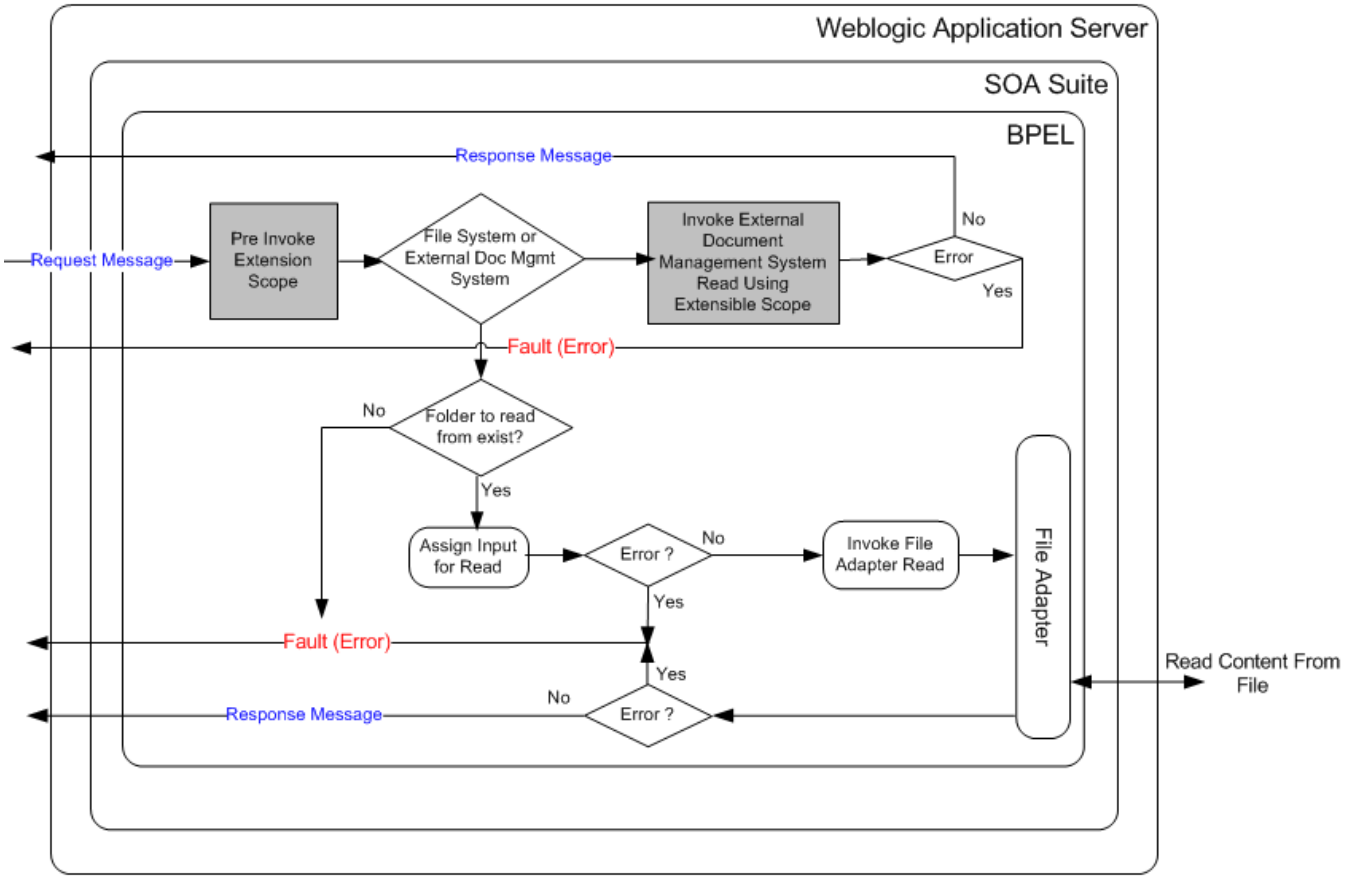
Technical Details

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 3.1.1* 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 3.1.1* 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.
- Service Configurations

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 Errorhandling 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 theURL 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/XAIApp/xaiserver/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.protoc

		ol://@EdgeApplications.OUCCB.ManagedServer.hostname:@EdgeApplications.OUCCB.ManagedServer.portnumber/@EdgeApplications.OUCCB.ManagedServer.context/XAIApp/xaiserver/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/XAIApp/xaiserver/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/XAIApp/xaiserver/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.OUMDM.ManagedServer.hostname:@EdgeApplications.OUMDM.ManagedServer.portnumber/@EdgeApplications.OUMDM.ManagedServer.context/XAIApp/xaiserver/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/XAIApp/xaiserver/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 OUCSSOUMDM2FormatGreenButtonData

		EBF is deployed. Shipped with this value: @SOAHostname.protocol://@SOAHostname:@SOAPortnumber/soa-infra/services/OUCSS/OUCSSOUMDM2FormatGreenButtonDataEBF/OUCSSOUMDM2FormatGreenButtonData_client_ep During install, the OUCSSOUMDM2FormatGreenButtonData EBF 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/XAIApp/xaiserver/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/XAIApp/xaiserver/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/XAIApp/xaiserver/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/XAIApp/xaiserver/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/XAIApp/xaiserver/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.FileSystemOrExternalDocumentManagementSystem	FILE	This value determines if the documents are stored in a file system or in an external 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/XAIXapp/xaiserver/WXGetCCBAlerts/WXGetCCBAlertsExt
ension/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/XAIXapp/xaiserver/WXGetCCBAlerts/WXGetCCBAlertsExt
ension/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://sdc60024sems.us.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 file 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:

A 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>
```

B 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 account_number	accountId	This is the customer's account Id. 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	WXServiceDetail		Outermos		

		Tag	s		t 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 UsagesByServiceType		OutermostTag	WX-MultipleAccount UsagesByServiceType		OutermostTag
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

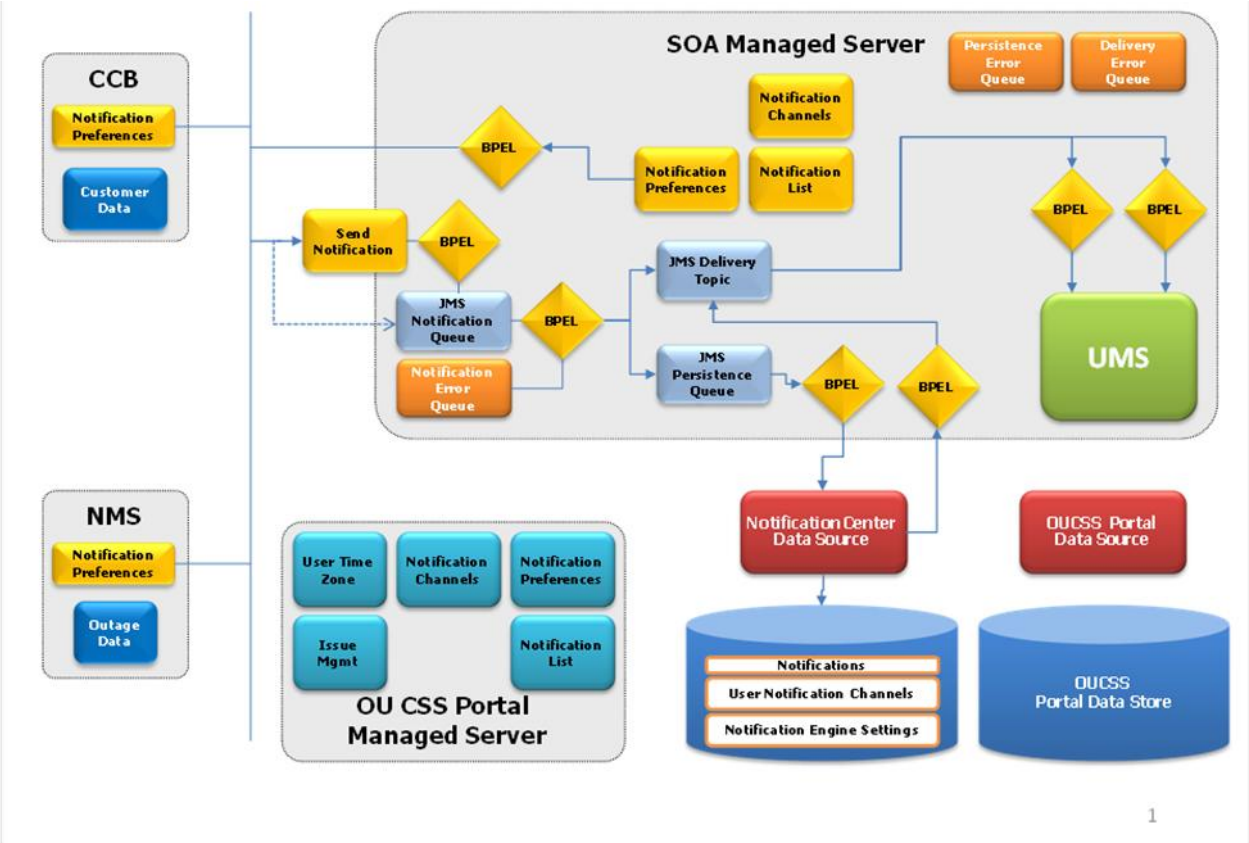
Chapter 8

Notifications

The Oracle Utilities Notification Center is pre-integrated with OUCSS, OUCCB, and 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.

Notification Center Architecture



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

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 assorted SMS number, email addresses, etc where notifications can be delivered.

Preferences	OUCSS captures the Notification Preferences for supported notification types. The list of supported notification types for each of the edge Applications NMS and CCB are retrieved dynamically. The Edge Applications may require additional information for a notification type.
Inbox	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.

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, 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	It is 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. The notification types for the preferences are retrieved dynamically for CCB and from configuration properties for NMS	WXSetNotificationPreferences	OUNCWXNotificationPreferences	
Inbox	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.		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 invoke the DB Adapter service OUNCDeleteUserChannelDBAdapter to delete the user channel.
   - 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.

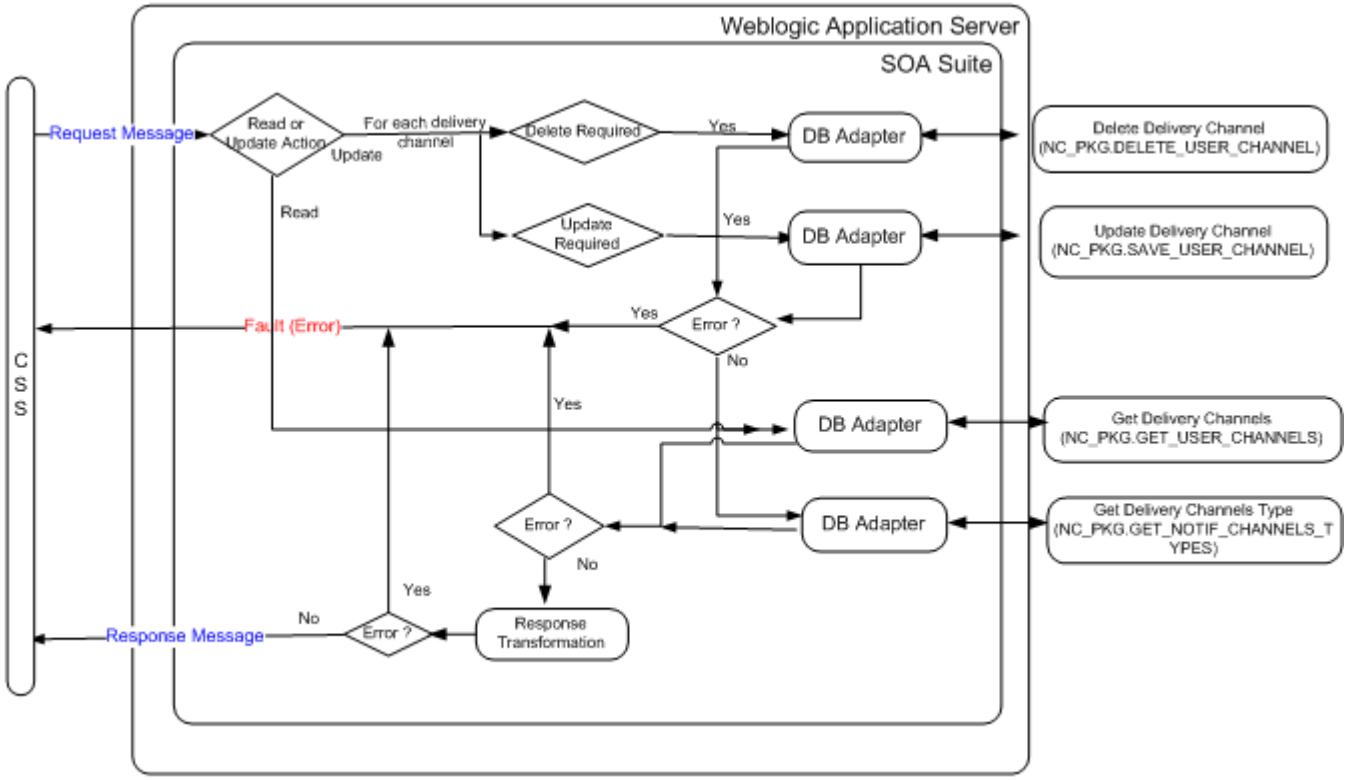
- **Error Handling**

  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.
  - 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	It 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.

The notification types for the preferences are retrieved dynamically for CCB and from configuration properties for NMS.

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
   - **A** Assign the CCB parameters required to invoke CCB to retrieve CCB Notification Types, notification options and available delivery types
   - **B** Invoke CCB Get Preferences with Read Action.
   - **C** Go to Step 4
3. If Request message Action is Update, for each user notification preference type :
   - **A** Check if User Preference needs to be Deleted, if yes invoke the DB Adapter service OUNCUserPrefDeleteDBAdapter to delete the user account preference.
   - **B** Invoke the DB Adapter service OUNCUserPrefSaveDBAdapter to update user preferences
   - **C** 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.
   - **D** Check if the CCB Enabled flag from Configuration Properties is true. If True:

i Transform message to a format to invoke CCB web service

ii Invoke CCB XAI Inbound web service WXSetNotificationPreferences

4 Get Channel Types using the DB Adapter Service OUNCChannelTypesDBAdapter.

5 Get Notification Preferences from the db using the DB Adapter Service OUNCGetNotifLocalPrefsDBAdapter

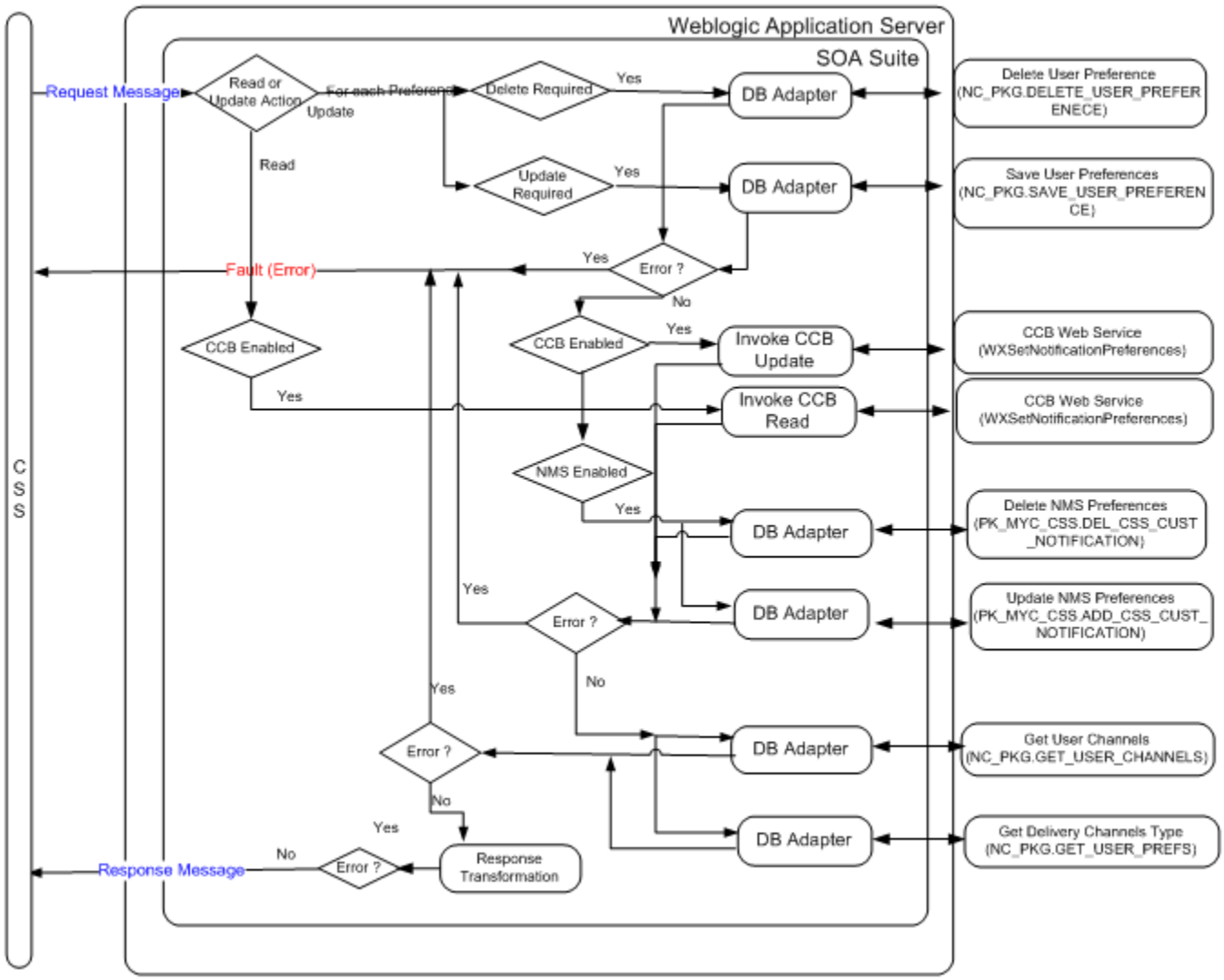
6 Transform the message to a format that needs to be returned to CSS as response to the message using the outputs from Step 3, 4 and 5 as inputs to the transformation.

7 Send 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.
  - PreInvokeCallCCBSetNotifPreferences extension scope is invoked before the CCB web service is invoked.
  - PreInvokeGetUserChannels extension scope is invoked before getting the user channels.
  - PreInvokeGetNotifPrefs extension scope is invoked before getting the notification preferences.
  - 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 notification types for the preferences are retrieved dynamically for CCB and from configuration properties for NMS

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

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
4. Get User Notifications List using the DB Adapter Service OUNCNotifListDBAdapter.
5. Get Notification Channels using the DB Adapter Service OUNCGetNotifChannelsDBAdapter.
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**

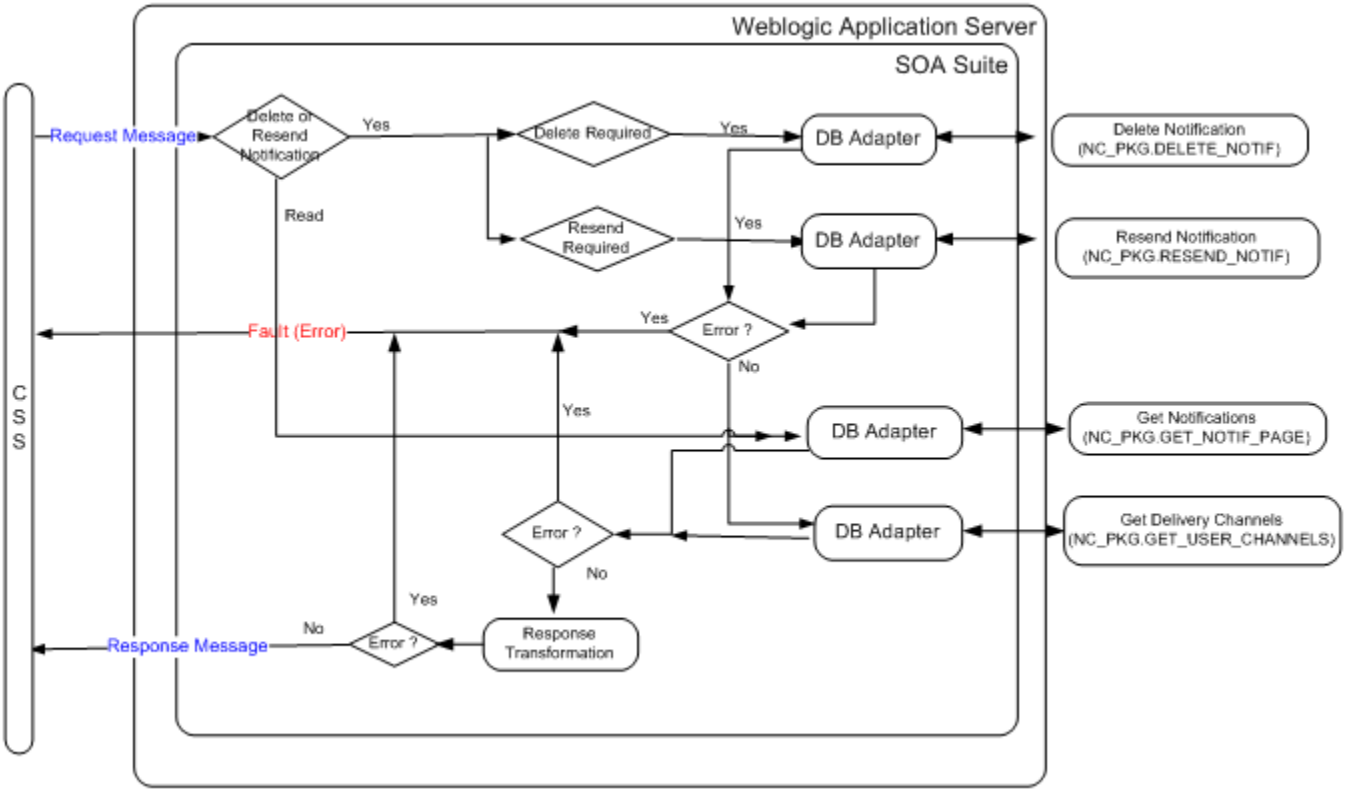
  When a business or technical error is encountered in the Integration, a SOAP fault will be returned to CSS with a specific business or technical message code. The message codes are obtained from the configuration properties file.

- **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

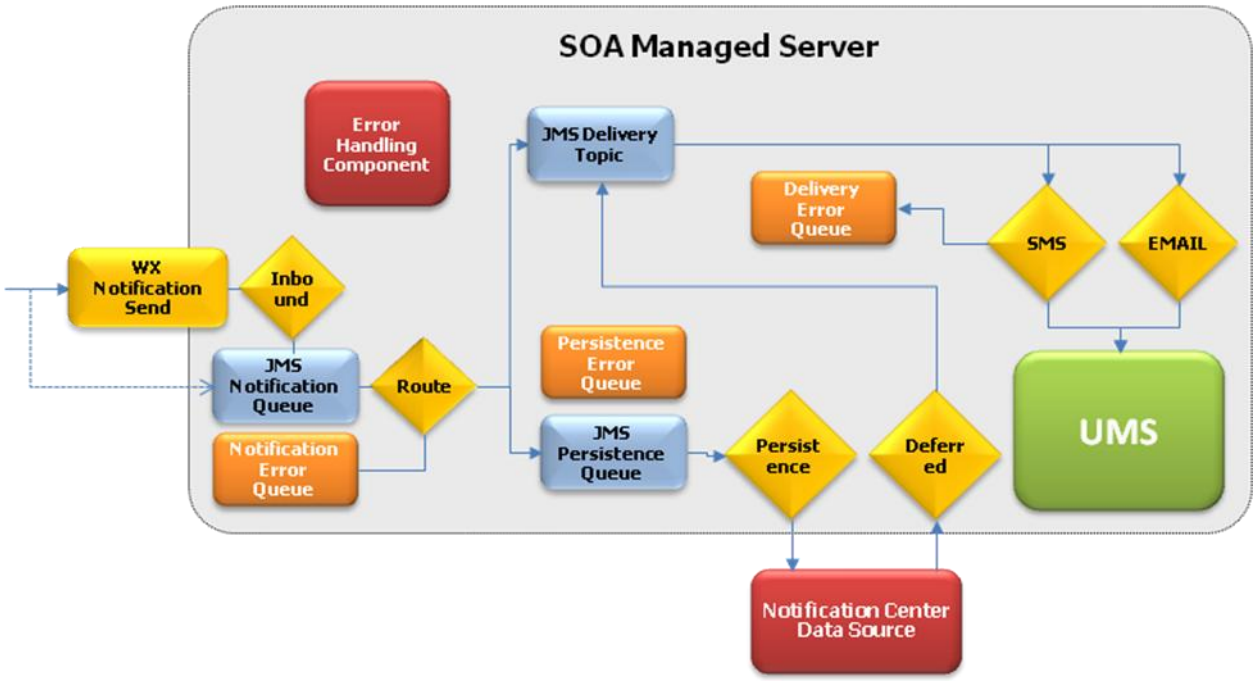
Name	Description
OUNCWXGetNotifications	This service retrieves all notifications sent out from edge application for a given account. Notifications previously sent from edge applications are retrieved from the OUNC database.

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	NC_PKG.RESEND_NOTIF

	to resend a specific notification.
OUNCNotifListDBAdapter	This adapter service invokes the plsql package NC_PKG.GET_NOTIF_PAGE to get the notification list.
OUNCGetNotifChannelsDBAdapter	This adapter service invokes the plsql package NC_PKG.GET_USER_CHANNELS to get the 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 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*.

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.

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.

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).

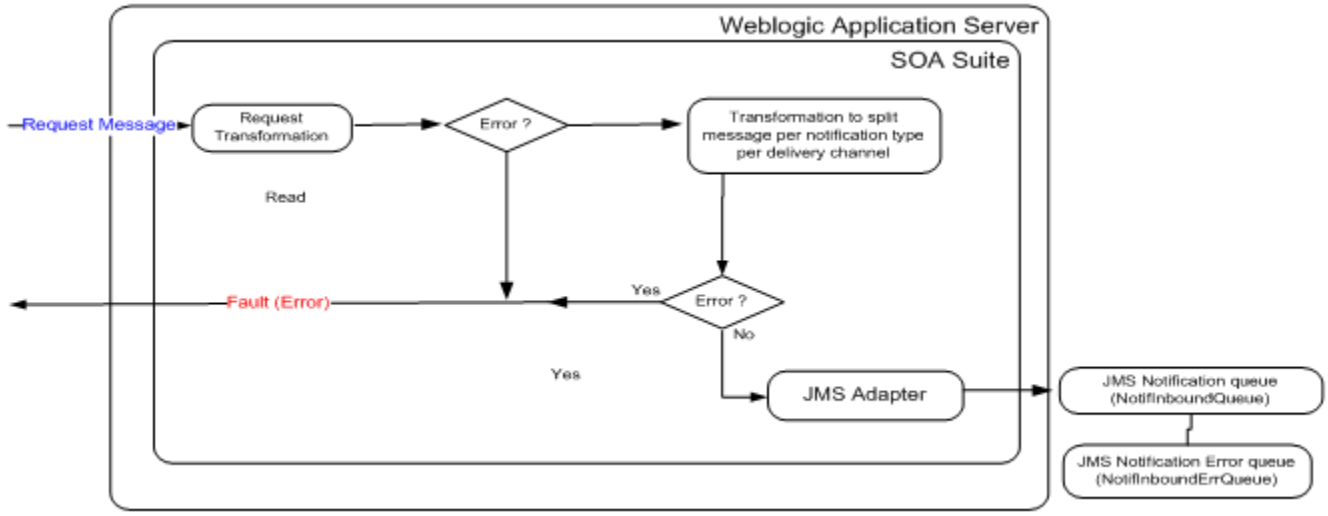
- **Error Handling**

  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. This BPEL process 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

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.

- 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.

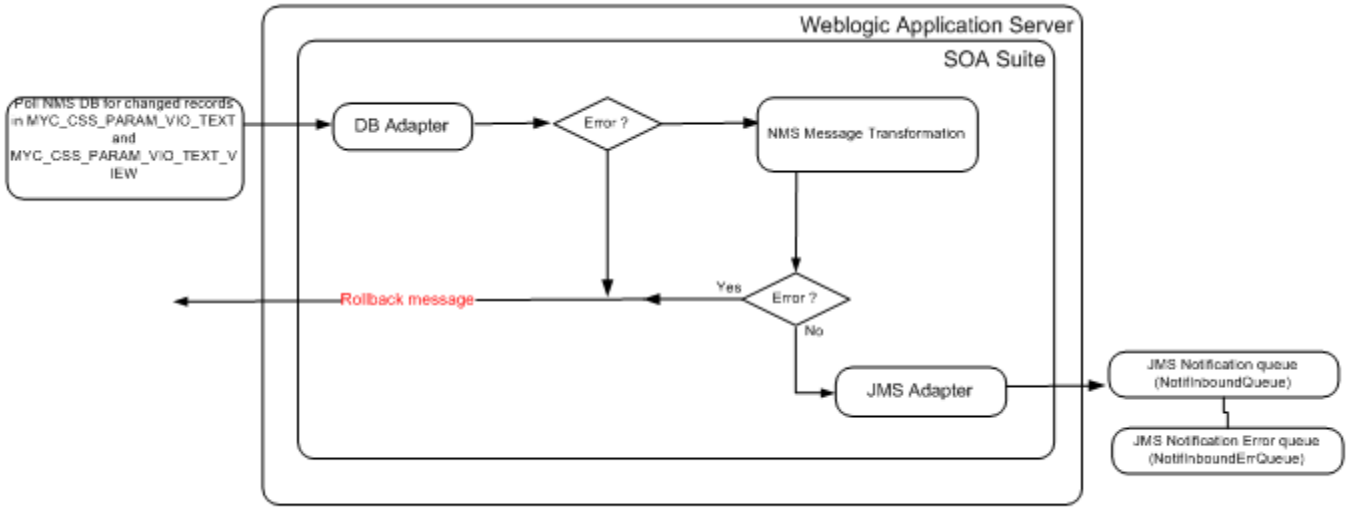
- **Customization**

  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 they are read from NMS db.	Poll NMS DB for changed records in MYC_CSS_PARAM_VIO_TEXT and MYC_CSS_PARAM_VIO_TEXT_VIEW

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:

- Consume message

  OUNCRetrieveInboundMessage consumes message from the 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.

- 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 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 into the Notification Persistence Queue (NotifPersistenceQueue)

- **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.

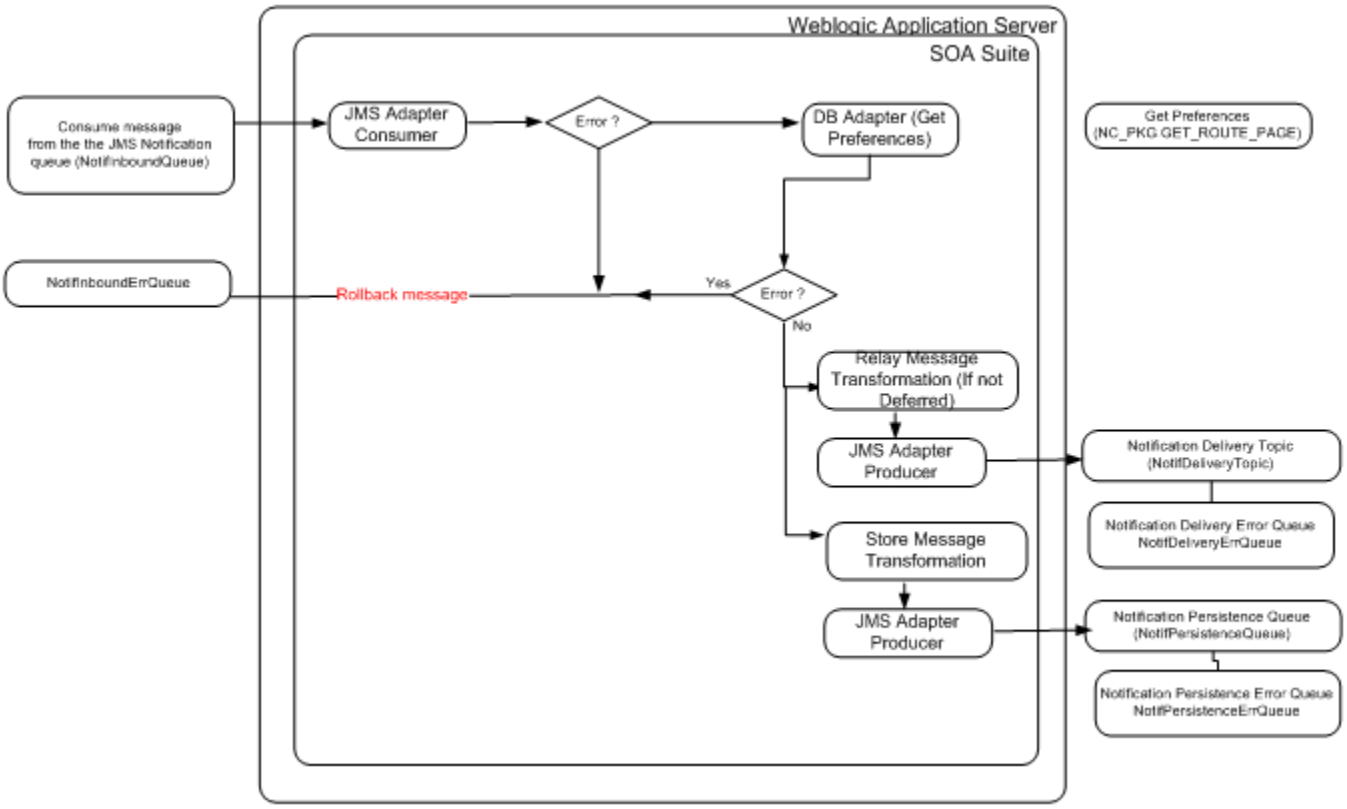
- **Customization**

  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

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 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**

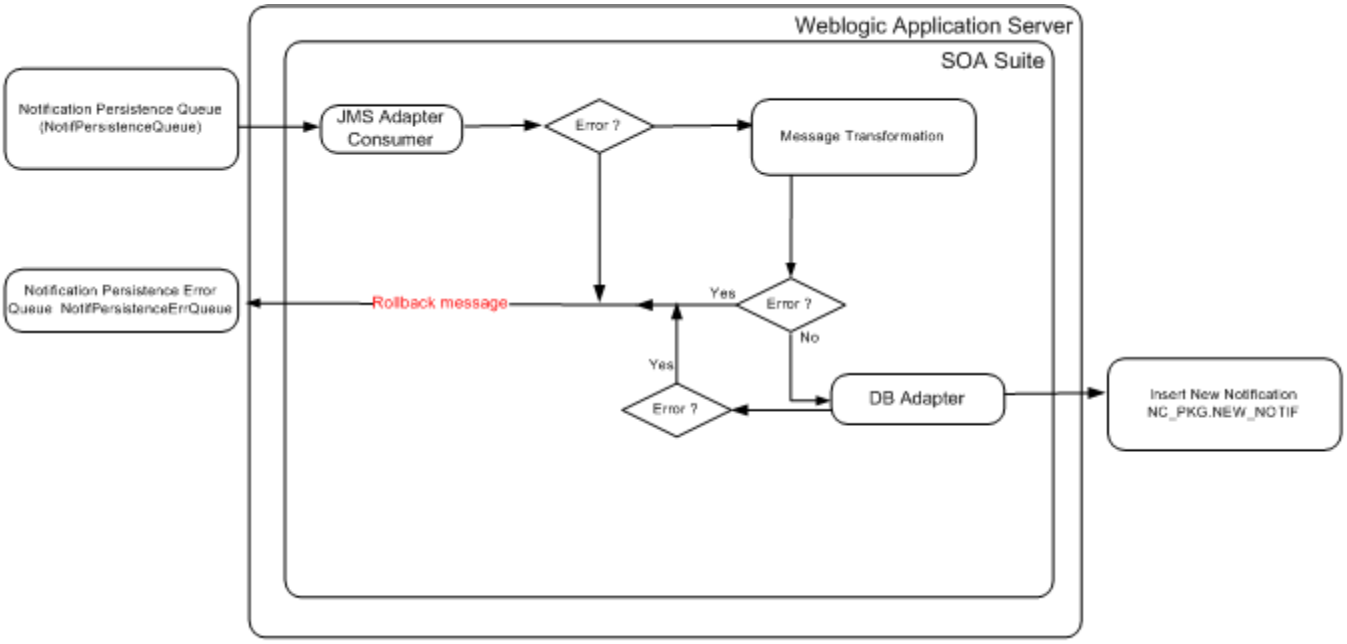
  When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **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 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**

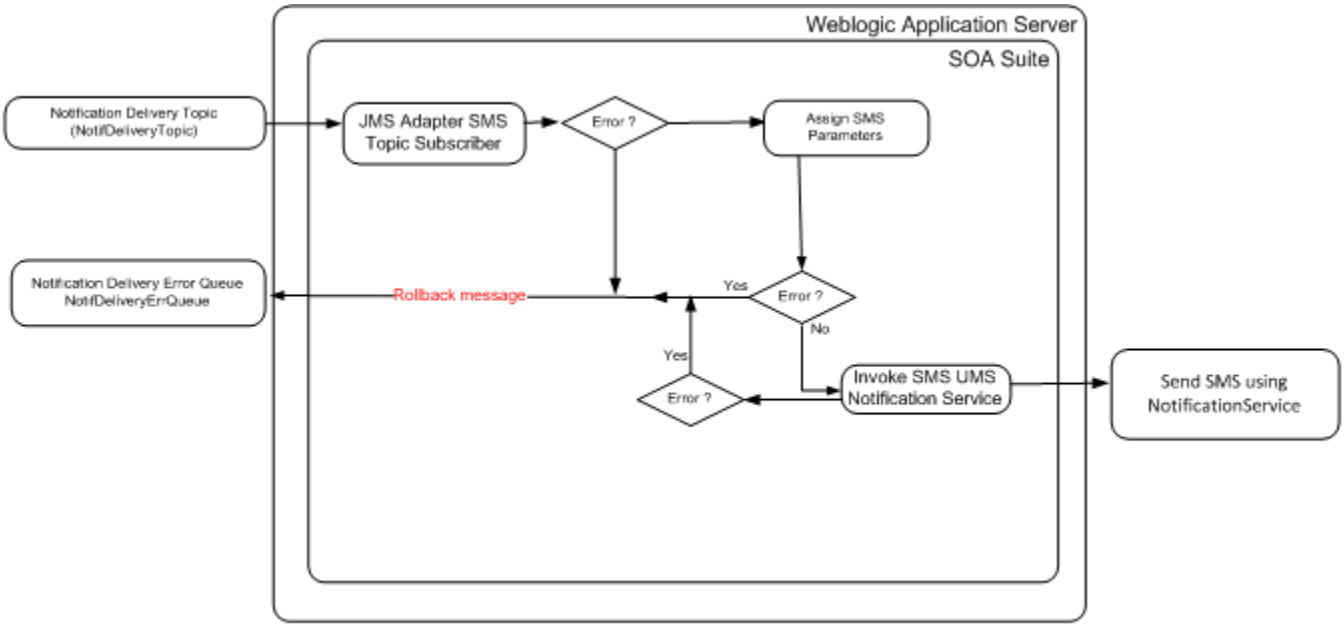
  When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **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 the JMS Notification Delivery Topic (NotifDeliveryTopic).

- Assign Email Parameters

  Assign the correct values into Email parameters needed to invoke Notification service using UMS

- Invoke Notification Service

  Invoke notification service using UMS to send the actual Email message.

- **Error Handling**

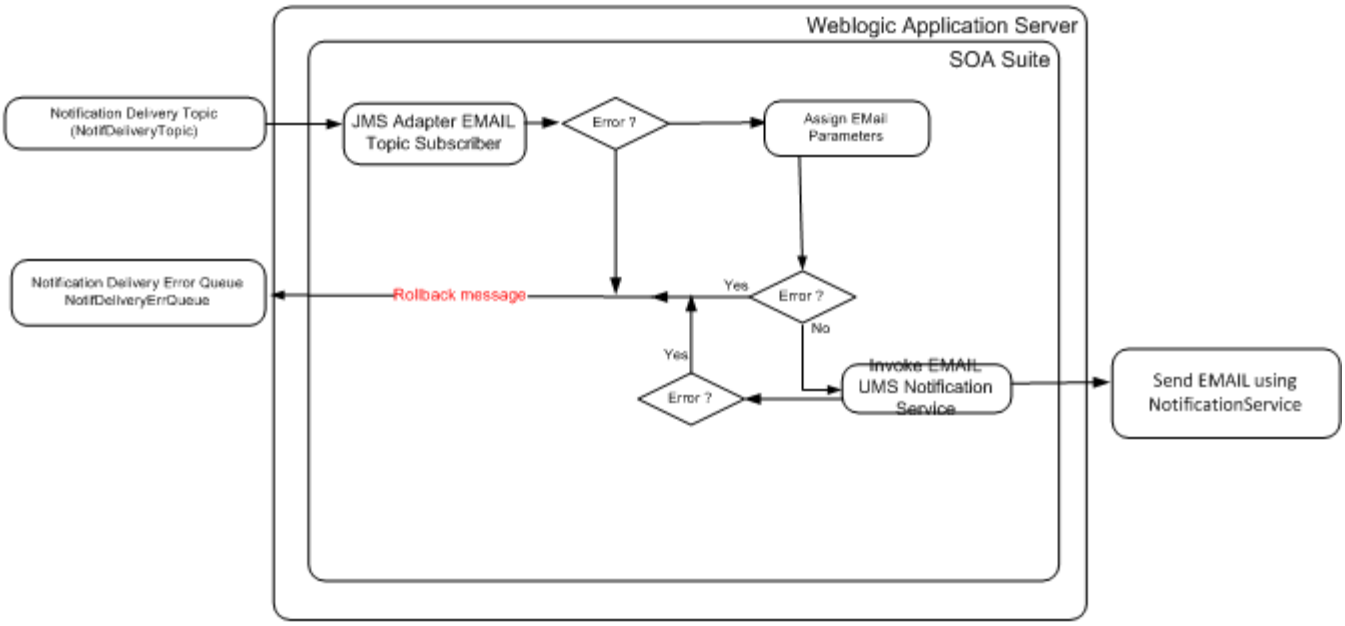
  When a technical error is encountered in the Integration, a SOAP fault will be returned with a specific technical message code. The message codes are obtained from the configuration properties file.

- **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 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.

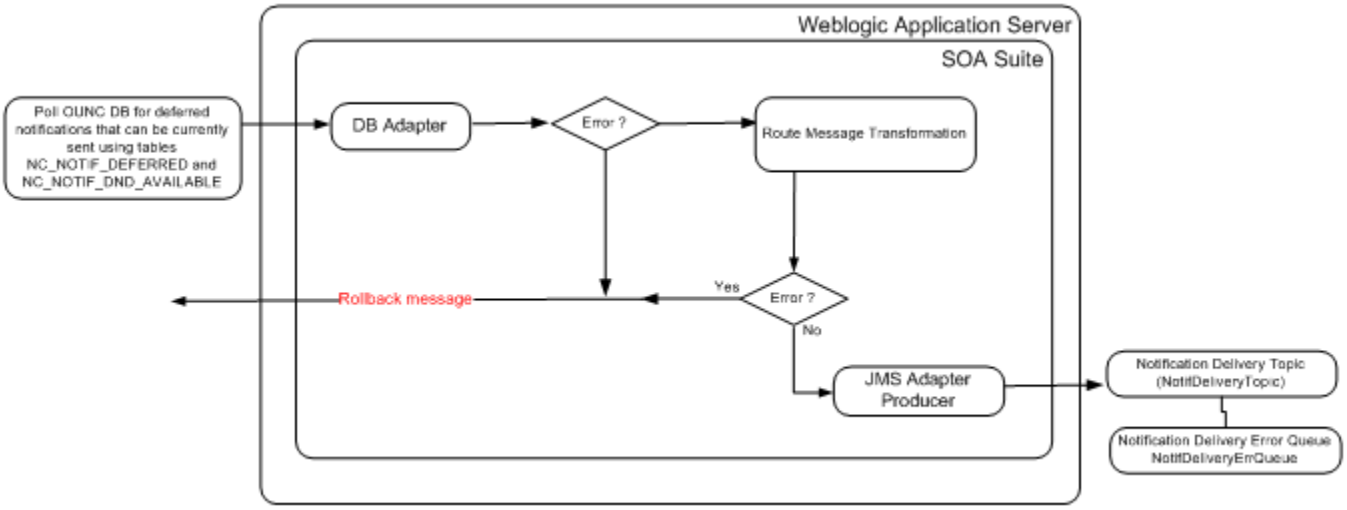
- **Customization**

  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

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

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 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 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 errorhandling 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.NotifType.Code.Outage	OUT	Code for NMS Notification type of Outage
NMS.NotifType.Descr.Outage	Outage	Description property for NMS Notification type of Outage
NMS.NotifType.Code.Restore	RST	Code for NMS Notification type of Restore
NMS.NotifType.Descr.Restore	Restore	Description property for NMS Notification type of Restore

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/@EdgeApplications.OUCCB.ManagedServer.context/XAIApp/xaiserver/ 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.

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.

NC_USER_NOTIF_PREF

This table stores user's preferences per notification type per account. Available notification types are stored in the Notification Center schema and should match those exposed by the edge applications.

NC_USER_NOTIF_PREF_DELIV_OPT

Many-to-many table between notification preferences and delivery channels defined by the user.

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 XAI 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: JDeveloper is required to customize/extend OUCSS taskflows.

Steps to Customize OUCSS Taskflows

Create Customization Documents

1. In CCB, configure custom fields (fields 1 through 10) that are intended to be extended for a given module/XAI Service.
2. Open JDeveloper and create a new application based on "WebCenter Portal - Spaces Task Flow Customization Application" template. Choose the defaults to create the application.
3. In the zip file provided in the installation files, find the OUCSS_Extension.war containing the taskflow of the module and its respective data control. For example, if you need to extend Account Summary module, then copy **oracle.ugbu.ss.billing.accountsummary.model_2.1.0.jar** and **oracle.ugbu.ss.billing.accountsummary.view_2.1.0.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.
4. In the Resource Palette, create a File System connection to the CM folder containing the ADF Libraries.
5. Choose the View Project of the newly-created application, then, from the resource palette file connection, right-click on all the ADF Libraries one by one and select/click on **Add to Project**.
6. Make sure that the corresponding data control (e.g., AccountSummaryService) is listed in the Data Control panel of the application.

7 In the JDeveloper Application Navigator, choose to show libraries.

8 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.

9 Expand ADF Library to find and open the the desired jsff (UI) file. For this sample, we will use **summary.jsff.**

10 In order to customize/extend the content, switch to JDeveloper’s Customization Mode.

11 Select **Tools > Preferences** in JDeveloper.

12 In the left pane, select **Roles**.

13 Choose **Customization Developer** from the list of roles on the right, and then click **OK**.

14 JDeveloper will restart (on Windows) or advise you to restart JDeveloper (on Linux). Restart JDeveloper.

15 After JDeveloper restart with Customization mode, a **Customization Context** window should show on bottom right corner in JDeveloper. Make sure that

16 "Edit with following Customization Context” radio is enabled/selected.

17 Tip Layer is selected with both Name and Value as “site”. By default this value is webcenter. OUCSS application is configured to listen to customizations with value “site”. To update the value to site

18 Click on “Configure application layer values” at the bottom to open CustomizationLayerValues.xml

19 Update the value=”webcenter” to value=”site” and Display-name=”WebCenter” to Display-name=”Site” (E.g. : <cust-layer-value value="site" display-name="Site" id-prefix="wc"/>)

20 Save the changes.

21 Confirm that Edit is enabled and Tip Layer/Customization Context is configured as site/site.

22 In customization mode, you can edit any content on the page. 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.jsf.xml`. To Customize.

23 Select any Custom Field (1 to 10) from the Data Control and drag it to the location in the jsff where it needs to be rendered.

24 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.

25 Ensure that you are extending using the corresponding Data Control of the module of the jsff.

26 All customization delta file(s) that needs to be uploaded to MDS is found under <<ApplicationFolder>>\ViewController\libraryCustomizations\ folder. Copy this folder to any location where the application managed server is running.

27 If any updates are erroneously applied, delete the *.jsff.xml and *.pageDef.xml file(s) from <<application_folder>>\ViewController\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. To import the customizations documents,

1 Run the WLST and connect to the server running the OUCSS Portal application as Admin user.

2 Run the WLST command importMetadata(application='<<Target_Application_Name>>', server='<<Target_Managed_Server>>', fromLocation='<<CopyLocationFromAbove>>', docs='/**').

Example:

```
importMetadata(application='OUCSSPortal',
server='WC_CustomPortal',fromLocation='/OUCSS/Customization', 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 XAI 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.

To be able to invoke the custom XAI web service from 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.

2 Execute the following configuration changes in the edge application (CCB or MDM).

- Change the XAI Adhoc Parameter called **serviceNameSources** in the XAIParameterInfo.xml file to add the ***url*** parameter, as shown in the following code block:

```
<AdHocParameters>
   <Options>
      <Option name="serviceNameSources" value="url,soapaction,message" />
   </Options>
</AdHocParameters>
```

The value defined in this option is a comma-separated list of sources that can be inspected in order to figure out what XAI inbound service to invoke. If the url parameter is provided, it will check the submitted url and extract the service name from the last component of the path. The default value of the **serviceNameSources** parameter does not include the url parameter.

Sample:

OUCSS is changed to call a custom web service located at http://ccbserver:host/spl/XAIApp/xaiserver/customXAIService

A call from OUCSS will redirect the service call to the new service.

In the URL http://ccbserver:host/spl/XAIApp/xaiserver/customXAIService, customXAIService would be the intended service.

Refer to your *Customer Care and Billing Implementation Guide* and *Meter Data Management Implementation Guide* for further information regarding ServiceScripts and XAI Inbound 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://www.oracle.com/technetwork/apps-tech/utilities/documentation/index.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 Application Deployments.
3. Click on OUCSSPortal (v2.1.0) to load the OUCSS Portal application summary page.
4. Monitor the "Response and Load" Graph to get an overall idea of how the application is performing.
5. To drill down more, click on Performance Summary from Application Deployment from the top 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 OUCSS Portal is performing, click on Application Deployment menu at the top and select "WebCenter Portal -> Overall Page Metrics". This will load a page to list proceessing times taken to load each Portal page.

Monitor the OUCSS Portlets Producer Application Using WebLogic EM (for Portlet based Solution)

1. Login to WebLogic enterprise manager as WLS Admin.
2. From the Domain Menu (e.g., Farm_<<domain_name>> on the left, expand Application Deployments

3. Click on OCUSSPortletsProducer (v2.1.0) to load Summary of OUCSS Producer application.
4. Monitor the Response and Load group on the right pane to check the health of the producer application.

   Optionally you can also select Performance Summary from "Application Deployment" menu on the top to get more details on Request, their processing time, active sessions and other metrics.

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 Domain Menu (e.g., Farm_<<domain_name>> on the left) expand Application Deployments.
3. Click on OUCSSPortal (v2.1.0) to load the OUCSS Portal application summary page.
4. From Application Deployment menu on top, Select Logs -> View Log Messages to load the Log Messages page.
5. Select the criteria from the form, for e.g., set Date Range to 5 hours and click on "Search".
6. 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.

OUCSS Producer Application Logs for a Portlet-based Solution

1. Login to WebLogic Enterprise Manager as WLS Admin.
2. From the Domain Menu (e.g., Farm_<<domain_name>> on the left, expand Application Deployments
3. Click on OUCSSPortletsProducer (v2.1.0) to load Summary of OUCSS Producer application.
4. From "Application Deployment" menu on top, Select Logs -> View Log Messages to load the Log Messages page.
5. Select the criteria from the form, for e.g., set Date Range to 5 hours and click on "Search".
6. 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 OUCSS Portal and OUCSS Producer 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:

7. Login to the server on which the managed servers are running. Make sure the user has permissions to the domain home.
8. Change directory to <<Domain_Home>>/servers.
9. To access OUCSS Portal logs, go to WC_CusotmPortal/logs folder and to access OUCSS Producer 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_ CusotmPortal.log and WC_ CusotmPortal -diagnostics.log) will be found under : /u01/oracle/product/webcenter/user_projects/domains/portal_domain/servers/WC_ CusotmPortal /logs.
   - OUCSS Producer logs if deployed under WC_Portlets manager server will be found under /u01/oracle/product/webcenter/user_projects/domains/portal_domain/servers/WC_Portlets/logs.

Monitoring Document References

- Monitor Oracle Fusion Middleware

http://download.oracle.com/docs/cd/E17904_01/core.1111/e10105/monitor.htm#CFAEHCGG

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 XAI. User will find all incoming requests and responses in xai.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 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 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 its not running or check the MapViewer URL configured in EM.	

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

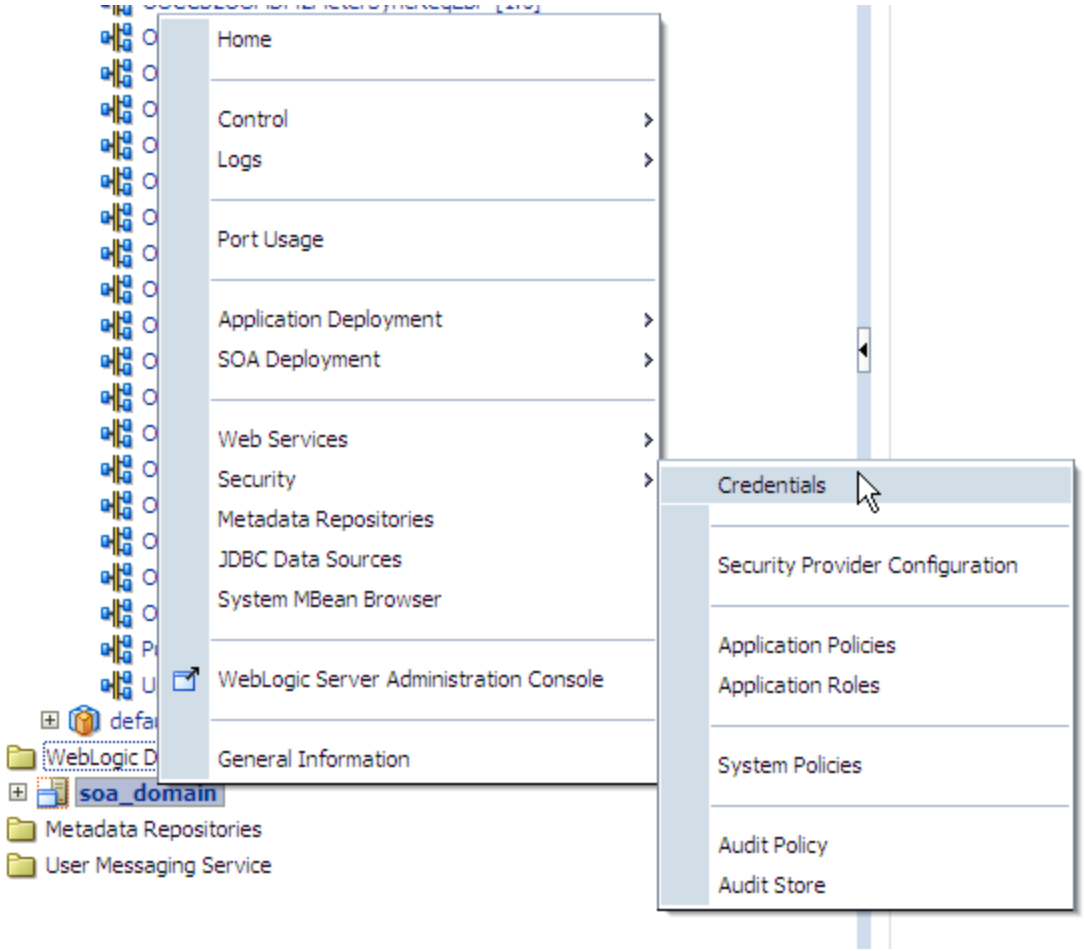
Check Process Instance

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.

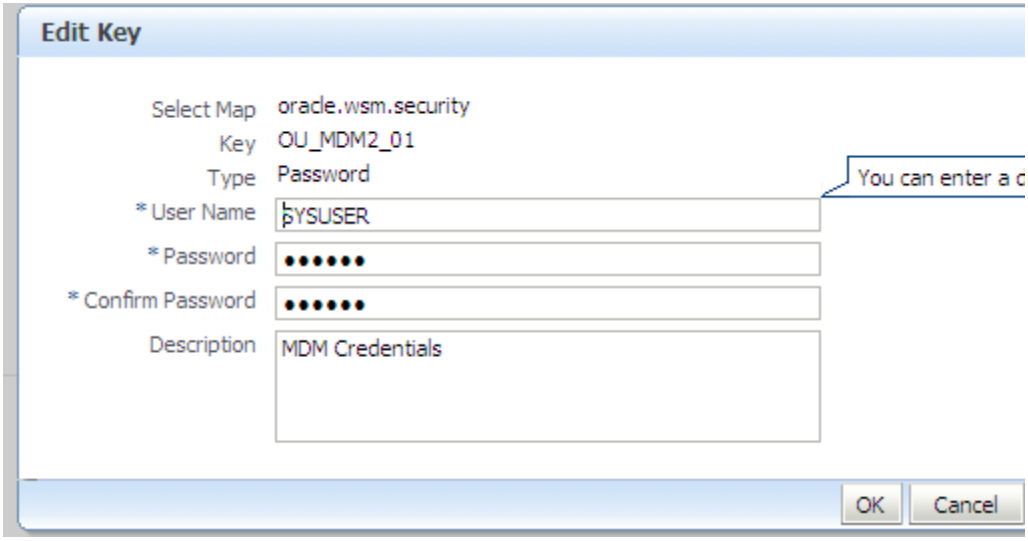
Check Credential Key (csf-key) Attached to Policy

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.

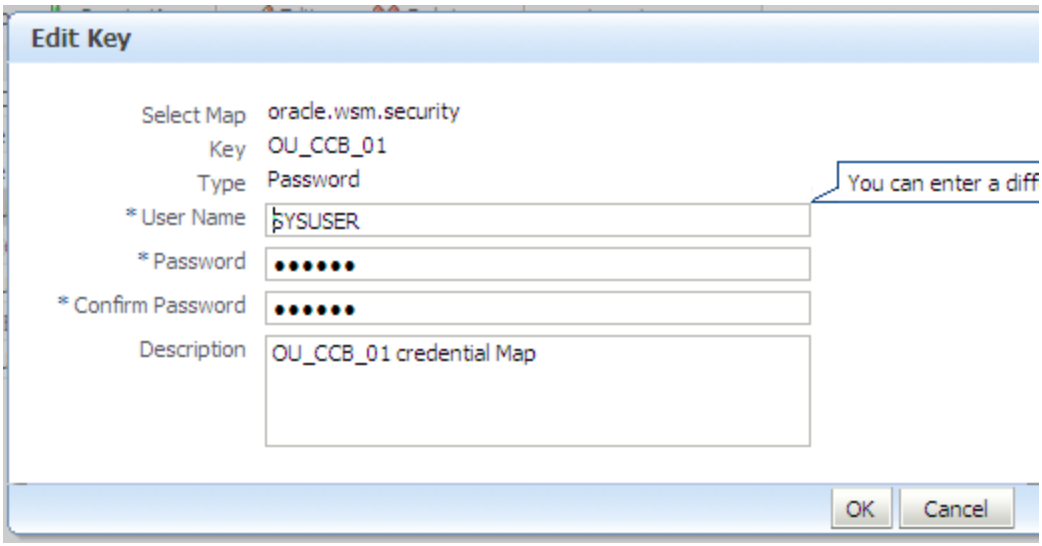


4 Expand oracle.wsm.security and OU_MDM2_01 and OU_CCB_01 keys should be defined there.

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 XAI Configuration to ensure they are configured correctly. Refer to the Setting Up Oracle Utilities Customer Care and Billing – XAI 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="xaixsd:WX-
GetUsageOverviewSoapBinding">
      <soap:address location="https://mdm server:9999/ouaf/XAIApp/xaiserver/WX-
GetUsageOverview"/>
   </wsdl:port>
</wsdl:service>
```

Note: The MDM wsdl is in $PRODUCT_HOME/MDS-Artifacts/CCB2-MDM2/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUMDM2/V1/wsdls

Chapter 11

OUCSS Mobility

OUCSS Mobile solution provides utility companies with an interface that enables customers to manage their accounts, monitor consumption and interact with the utility using mobile devices. This solution renders interfaces conforming to mobile browser standards to enable utilities customers to use handheld device to interact with the utility.

Functional Overview

Oracle Utilities Customer Self Service modules include the following functionality:

- Account Management Module:
  - Alerts and notifications
  - User Registration
  - Account Enrollment
  - Account Automatic Payment Settings
- Billing and Payment Management Module:
  - Account charges summary
  - Service charges to-date
  - One-time payments
  - View promotions
  - Financial History
  - Balance and Charges for Prepaid Accounts
  - Estimates and Costs for Prepaid Accounts
  - Viewing Bills online
- Customer Service Management Module:

  - Add scalar meter read data
  - View Usage Consumption (Scalar Meter)
  - View Usage Consumption (Smart Meter)
  - Forms Management
- Outage Module:
  - Outage Table - Display outage information for the utility as text. Out of box, outages are aggregated by the postal code, city name and county name .
  - 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
  - Outage Maps – Display outage information for the utility in a map.

Technical Overview

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

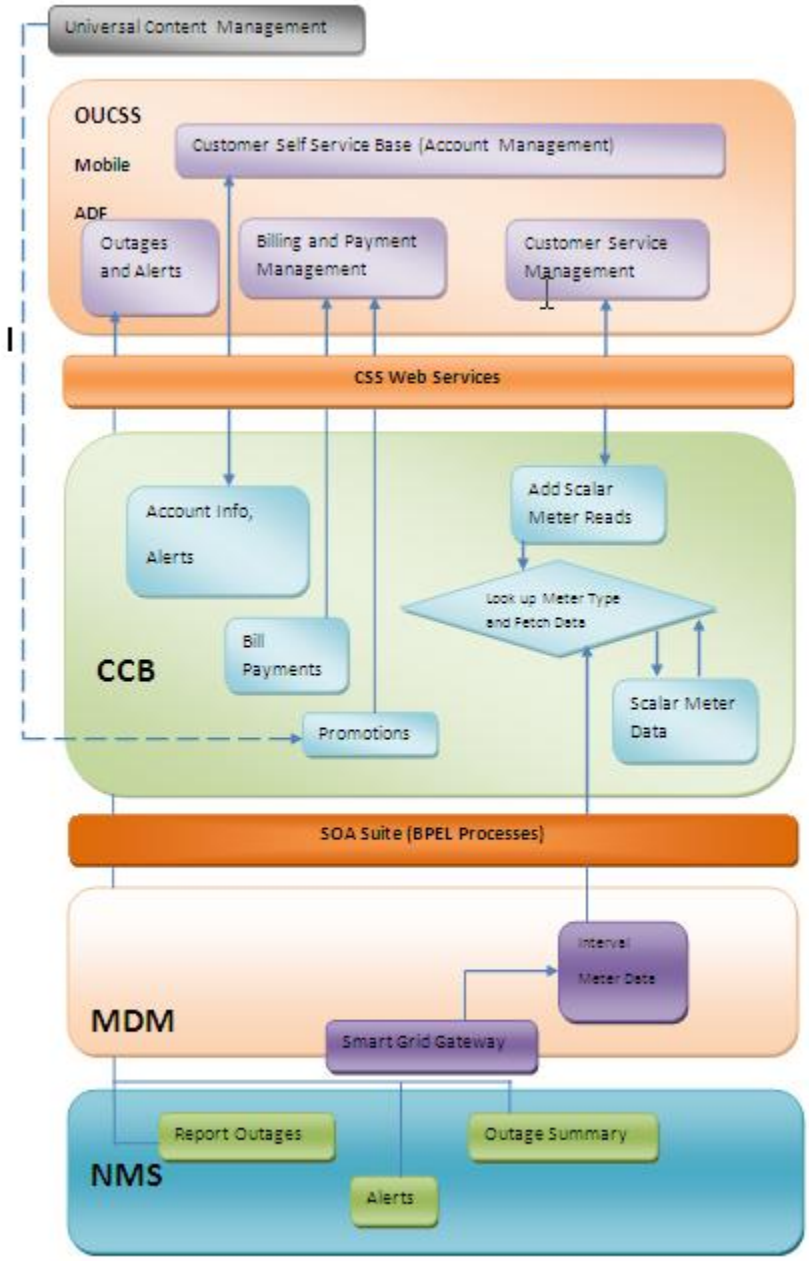
- Taskflow components are developed using Oracle Application Development Framework (ADF) 11g
- Taskflows are pre-integrated with OU Customer Care and Billing services using standards based web service API
- Reference OUCSS mobility solution (with preconfigured security, navigation model and extension) is provided with the release package to facilitate implementation and development activities.

Security

OUCSS offers Tier-1 and Tier-2 security.

- Tier-1 Security: Most of the pages in OUCSS Portal are secured and are accessed through specific roles only. For more information on this type of security see the Reference Security Roles section.
- 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 Mobile Architecture



OUCSS Mobile Topology

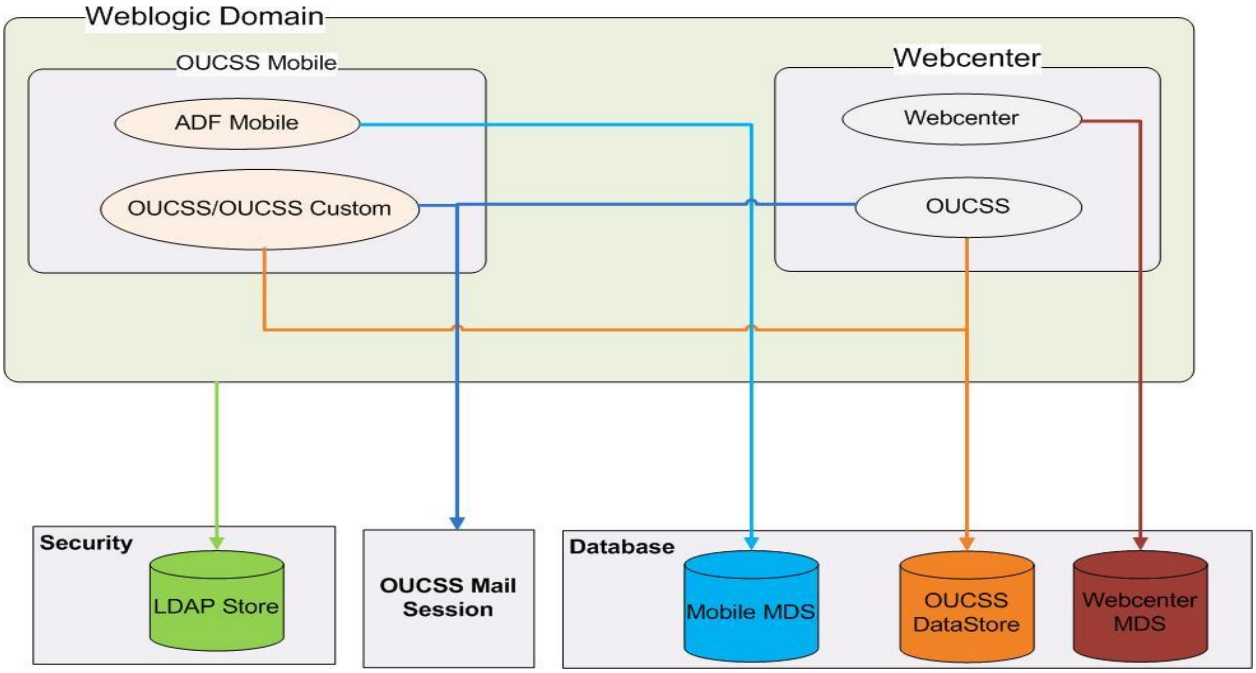
Mobility and OUCSS Portal in the Same WebLogic Domain

An OUCSS Portal and an OUCSS mobility solution can co-exist in the same WebLogic domain, though not on the same managed server (e.g., the same JVM). OUCSS mobility should, therefore, be deployed on a separate managed server.

OUCSS mobility uses the same OUCSS data store and user store (LDAP) as used by OUCSS Portal application. User registration and account enrollment can be done via OUCSS Portal application or the OUCSS mobile solution.

Once registered and enrolled througheither the OUCSS Portal application or the OUCSS Mobile solution , the user can login to OUCSS mobile (using the same credentials used for registration and enrollment). The OUCSS mobility solution is configured to use an independent MDS store and all ADF customization artifacts should be uploaded to the Mobile MDS store. This type of solution enables customers to use WebLogic embedded LDAP for both OUCSS Portal and OUCSS mobile solutions. This type of solution is supported out-of-the-box using installation scripts provided with this release.

OUCSS Mail Sessions and Datastore resources are targeted to the OUCSS Mobile managed server as the mentioned resources gets created while installation of the Webcenter OUCSS base.



Mobility and OUCSS Portal in Separate WebLogic Domains

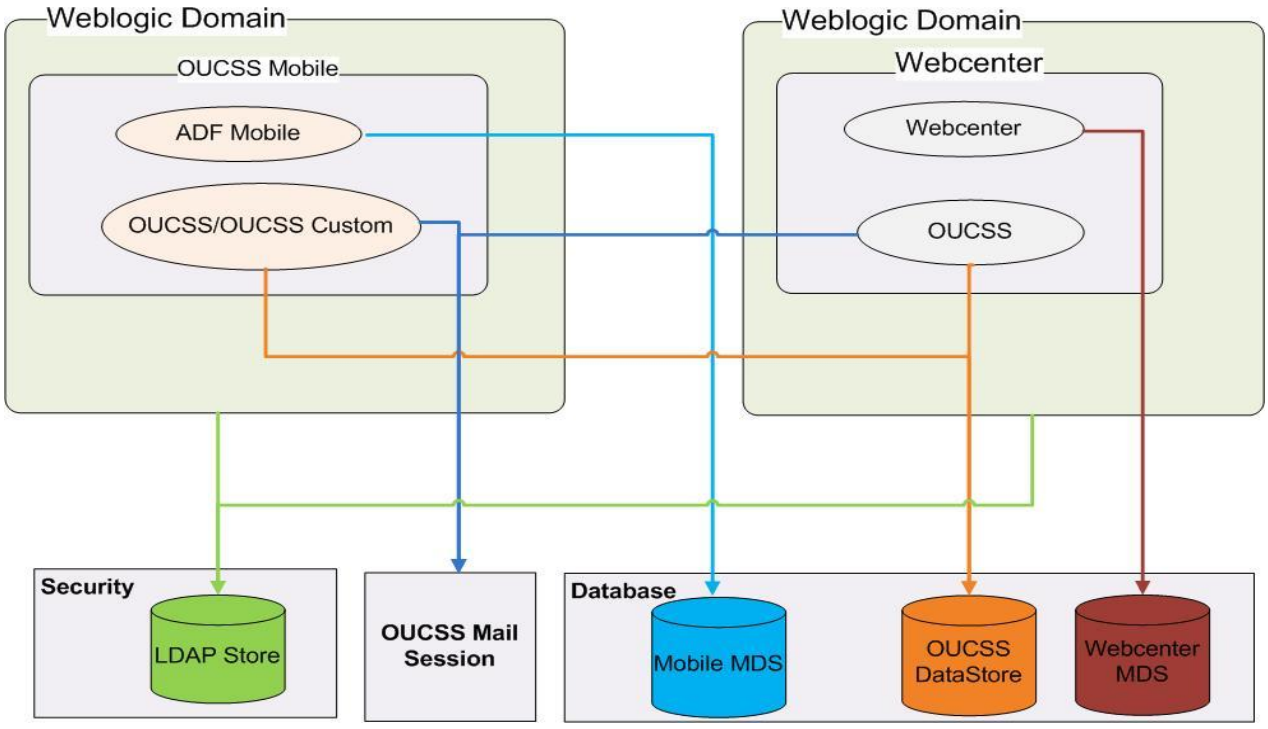
OUCSS Portal application and OUCSS mobility solution can be deployed in the different weblogic domain. OUCSS mobility should utilize the same OUCSS datastore & user store (LDAP) as used by OUCSS Portal.

User registration & account enrollment is not supported through OUCSS mobile solution, but user can register and enroll through OUCSS Portal. Once registered and enrolled user can login to OUCSS mobile (using same credential use for registration and enrollment).

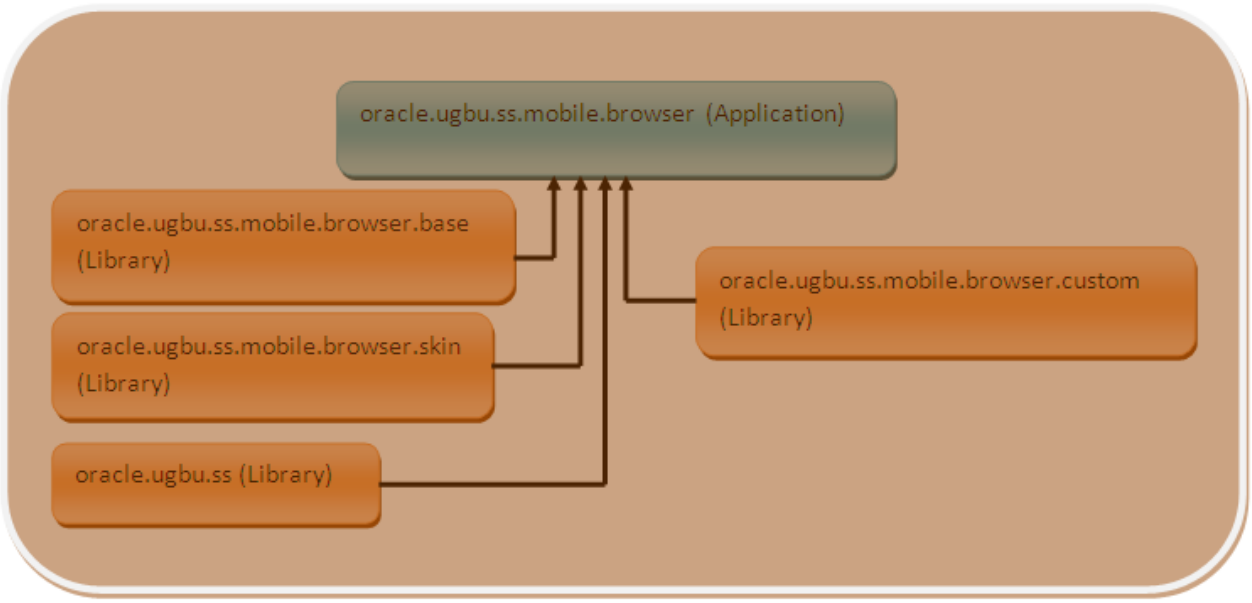
OUCSS mobility solution is configured to use independent MDS store & all ADF customization artifact should be upload to Mobile MDS store. WebLogic embedded LDAP can not be use in this type of configuration & both domains need to be mapped to same LDAP store (user credential storage) & thus an external LDAP should be configure.

OUCSS Mail Sessions and Datastore resources needs to be created on the domain where the OUCSS Mobile managed server is created.

This type of solution is not supported out-of-box installation scripts provided with this release but customer can setup there environment by manually deploying OUCSS mobile artifacts.



OUCSS Mobility Components



OUCSS Mobile Artifacts

Artifact Name	Description	Type
com.oracle.ugbu.ss.lib	com.oracle.ugbu.ss.lib is OUCSS core shared library that provides implementation of model layer for taskflows in Mobile Application. This library also contains implementation of OUCSS taskflows referenced in OUCSS Portal application. Any changes done at this library level, e.g., customization/personalization, will be impacting OUCSS Mobile and OUCSS Portal. This artifact point to physical artifact as <PRODUCT_HOME>/Install/application/ OUCSS_Extension.war	Library
oracle.ugbu.ss.mobile.browser.base_2.1.0	This artifact contain OUCSS Mobile implementation as ADF libraries. This is web archive which contain all OUCSS Mobile implementation as ADF libraries in WEB-INF/lib. This artifact point to physical artifact as <PRODUCT_HOME>/Install/application/oracle.ugbu.ss.mobile.browser.base _2.1.0.war	Library
oracle.ugbu.ss.mobile.browser.skin_2.1.0	This artifact contain skinning setup for OUCSS Mobile implementation as Jar. Skinning configuration is externalize as Java jar to enable customer to implement there own skinning support. This artifact point to physical artifact as <PRODUCT_HOME>/Install/application /oracle.ugbu.ss.mobile.browser.skin_2.1.0.jar	Library
oracle.ugbu.ss.mobile.browser.custom_2.1.0	This artifact is provided for customer to extend OUCSS Mobile solution by implementing/re-implementing functionality as per their liking. Custom can deploy their implementation as this library & OUCSS Mobile solution will be able to pick it up. This artifact point to physical artifact as <PRODUCT_HOME>/Install/application/oracle.ugbu.ss.mobile.browser.custo m_2.1.0.war	Library
oracle.ugbu.ss.mobile.browser.2.1.0	This artifact is OUCSS Mobile renderer which render all Mobility component	Application

	by including all above defined library. This is the only executable layer in OUCSS Mobile solution. This artifact point to physical artifact <PRODUCT_HOME>/Install/application/oracle.ugbu.ss.mobile.browser_2.1.0.ear

Navigation Model

OUCSS mobility solution is implemented using ADF taskflow containing view specifically developed for mobile browser. Out-of-box OUCSS mobility solution navigation model is package as ADF library jar (named oracle.ugbu.ss.mobile.browser.navigation_2.1.0.jar) and present inside WEB-INF/lib folder of <OUCSS product home>/ Install/application/oracle.ugbu.ss.mobile.browser.base_2.1.0.war web archive. Mobility URL's are externalized to OUCSS navigation store. Mobility solution contains two categories of page as listed below.

- Public: Welcome page, login page, registration, outage table, outage map, report public outage.

  Header section of public pages are externalize to a separate jspx page & included using jsp:include on all page. Header page url is derived from the property value of "MOB_B_PUBLIC_HEADER" property present in resource store of OUCSS application. Mobility pages require special metadata in html header tag for proper rendering on webkit enabled mobile browser. This metadata, viewport generally specify width, initial scale, maximum scale & user scalable properties which are must to render view properly on mobility browser. HTML header tag level metadata definition is externalized in separate jspx & included on all pages through "MOB_B_PUBLIC_HTML_HEADER" property present in OUCSS resource store. Example – For iPhonewebkit, below verbatim is use

  ```
  <f:verbatim rendered="#{requestContext.agent.skinFamilyType eq 'iPhonewebkit'}">
      <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-
  scale=1.0; user-scalable=0;"/>
  </f:verbatim>
  ```

  Verbatim is defined for linuxwebkit, blackberry & defaultwebkit skin family type

- Secured – home page for authenticated user, OUCSS taskflows etc

  Header section of public pages are externalize to a separate jspx page & included using jsp:include on all page. Header page url is derived from the property value of "MOB_B_AUTH_HEADER" property present in resource store of OUCSS application. Mobility pages require special metadata in html header tag for proper rendering on webkit enabled mobile browser. This metadata, viewport generally specify width, initial scale, maximum scale & user scalable properties which are must to render view properly on mobility browser. HTML header tag level metadata definition is externalized in separate jspx & included on all pages through "MOB_B_AUTH_HTML_HEADER" property present in OUCSS resource store. Example – For iPhonewebkit, below verbatim is use

  ```
  <f:verbatim rendered="#{requestContext.agent.skinFamilyType eq 'iPhonewebkit'}">
      <meta name="viewport" content="width=device-width; initial-scale=1.0; maximum-
  scale=1.0; user-scalable=0;"/>
  </f:verbatim>
  ```

  Verbatim is defined for linuxwebkit, blackberry & defaultwebkit skin family type

  All taskflow, page & images url's are externalize to OUCSS resource store. Resource store lifecycle is maintained using admin module of OUCSS Portal only. Resource store lifecycle maintenance is not supported through OUCSS mobile solution. Inorder to reflect the changes (done through OUCSS Portal admin module) in mobility restart the mobility application as follows:

1. Log in to the WebLogic console as Administrator.
2. Click **Deployments** and select the "oracle.ugbu.ss.mobile.browser_2.1.0" application.
3. Click stop > Force stop now.
4. Select the "oracle.ugbu.ss.mobile.browser_2.1.0" application, then click on start > Servicing all request.

The oracle.ugbu.ss.mobile.browser_2.1.0 application status will turn to **active** or **running**. This process invalidates/deletes all active user sessions on the environment, and the user must log in again to the OUCSS mobile solution using a fresh browser instance.

The following tables describe the property names, descriptions and default values in the OUCSS resource store.

Resource Code	Description	Value
MOB_B_ACCOUNT_LIST_URL	This property point to mobility view account list taskflow url	/faces/adf.task-flow?_id=viewaccountlisttaskflow&_document=/web-inf/com/oracle/ugbu/ss/mobile/browser/viewaccountlist/viewaccountlisttaskflow.xml
MOB_B_ACCOUNT_VIEW_URL	This property point to mobility view account overview taskflow url	/faces/adf.task-flow?_id=ViewAccountTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/commercial/viewaccount/ViewAccountTaskFlow.xml
MOB_B_ALERTS_URL	This property point to mobility alerts taskflow url	/faces/adf.task-flow?_id=AlertsTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/alerts/AlertsTaskflow.xml
MOB_B_AUTHENTICATE_HOME_PAGE	This property point to mobility home page for authenticated user	/faces/com/oracle/ugbu/mobile/browser/navigation/secured/home/home.jspx
MOB_B_AUTH_HEADER	This property point to header page included in all secured pages for authenticated user	/com/oracle/ugbu/mobile/browser/navigation/secured/header/Header.jspx
MOB_B_AUTH_HTML_HEADER	This property point to header page which define html header metadata for proper render of page on a mobile browser for secure pages. This property points to page which contain verbatim for each requesting agent type	/com/oracle/ugbu/mobile/browser/navigation/secured/header/htmlHeader.jspx
MOB_B_PUBLIC_HEADER	This property point to header page included in all public pages for mobility solution	/com/oracle/ugbu/mobile/browser/navigation/secured/header/Header.jspx
MOB_B_PUBLIC_HTML_HEADER	This property point to header page which define html header metadata for proper render of page on a mobile browser. This property points to page which contain verbatim for each requesting agent type	/com/oracle/ugbu/mobile/browser/navigation/public/header/htmlHeader.jspx
MOB_B_BILLING_ACC_SUMMARY_URL	This property points to mobility billed charges taskflow url	/faces/adf.task-flow?_id=AccountSummaryTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/accountsummary/AccountSummaryTaskflow.xml
MOB_B_BILLING_SCTD_URL	This property points to mobility service charge to date taskflow url	/faces/adf.task-flow?_id=ServiceChargeToDate-task-flow-definition&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/sctoDate/ServiceChargeToDate-task-flow-definition.xml
MOB_B_CUSTOMER_USAGE_OVERVIEW	This property points to mobility usage overview taskflow url	/faces/adf.task-flow?_id=UsageOverviewTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/cus

		tomer/usageoverview/UsageOverviewTaskflow.xml
MOB_B_CUST_CONSUMPTION_SUM_URL	This property points to mobility consumption summary overview taskflow url	/faces/adf.task-flow?_id=ConsumptionSummaryTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/consumptionsummary/ConsumptionSummaryTaskflow.xml
MOB_B_LOGIN_PAGE	This property point to mobility login page url	/faces/com/oracle/ugbu/ss/mobile/browser/login/login.jspx
MOB_B_LOGO	This property point to mobility branding company logo used in mobility header on all secure and public pages/views	/adf/images/oracle_mobi.gif
MOB_B_OUTAGE_URL	This property points to mobility outage taskflow url	/faces/adf.task-flow?_id=OutageTableTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/outage/geomap/OutageTableTaskflow.xml
MOB_B_PAYMENT_ONETIME_URL	This property points to mobility one time payment taskflow url	/faces/adf.task-flow?_id=OneTimePaymentTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/payments/onetime/OneTimePaymentTaskflow.xml
MOB_B_SCALAR_METER_URL	This property points to mobility scalar meter entry taskflow url	/faces/adf.task-flow?_id=ScalarMeterTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/scalarmeter/ScalarMeterTaskFlow.xml
MOB_B_WELCOME_PAGE	This property point to mobility welcome page url. Page pointed through this url will be render when user browser makes a request for the mobility solution	/faces/com/oracle/ugbu/ss/mobile/browser/login/login.jspx
MOB_B_PROMOTIONS_URL	This property points to mobility promotion taskflow url	/faces/adf.task-flow?_id=StandardPromotionsMBTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/offers/promotions/StandardPromotionsMBTaskflow.xml&offerSet=SAMPLE_MOBILE_STANDARD_PROMOTION
MOB_B_REP_PREM_OUTAGE_URL	This property points to mobility report premise outage taskflow url	/faces/adf.task-flow?_id=ReportPremiseOutageTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/outage/reportoutage/ReportPremiseOutageTaskflow.xml
MOB_B_REP_PUB_OUTAGE_URL	This property points to mobility report public outage taskflow url	/faces/adf.task-flow?_id=ReportPublicOutageTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/outage/reportoutage/ReportPublicOutageTaskflow.xml
MOB_B_ACCOUNT_ACCESS_URL	This property point to mobility account access taskflow url	/faces/adf.task-flow?_id=viewAccountListTaskflow&_docu

		ment=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/viewaccountList/viewAccountListTaskflow.xml
MOB_B_AUTOPAY_URL	This property point to the mobility autopay taskflow url	/faces/adf.task-flow?_id=SSAutoPayTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/autopay/SSAutoPayTaskFlow.xml
MOB_B_AUTOPAY_TO_HOME_URL	This property point to the autopay taskflow url. For use with Autopay Settings on the home page	/faces/adf.task-flow?_id=SSAutoPayTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/autopay/SSAutoPayTaskFlow.xml&backButtonTitle=MOB_B_HOME&backButtonURL=MOB_B_AUTHENTICATE_HOME_PAGE
MOB_B_BILLING_BILL_URL	This property points to the mobility view bill taskflow url	/faces/adf.task-flow?_id=viewBillTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/bill/viewBillTaskflow.xml
MOB_B_ENROLLMENT_URL	This property points to mobility enrollment taskflow url	/faces/adf.task-flow?_id=enrollmentTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/enrollment/enrollmentTaskflow.xml
MOB_B_FINANCIAL_HISTORY_URL	This property points to mobility financial history taskflow url	/faces/adf.task-flow?_id=SSFinancialHistoryTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/financialhistory/SSFinancialHistoryTaskflow.xml
MOB_B_FORMS_ISSUE_URL	This property points to mobility create forms taskflow url. For use with Issue Forms	/faces/adf.task-flow?_id=EnterFormTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/formsmgmt/taskflows/controller/EnterFormTaskFlow.xml&formType=ISSUE&formLabel=MOB_B_FORMS_ISSUE_LBL
MOB_B_FORMS_SEARCH_URL	This property points to mobility forms search taskflow url	/faces/adf.task-flow?_id=FormListTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/formsmgmt/taskflows/controller/FormListTaskFlow.xml
MOB_B_FORMS_UPDATE_URL	This property points to mobility forms update taskflow url	/faces/adf.task-flow?_id=UpdateFormTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/formsmgmt/taskflows/controller/UpdateFormTaskFlow.xml
MOB_B_CURRENT_OUTAGE_IMG	This property points to the icon used with current outages.	/com/oracle/ss/shared/view/images/current_

		outage.png
MOB_B_IMPORTANT_IMG	This property points to the icon used with info alerts / messages	/com/oracle/ss/shared/view/images/info.png
MOB_B_OUTAGE_MAP_URL	This property points to the mobile outage map task flow	/faces/adf.task-flow?_id=OutageMapTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/outage/geomap/OutageMapTaskflow.xml
MOB_B_PLANNED_OUTAGE_IMG	This property points to the icon used planned outages	/com/oracle/ss/shared/view/images/planned_outage.png
MOB_B_PREPAID_BALANCEANDCHARGES_URL	This property points to the mobility prepaid balance and charges taskflow url.	/faces/adf.task-flow?_id=BalanceAndChargesTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/prepaidbilling/BalanceAndChargesTaskflow.xml
MOB_B_PREPAID_ESTIMATESANDCOST_URL	This property points to the mobility prepaid estimates and costs taskflow url.	/faces/adf.task-flow?_id=EstimatesAndCostTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/billing/prepaidbilling/EstimatesAndCostTaskflow.xml
MOB_B_REGISTRATION_URL	This property points to the mobility registration taskflow	/faces/adf.task-flow?_id=registrationTaskflow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/registration/registrationTaskflow.xml
MOB_B_WARNING_IMG	This property points to the icon used for warning in alerts/outages	/com/oracle/ss/shared/view/images/warning.png
MOB_B_FORMS_ISSUE	This property points to the mobility taskflow to be called by open issues alert.	/faces/adf.task-flow?_id=FormListTaskFlow&_document=/WEB-INF/com/oracle/ugbu/ss/mobile/browser/customer/formsmgmt/taskflows/controller/FormListTaskFlow.xml

Skinning Support

Skinning level configuration/implementation is externalized to J2EE artifact to enable customer to implement/configure skinning as per their standards. Skinning level configuration is present in <PRODUCT_HOME>/Install/application /oracle.ugbu.ss.mobile.browser.skin_2.1.0.jar and deployed as WebLogic library (name oracle.ugbu.ss.mobile.browser.skin_2.1.0). This library contains following artifacts-

- trinidad-skins.xml – This file contain mapping between stylesheet and skin family.
- Sample stylesheets – Some sample stylesheets are provided out-of-box with OUCSS mobile solution. These stylesheet are mapped to skin family type & uses sample images shipped with this solution.
- Sample images – Some sample images, like button, backbutton etc are shipped with this solution and this is used by sample stylesheet

Customer can implement their specific style sheets and setup the mapping in Trinidad-skin.xml file. Deploy a higher version of this library and restart OUCSS mobile application. This will enable OUCSS mobile application to use latest implementation version of this library.

Library configuration information:

- Extension-name - oracle.ugbu.ss.mobile.browser.skin_2.1.0
- Specification-Version: 11.1.1
- Implementation-Version: 2.1.0

The following table describes some style classes used in the OUCSS mobile application.

Style Class Name	Description
af_m_button_header	This style class is use in button present in the header section on secure pages
af_m_button	This style class is use in button present in non-header section on both secure and un-secure pages

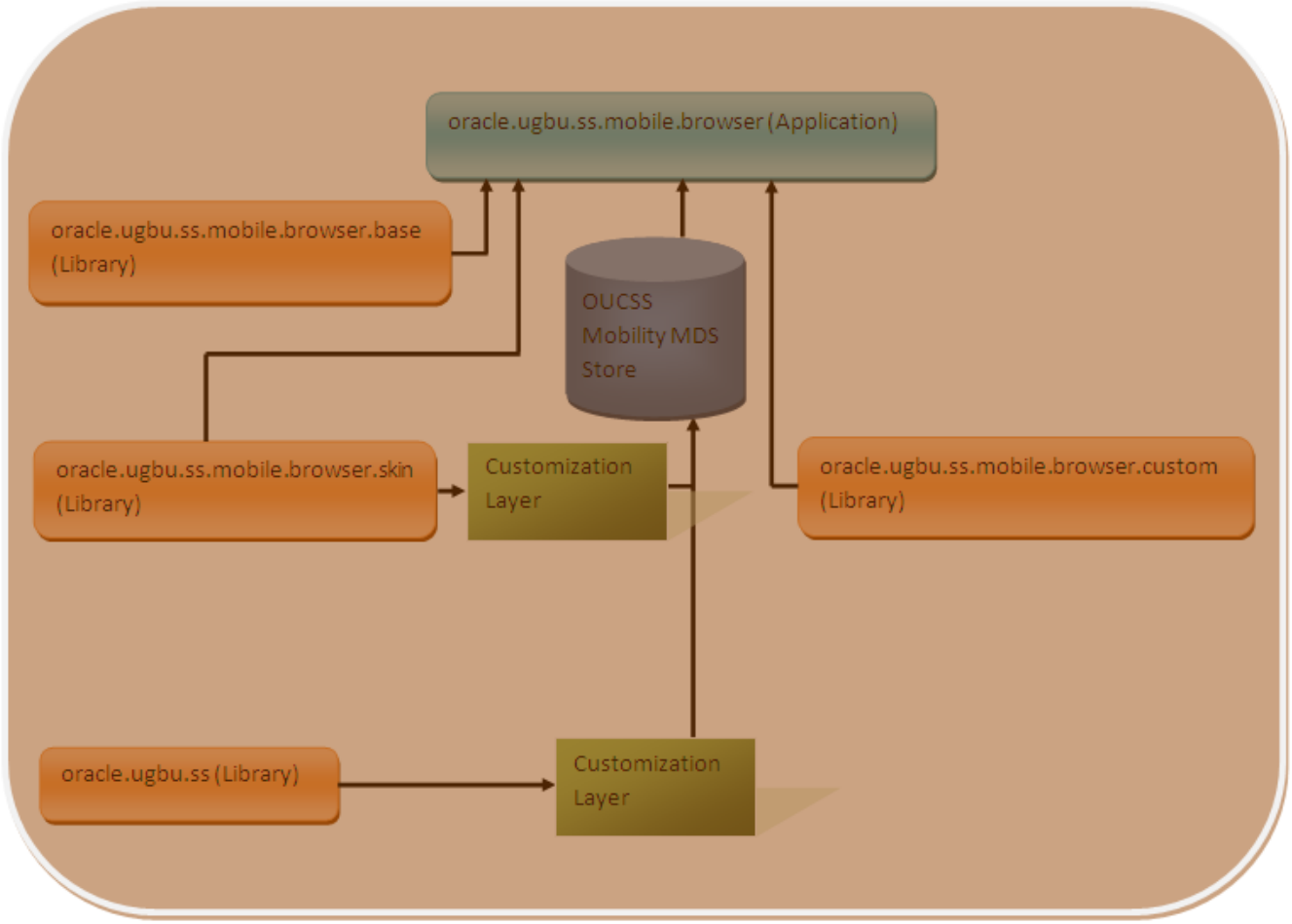
Branding Support

Branding uses the following values to enable customers to define their brand logo and background color on rendered pages.

Property	Location	Default Value
MOB_B_LOGO	Resource	/adf/images/oracle_mobi.gif
mobile.brow.branding.image.bg	Properties	#FFFFFF

The properties listed above are set using OUCSS Portal by logging in to the Admin module with a user who has the **WSSAdmin** role. After changes or updates are made, an administrator can log in to the console and restart OUCSS Mobile solution. This will reload all changes.

Customization and Extension Support in Mobility



OUCSS Mobility taskflows can be extended using ADF customization methodology. Customizations are stored in MDS. The OUCSS Mobility application is pre-configured to allow for customization.

Note: JDeveloper is required to customize/extend taskflows.

Steps to Customize a Taskflow

To create customized documents:

1. In CCB, configure custom fields (fields 1 through 10) that are intended to be extended for a given module.
2. Open JDeveloper and create a new Generic Application, and ensure the following technology scopes are selected for view project:
   - ADF Faces
   - ADF Mobile Browser
   - ADF Page Flow
   - Java
   - JSF

- JSP and Servlets

3 In the zip file provided in the installation files, find the ADF library containing the taskflow of the module and its respective data control. For example, if you need to extend Account Summary module, then copy **oracle.ugbu.ss.billing.accountsummary.model_2.1.0, oracle.ugbu.ss.billing.accountsummary.view_2.1.0** and **oracle.ugbu.ss.mobile.browser.billing.accountsummary _2.1.0** to a CM folder (e.g., `C:\OUCSS\Mobile\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.

4 Under the project > Project Settings > ADF Views, and set Enable Seeded Customizations.

5 Under Application Resources Descriptors >ADF META-INF:

A Open the adf-config.xml file and select the Overview tab.

B Select the MDS Configuration.

C Add the default Customization Class (`oracle.adf.share.config.SiteCC`), or extended Class of the Customization class.

6 Restart into Customization Role.

7 In the Resource Palette, create a File System connection to the CM folder containing the ADF Libraries.

8 Choose the project of the newly-created application, then, from the resource palette file connection, right-click on all the ADF Libraries one by one and select/click on **Add to Project**.

9 Make sure that the corresponding data control (e.g., AccountSummaryService) is listed in the Data Control panel of the application.

10 In the JDeveloper Application Navigator, choose to show libraries.

11 Browse and open the **summary.jspx** file from within **com.oracle.ugbu.ss.mobile.browser.billing.accountsummary** in the ADF Library listed in the Application Navigator. This is the view page used by the Account Summary mobile portlet.

12 In order to customize/extend the content, switch to JDeveloper's Customization Mode.

13 Select **Tools > Preferences** in JDeveloper.

14 In the left pane, select **Roles**.

15 Choose **Customization Developer** from the list of roles on the right, then click **OK**.

16 JDeveloper will restart (on Windows) or advise you to restart JDeveloper (on Linux). Restart JDeveloper.

17 Once restarted, the **Customization Context** (bottom right panel in JDeveloper) should show the following:

- Edit with following Customization Context is enabled.
- **Tip Layer** is selected with both **Name** and **Value** as "**site**". This is important, since the OUCSS application is configured to listen to customizations with value "**site**".

18 In customization mode, you can edit any content on the page. 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 this case, the file generated will be `summary.jspx.xml`. Make the necessary changes to extend the default display. Choose Trinidad components when customizing the page.

19 Select any Custom Field (1 to 10) from the Data Control and drag it to the location in the jsff where it needs to be rendered.

Optionally, customers can select other fields (which are not custom) that are available in data control but not part of the out-of-box UI.

20 Ensure that you are extending using the corresponding Data Control of the module of the jspx.

21 The directory `<<ApplicationFolder>>/<<Project Name>>/libraryCustomizations` contains all the customization files created from above. Copy this folder to any location where the application managed server is running.

22 If any updates are erroneously applied, delete the `jspx.xml` and `pageDef.xml` files from `<<ApplicationFolder>>/<<Project Name>>/libraryCustomizations` folder and start over.

Applying the Customization

1 Run the WLST and connect to the server running the OUCSS Mobile application.

2 Run the WLST command importMetadata(application='<<OUCSS_MobileApplication_Name>>', server='<<MobilityManagedServerName>>',fromLocation='<<CopyLocation from Step 21 above>>', docs='/**').

3 Login to the application and verify the changes. (A restart is generally not necessary.)

Steps to Customize an Edge Application Web Service

See Customization and Extension Methodology for details.

Steps to Extend the OUCSS Mobile Solution

This procedure describes how to extend the OUCSS Mobile solution by implementing/re-implementing new or existing taskflows and pages.

1 Download the OUCSS Mobile custom application zip file (<PRODUCT_HOME>/Install/application/oracle.ugbu.ss.mobile.browser.custom.zip).

2 Unzip and open the application in JDeveloper.

3 Modify the application as required.

4 Package this application as a web archive (*.war) and deploy as a WebLogic library with the following the metadata:

- Extension-Name: oracle.ugbu.ss.mobile.browser.custom_2.1.0
- Specification-Version: 11.1.1
- Implementation-Version: < value higher than 2.1.0.1>

5 Deploy the archive on the WebLogic domain, targeting it to the WebLogic managed server hosting the OUCSS Mobile solution.

6 Restart the oracle.ugbu.ss.mobile.browser_2.1.0 application using the WebLogic console, or restart the OUCSS Mobile managed server.