

**Oracle® Utilities Customer Care and Billing  
Integration to Oracle Utilities Network  
Management System**

Implementation Guide

Oracle Utilities Network Management System

v1.10.0.0.1

Oracle Utilities Customer Care and Billing v2.3.1

E18346-01

August 2010



Copyright ©2010, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are “commercial computer software” or “commercial technical data” pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any lost, costs, or damages incurred due to your access to or use of third party content, products or services.

# Table of Contents

---

Part 1: Understanding the Delivered Integrations .....	5
Chapter 1: Participating Applications Overview .....	5
Oracle Utilities Customer Care and Billing .....	5
Oracle Utilities Network Management System .....	5
Terms .....	6
Prerequisites .....	7
Chapter 2: Understanding the Integration .....	7
Business Process Overview .....	7
Understanding the Integration Processes .....	9
Part 2: Implementing the Delivered Integrations .....	29
Chapter 3: Configuration Guidelines .....	29
Setting up Oracle Utilities Customer Care and Billing .....	29
Setting Up Oracle Utilities Network Management System .....	51
Setting up the Integration Pack .....	52
Chapter 4: Error Handling, Monitoring and Troubleshooting .....	67
Error Handling by Integration Point .....	67
Monitoring from Oracle Utilities Customer Care and Billing .....	74
Monitoring from Oracle Utilities Network Management System .....	75
Monitoring from the Integration .....	75
Chapter 5: Extensibility Options .....	77
Pre Transformation Extension Point .....	77
Post Transformation Extension Point .....	77
Custom Transformations .....	77
Appendix A: Data Mapping .....	81
Customer Data Synchronization .....	81
Trouble Call Interface .....	88
Job History Query .....	93
Trouble Call History Query .....	95
Planned Outages Query .....	96
Appendix B: Cross References .....	98



# Part 1: Understanding the Delivered Integrations

This section provides an overview of the participating applications and information regarding the business processes addressed by this integration.

## Chapter 1: Participating Applications Overview

Customer Care and Billing Integration to Oracle Utilities Network Management System integrates the following products:

- Oracle Utilities Customer Care and Billing (CC&B)
- Oracle Utilities Network Management System (NMS)

This chapter provides a general description of each of these applications.

---

### Oracle Utilities Customer Care and Billing

Oracle Utilities Customer Care and Billing (CC&B) is a central repository for customer information - such as, name, address, phone number, etc. – which manages all aspects of the utility customer lifecycle including service connections, trouble calls and outages.

---

### Oracle Utilities Network Management System

Oracle Utilities Network Management System processes trouble calls from customers and analyzes them to determine probable outage locations. It can generate estimated restoration times (ERTs) that can then be provided back to customers. Also, it 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 customers. When 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.

## Terms

The following terms and acronyms are used throughout this guide.

### Application Names

CC&B	Oracle Utilities Customer Care and Billing
NMS	Oracle Utilities Network Management System

### General Terms

General terms to be aware of include:

DVM	Domain Value Map
BPEL	Business Process Execution Language
MDS	Metadata Store
EBF	Enterprise Business Flow
JMS	Java Message Service
JMS Queue	A staging area that contains messages those have been sent and are waiting to be read. The JMS Queues are available on the Weblogic Application Server
SOA	Service-Oriented Architecture – Software modules that are provided as services can be integrated or used by several applications using SOA, even if their respective architectures are substantially different. Rather than defining an API, SOA defines the interface in terms of protocols and functionality.
Edge applications	The applications that are involved in the integration - CC&B and NMS.
SOAP	Simple Object Access Protocol is a protocol specification for exchanging structured information in the implementation of Web Services in computer networks.
SA	CC&B Service Agreement
SP	CC&B Service Point
XAI	XML Application Integration. A CCB utility used to configure the system transfer information between CCB and external applications using XML. XAI exposes system business objects as a set of XML based web services. The service can be invoked via different methods (such as Hypertext Transfer Protocol (HTTP) or Java Message Service (JMS)). Consequently, any application or tool that can send and receive XML documents can now access the rich set of system business objects.
XSD	A schema definition file.
Fuzzy Calls	Trouble Calls that are not initially associated with a customer or device
UI	User Interface

## References

For more information on some of the terms and entities related to the integration, see Appendix B: Cross References.

## Prerequisites

Prior to starting this implementation you should have installed, setup and verified the following applications:

- Installed and configured Oracle Utilities Customer Care and Billing 2.3.1 and all of its prerequisite components as defined in the Oracle Utilities Customer Care and Billing Installation Guide.
- Installed and configured Oracle Utilities Network Management System 1.10.0.0.1 and all of its prerequisite components as defined in the Oracle Utilities Network Management System Installation Guide.
- Installed and configured Weblogic Application Server 11g and all of its prerequisite components as defined in the Installation Guide.
- Installed and configured SOA Suite 11g and all of its prerequisite components as defined in the Installation Guide.
- Completed, and verified, all of the steps identified in the installation guide for this product.

Please refer to the installation guide for this product for complete installation details.

## Chapter 2: Understanding the Integration

This chapter provides an overview of the integration and outlines the applicable business processes and the integration points in the following sections:

- Business Process Overview
- Understanding the Integration Processes

---

### Business Process Overview

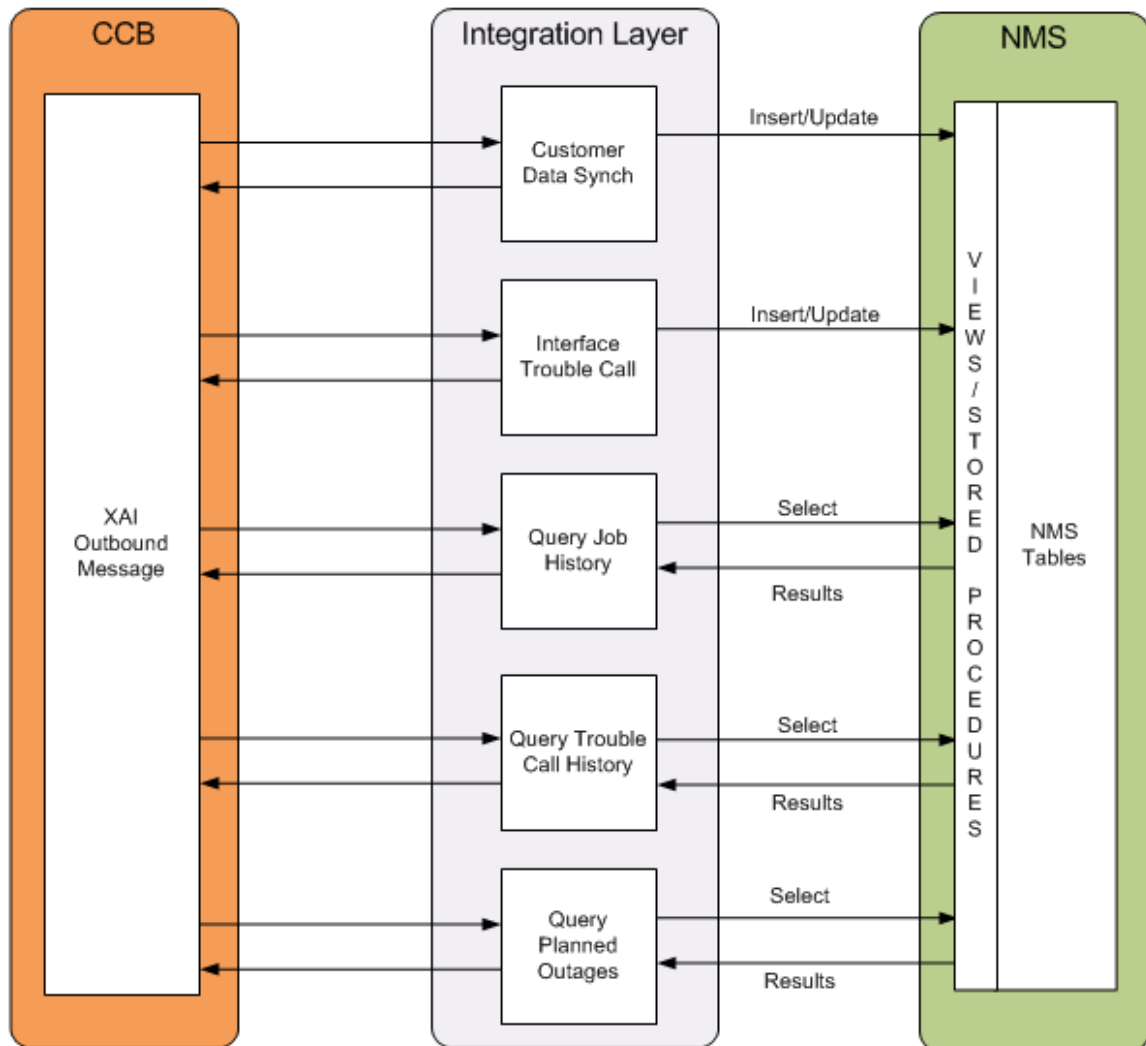
This integration supports synchronization of customer data and trouble calls from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System. It also supports the ability to query job history, trouble call history and planned outages from Oracle Utilities Customer Care and Billing.

The following list summarizes the functionality included in the integration:

- **Synchronize data:** Customer data is synchronized between Oracle Utilities Customer Care and Billing and Oracle Utilities Network Management System. To view customer information in the Oracle Utilities Network Management System application, only current information is required to associate customers with service location and supply nodes in the network data model.

- **Send trouble calls to Oracle Utilities Network Management System:** Capture trouble calls created or updated in Oracle Utilities Customer Care and Billing and send to Oracle Utilities Network Management System. The integration handles both trouble calls created for a particular customer with a known service point as well as “fuzzy” calls which are not initially associated with a customer or device.
- **Query trouble calls:** Query trouble calls that were placed by a particular customer or caller in Oracle Utilities Network Management System and display the results in Oracle Utilities Customer Care and Billing.
- **Query job history:** Query the current or recent Oracle Utilities Network Management System jobs that impact a particular customer and display the results in Oracle Utilities Customer Care and Billing.
- **Query planned outages:** Query planned outage jobs in Oracle Utilities Network Management System impacting a particular customer and display the results in Oracle Utilities Customer Care and Billing.

The following image provides a graphical representation of this functionality:



High Level Diagram



---

## Understanding the Integration Processes

This section provides detailed business process overviews and technical overviews of each of the business processes facilitated by the integration. These include the following:

- Customer Data Synchronization
- Trouble Call Interface
- Job History Query
- Trouble Call History Query
- Planned Outages Query

### Email Notifications

No email notification or error stores are delivered as default for any of the integration points, however the implementation team can choose to enable these functions for customer data sync and trouble calls interface by modifying the configuration properties.

### DVM Lookups

For customer data sync and trouble calls interface integration points, each DVM lookup which is configured in the integration layer has a corresponding DVM exception switch defined in the configuration file. The DVM exception switch determines whether or not the BPEL Process triggers a business error when the following DVM scenarios are encountered:

- When the value coming from Oracle Utilities Customer Care and Billing is not found in the DVM lookup
- When the value coming from Oracle Utilities Customer Care and Billing is found, but it does not have an equivalent Oracle Utilities Network Management System conversion value (null).

If the DVM exception switch is set to true, BPEL Process triggers a business error when a conversion error is encountered. Otherwise, the original value from Oracle Utilities Customer Care and Billing is mapped to Oracle Utilities Network Management System.

For all the query integration points, there are no corresponding DVM exception switches for the DVM lookup configured in the integration. When the DVM scenarios described above is encountered, the original value from Oracle Utilities Customer Care and Billing is mapped to Oracle Utilities Network Management System.

---

## Customer Data Synchronization

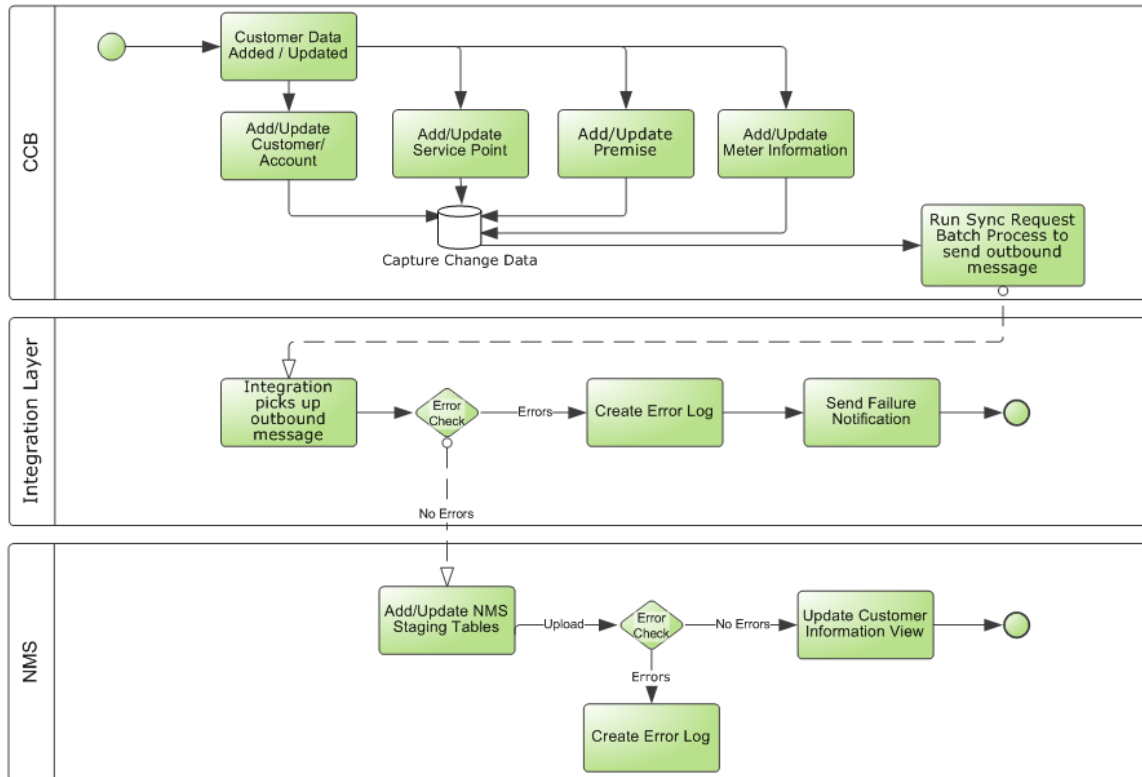
Customer data is synchronized in one direction from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System to support the following functionality.

### Supported Functionality

This integration point supports the following functionality:

- Send data from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System
  - Initial Sync (or Full Initial Load) - is the first load of data to create a Customer Data model in Oracle Utilities Network Management System.

- Incremental Sync - The changes since the last synchronization to the customer information in Oracle Utilities Customer Care and Billing are sent to Oracle Utilities Network Management System overwriting the last synchronized information.
- Only relevant and current electric customer data which is stored and maintained in Oracle Utilities Customer Care and Billing and that is needed to create the Oracle Utilities Network Management System customer model is synchronized. This involves getting data from the Person, Account, Premise, Service Point, Meter and Item tables in Oracle Utilities Customer Care and Billing.



Customer Data Sync Business Flow Diagram

### Creation and Update of Customer Data

Customer information is created and updated in Oracle Utilities Customer Care and Billing and this application is always the owner of customer data. Customer information must be kept up to date in Oracle Utilities Network Management System so that outage information can be properly synchronized with the appropriate customers and service points.

### Initial Synchronization / Incremental Updates

At the start of the implementation, the current customer data is synchronized from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System by batch processing which is run on initial load from Oracle Utilities Customer Care and Billing. Oracle Utilities Customer Care and Billing then keeps the data in sync with Oracle Utilities Network Management System using periodic incremental updates.

Oracle Utilities Customer Care and Billing sends one message for every customer that needs to be synchronized in Oracle Utilities Network Management System. This message contains all the customer related data load or updates for the customer relevant to Oracle Utilities Network Management System (i.e. person, account, premise, sp, meter/item information).

Only current customer information from Oracle Utilities Customer Care and Billing is synchronized when the following criteria is satisfied:

- The customer has an **active** or **pending stopped** service agreement (SA)
- The SA has an effective SA/SP link
- The SP linked to the SA is connected and in service

When Oracle Utilities Customer Care and Billing sends an update to Oracle Utilities Network Management System, the message may contain customer data with meter information or customer data with item information.

#### *Updates containing Meter Information*

Messages containing customer data with meter information are processed and sent over to NMS.

#### *Updates containing Item Information or Null Information*

Messages containing customer data with item information or with no meter or item information trigger an error since item information updates and null updates are not supported. Clients can configure the integration to support these types of updates by setting the appropriate customization flag to "true." For item information, the information must be manually mapped from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management.

#### *Inactive Customer Data*

When a customer becomes inactive in Oracle Utilities Customer Care and Billing, this information is sent so that Oracle Utilities Network Management System can mark the customer as inactive. This update only indicates that the customer is inactive, but does not provide details regarding whether the customer is inactive due to disconnection of service, for non-payment or if the customer has moved out.

## **Assumptions and Constraints for Customer Data Synchronization**

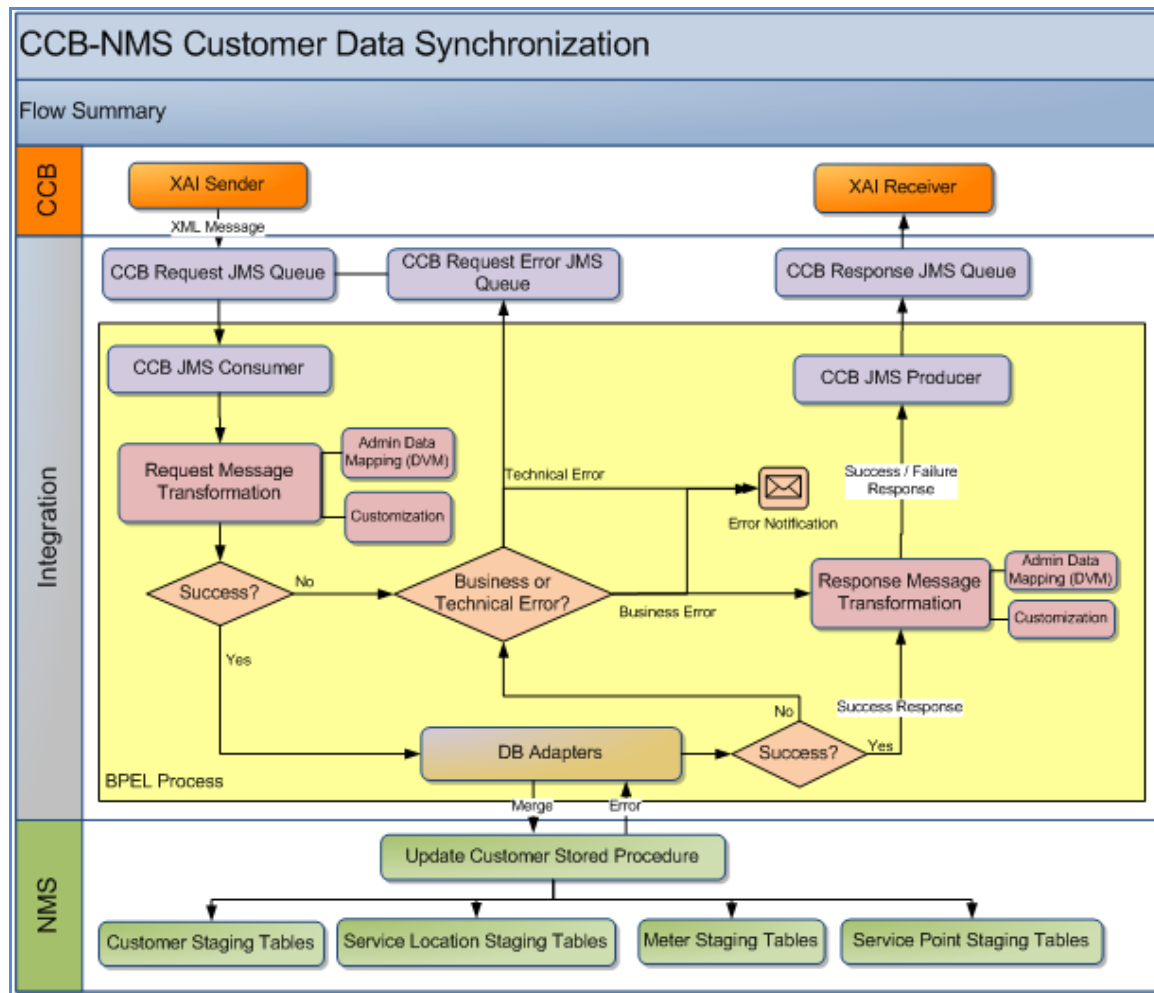
- Device information for each customer/sp must be setup in Oracle Utilities Customer Care and Billing first before the customer synchronization batch is run. Device ID used in Oracle Utilities Customer Care and Billing must be in the Oracle Utilities Network Management System Supply Nodes table. Oracle Utilities Customer Care and Billing stores the device information in the SP Geo Type and Oracle Utilities Network Management System stored it in Device ID.
- The Oracle Utilities Customer Care and Billing Customer Data sync is driven by SP and the data to be synchronized to Oracle Utilities Network Management System can be filtered by SP Type(s). The Oracle Utilities Network Management System Customer Data Sync BO Pre-processing algorithm filters the data to be synchronized by SP Type. Implementation teams can configure the SP Type(s) that need to be synchronized in the algorithm parameters and only those specified are synched over to Oracle Utilities Network Management System. If nothing is specified in the algorithm parameters, all SP Types are synched over.

Refer to the Oracle Utilities Framework *User Guide* section titled "Data Synchronization" for more information.

## Process Flow and Technical Details

This integration point supports a-synchronous data synchronization from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System with the following processing:

- Oracle Utilities Customer Care and Billing sends the synchronization message to an Oracle Utilities Customer Care and Billing Request JMS Queue for the integration layer to consume and process.
- The integration layer receives messages from the JMS Queue, transforms the message to the equivalent Oracle Utilities Network Management System field format, invokes the Customer Update Stored Procedure to insert/update the customer information in the Oracle Utilities Network Management System and sends the response or error messages to JMS Queues.
  - Weblogic JMS queues are used as a queuing mechanism in the integration layer between Oracle Utilities Customer Care and Billing and BPEL processes. 4 JMS queues support this integration.
    - Oracle Utilities Customer Care and Billing Request Queue - For Oracle Utilities Customer Care and Billing to add messages to this queue which will be picked up by the integration for processing.
    - Oracle Utilities Customer Care and Billing Response Queue - Business errors in the integration and success or failure of DB insert/update operations will be written to this queue.
    - Oracle Utilities Customer Care and Billing Request Error Queue - Technical errors encountered in the integration request flow will be written to this queue.
    - Oracle Utilities Customer Care and Billing Response Error Queue - Technical errors encountered when Oracle Utilities Customer Care and Billing reads the messages from the Oracle Utilities Customer Care and Billing response queue will be written to this queue.
- A BPEL Process with the following components processes the message. This BPEL process is not exposed as a web service.
  - JMS Consumer to read messages from the Oracle Utilities Customer Care and Billing request queue
  - JMS Producer to write to the Oracle Utilities Customer Care and Billing response queue
  - Transformation to convert message from source format to target format. DVM's will be used for the transformation.
  - Error handling and error notification
  - DB Adapter to interact with the Oracle Utilities Network Management System Database to invoke the Customer Update Stored Procedure to insert/update customer information in the Oracle Utilities Network Management System customer related staging tables.
  - Customization by putting placeholders for custom xsl and calls to pre and post transformation extension points for each transformation.
- The JMS consumer and BPEL process are configured to participate in a global transaction so that the BPEL process can issue rollback and commits on the queue. The BPEL process issues rollbacks on the queue when technical errors are encountered (Oracle Utilities Network Management System DB is down or connectivity issues) and the message is moved to the corresponding Error Queue.



Technical Process Flow for Customer Data Synchronization

### Successful Update

When the Oracle Utilities Network Management System tables are successfully updated, the integration layer sends a positive acknowledgement to the Oracle Utilities Customer Care and Billing response queue for Oracle Utilities Customer Care and Billing to process.

### Business Process Errors

When a business error is encountered during insert or update of one of the customer related staging tables, the Customer Update Stored Procedure returns an error to the integration layer and a negative acknowledgement is sent to the Oracle Utilities Customer Care and Billing response queue. Oracle Utilities Customer Care and Billing fixes the error and resends the message. The Oracle Utilities Network Management System Customer Update Stored Procedure issues a rollback on all successful activities (database operations) that were issued to the tables.

## Technical Errors

When a technical error is encountered the message is sent to the Oracle Utilities Customer Care and Billing JMS error queue and processing fails. Common technical errors would be if the Oracle Utilities Network Management System database is down or if there are connectivity errors. As mentioned, the integration can be configured to send email notification of errors or to store them in an error table.

## Integration Services and Components

### Integration Service

These values are cross referenced in the [configuration guidelines section](#) for the integration.

Name	Description
OUCCBOUNMSCustomerDataSyncReqEBF	CCB-NMS Customer Data Sync Request BPEL Process  This BPEL process reads messages from CCB request queue and merges the customer data to the NMS Customer Related Staging tables after successful transformation. The BPEL process includes transformations, extensions and error notifications.

### Adapter Services

Name	Description
OUCCBCustomerSyncReqJMSConsumer	CCB Customer Sync Request JMS Consumer  This is the JMS consumer service in BPEL responsible for listening to the CCB Request Queue. This is created as part of the BPEL process.
OUCCBCustomerSyncResponseJMSProducer	CCB Customer Sync Response JMS Producer  This is the JMS producer service in BPEL responsible for adding a response message to the CCB Response Queue. This is created as part of the BPEL process.
OUNMSUpdateCustomerStoredProc	NMS DB Adapter – Update Customer Stored Procedure. This is created as part of the BPEL process.

### JMS Queues

Name	Description
OUCCBCustomerDataSyncRequest	CCB Customer Data Sync Request Queue
OUCCBCustomerDataSyncRequestError	CCB Customer Data Sync Request Error Queue
OUCCBCustomerDataSyncResponse	CCB Customer Data Sync Response Queue
OUCCBCustomerDataSyncResponseError	CCB Customer Data Sync Response Error Queue

## Trouble Call Entry

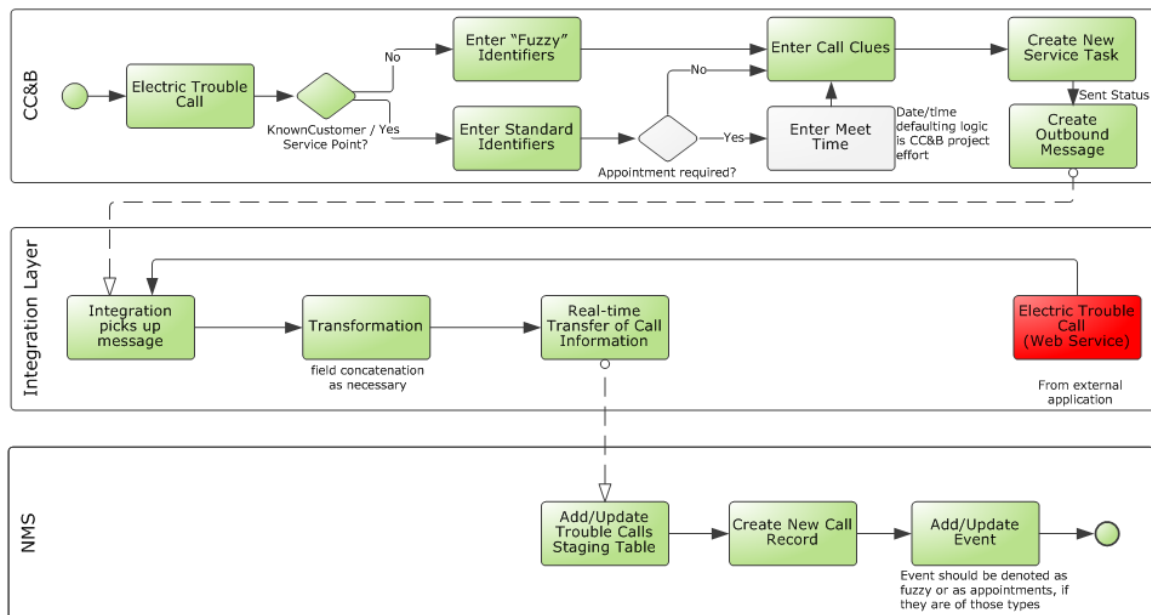
This process is a real time synchronous interface of the trouble calls created in Oracle Utilities Customer Care and Billing.

Oracle Utilities Network Management System is the central repository for trouble calls. However, trouble calls may originate in Oracle Utilities Customer Care and Billing and these trouble calls must be sent to Oracle Utilities Network Management System.

### Supported Functionality

This the integration point supports the following functionality

- Transmit to Oracle Utilities Network Management System trouble calls created, updated or canceled in Oracle Utilities Customer Care and Billing. The following types of calls are interfaced:
  - Electric trouble calls for a particular customer (known premise/service point)  
This includes entering the meeting time for job site appointments when there needs to be a planned outage to perform non-utility work at a location, such as tree removal near a power line or house painting.
  - Fuzzy calls  
When a fuzzy call is created at least one of the following call identifiers must be provided:
    - The caller's name
    - The caller's phone number
    - The caller's ID (i.e. 911 reference ID provided by the caller (911)).
    - Location must also be provided. A Location can be a
      - Street intersection (provide two street names) or
      - Street segment (provide a block number and a street name)
      - City and State are optional



Trouble Call Entry Business Flow Diagram

## Mapping

Trouble code mapping must be synchronized between the edge applications so that the trouble code sent from Oracle Utilities Customer Care and Billing is interpreted similarly when the trouble code is received by Oracle Utilities Network Management System.

## New Trouble Calls Created in Oracle Utilities Customer Care and Billing

When a trouble call is created in Oracle Utilities Customer Care and Billing, the contact name and contact phone passed to Oracle Utilities Network Management System are not always used as the customer name and customer phone stored in the incident record.

If the Generic IVR Adapter, which processes trouble calls received from Oracle Utilities Customer Care and Billing is run with the 'command line option '-docustquery' for customers that exist in the Oracle Utilities Network Management System Customer Model, the system uses the customer name and customer phone stored in the Oracle Utilities Network Management System customer model rather than the contact name and contact phone coming from Oracle Utilities Customer Care and Billing.

Make sure when running the Generic IVR Adapter in Oracle Utilities Network Management System that the command-line option '-docustquery' is not used if you want to store the contact name and contact number from Oracle Utilities Customer Care and Billing.

For more information on configuring this option please refer to [Chapter 3 Setting Up Oracle utilities Network Management System](#)



## Updates

The edge applications determine which fields can be updated and which fields are restricted. Depending on the needs of the customer, this decision is implementation specific since some customers are very sensitive about the ability to modify customer reported information so they require new calls to be entered for significant changes to a reported call while some only allow certain fields to be updated. No restrictions are applied as default.

### *Updating Phone Numbers*

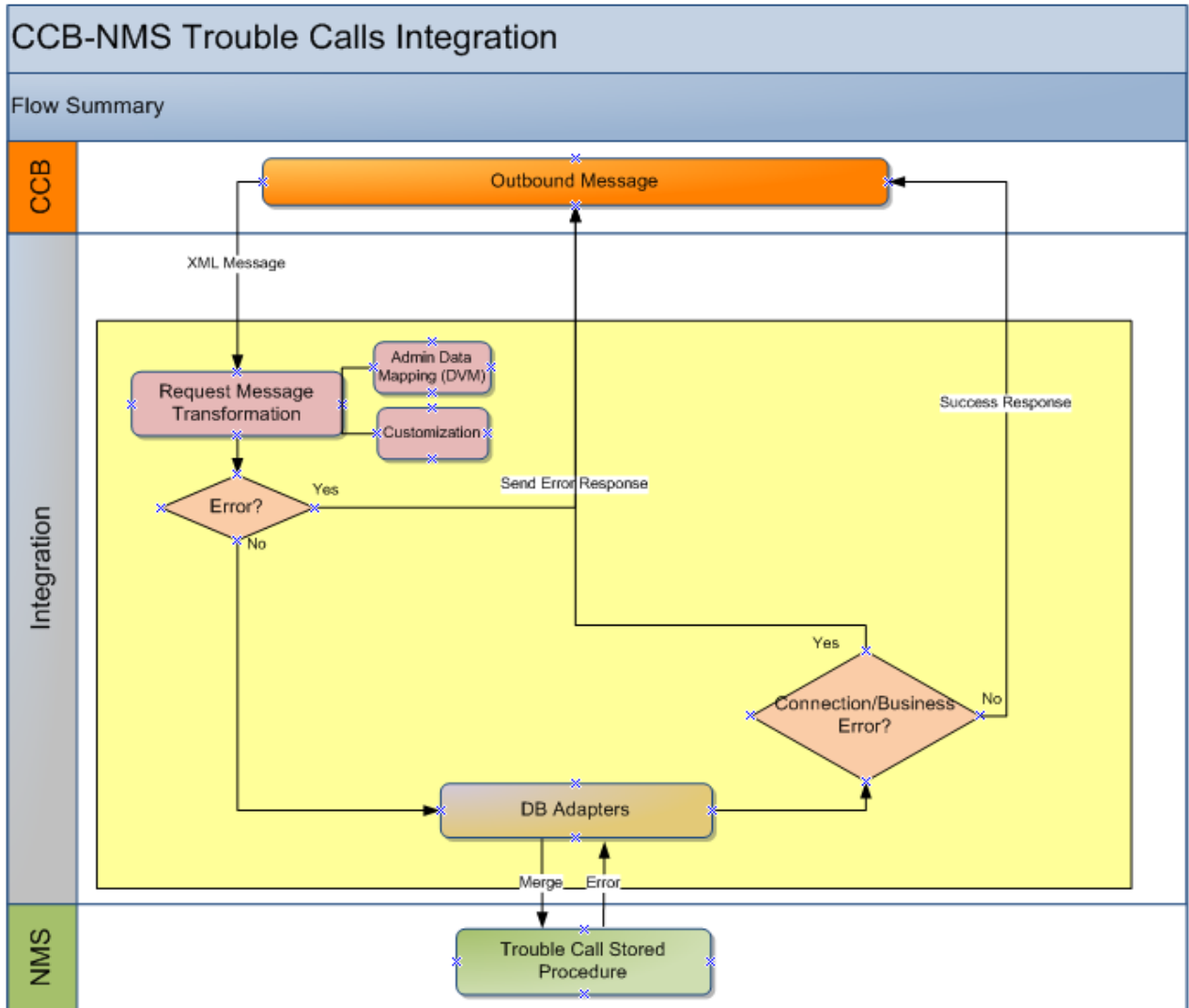
When Oracle Utilities Customer Care and Billing passes a contact phone to the integration layer, the integration layer removes all delimiters and characters and only passes the numeric value to Oracle Utilities Network Management System.

## Process Flow and Technical Details

This the integration point supports trouble calls created or updated in Oracle Utilities Customer Care and Billing and sent to Oracle Utilities Network Management System with the following processing:

When a trouble call is created in Oracle Utilities Customer Care and Billing as a Service Task, a synchronous xml message is sent to the BPEL Process. The BPEL process transforms the message to the equivalent Oracle Utilities Network Management System field format and invokes the Submit Call Stored Procedure to insert/update the trouble call information in the Oracle Utilities Network Management System Trouble Calls table. The BPEL process handles the following:

- Request Message transformation from the source (CCB) to the target (NMS) application format. DVM's are used for the transformation.
- Insert/Update of trouble calls in the Trouble Calls table using a DB Adapter to interact with the Oracle Utilities Network Management System Database to invoke the Oracle Utilities Network Management System trouble calls stored procedure that inserts/updates the trouble call record to the Trouble Calls table.
- Error handling and optional error notification
- Customization by putting placeholders for custom xsl and calls to pre and post transformation extension points for each transformation.



Technical Flow Diagram for Trouble Call Interface

### Successful Update

When the Oracle Utilities Network Management System tables are successfully updated, the integration layer sends a positive acknowledgement to the Oracle Utilities Customer Care and Billing response queue.

### Errors

When a business or technical error is encountered, the integration layer synchronously responds with an error to Oracle Utilities Customer Care and Billing. As mentioned, the integration can be configured to send email notification of errors or to store them in an error table.

## Integration Services and Components

### Integration Service

These values are cross referenced in the [configuration guidelines section](#) for the integration.

Name	Description
OUCCBOUNMSTroubleCallInterfaceEBF	CCB-NMS Trouble Call Interface BPEL Process  Main BPEL process that transforms incoming CCB trouble call message to NMS format and insert/update the trouble call record in NMS. The BPEL process includes transformations, extensions and error notifications.

### Adapter Services

Name	Description
OUNMSSubmitCallStoredProcedure	NMS DB Adapter – Insert Trouble Calls Stored Procedure. This is created as part of the BPEL process.

---

## Job History Query

This process is a real-time synchronous interface from Oracle Utilities Customer Care and Billing to retrieve job history information from Oracle Utilities Network Management System for a particular customer, location or call identifier and display the results back in Oracle Utilities Customer Care and Billing.

### Supported Functionality

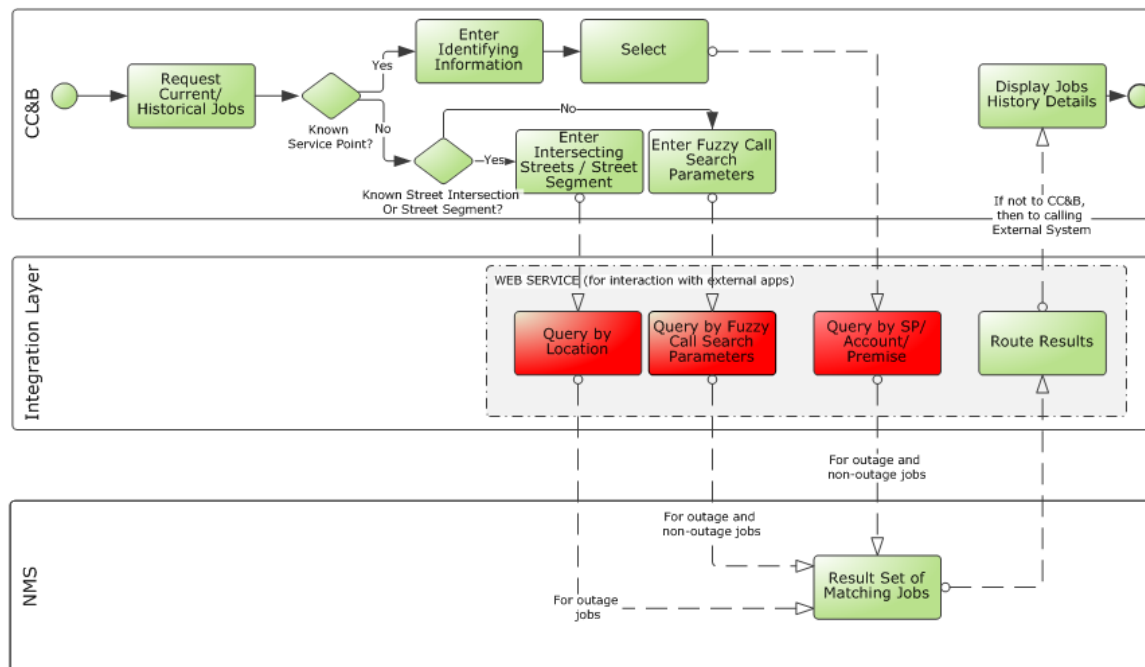
The job history query supports the following functionality:

- Query and view job history details from Oracle Utilities Customer Care and Billing using any of the search criteria:
  - Standard outage job history query for known customers. Search by IDs:
    - Service Point ID
    - Account ID
    - Premise ID
  - Nearby Outage Job History Query. Search by location:
    - Query by Street Intersection. The possible inputs to the query are:  
Street Intersection (street name and cross street)  
City (optional)  
State (optional)
    - Query by Street Segment. The possible inputs to the query are:  
Street segment (street name and block number)

City (optional)

State (optional)

- Fuzzy Call Identifier Query. This query can find the job details for a fuzzy call that was placed. This will require first looking up the fuzzy call from the Call History using any of the following criteria and finding the associated jobs.
  - Caller's Name
  - Caller's Phone Number
  - Call Identifier Number (911 Call Identifier)
  - External ID (Outage Call ID in Oracle Utilities Customer Care and Billing or IVR ID)
- Oracle Utilities Customer Care and Billing default display order of trouble calls with most recent at top.
- Amount of history to be retrieved will be restricted by the number of days defined in the integration layer.



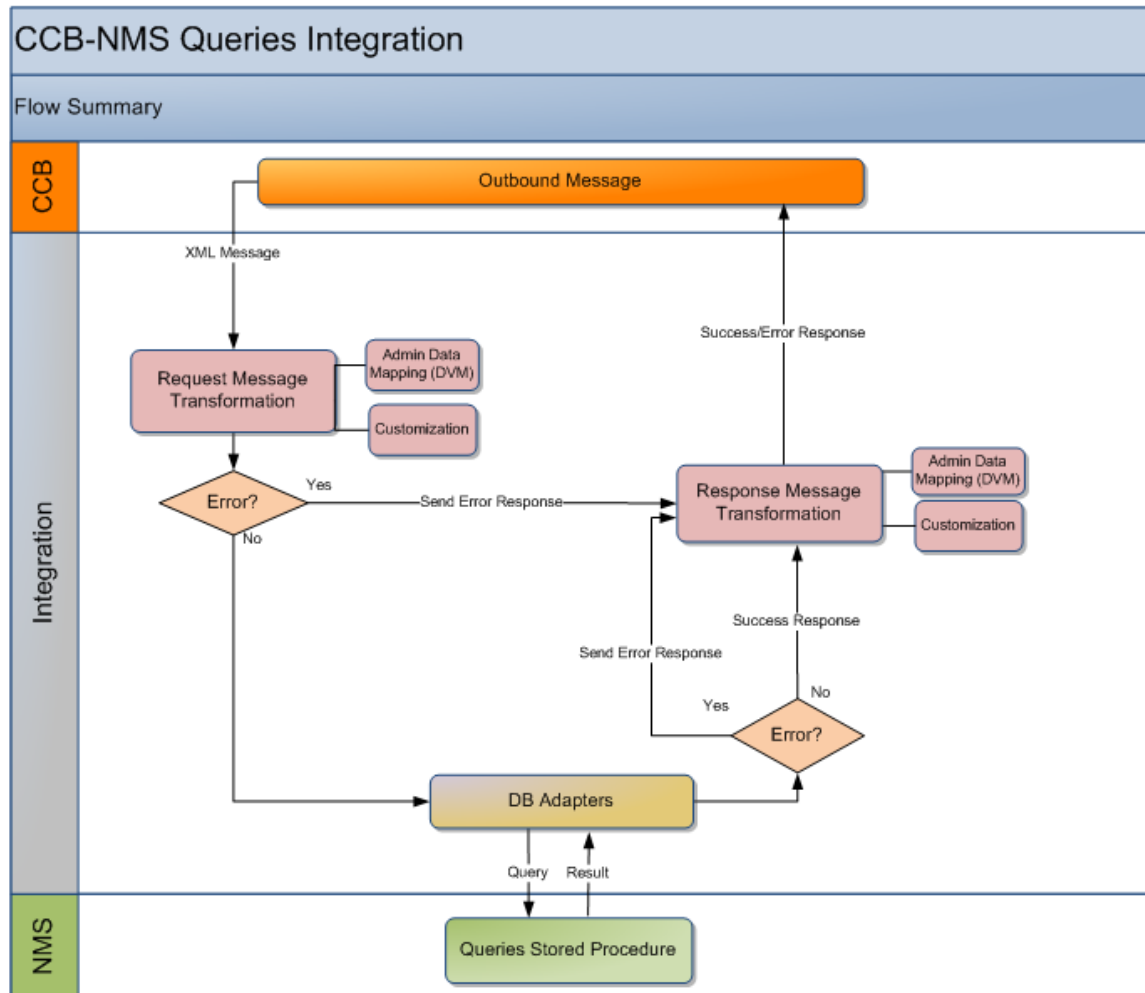
Job History Query Business Flow Diagram

## Assumptions and Constraints for Job History Query

- When searching Job History by contact phone, the integration layer removes all delimiters and characters from the contact phone that Oracle Utilities Customer Care and Billing and only passes the numeric value to Oracle Utilities Network Management System.
- To make the name search and street name search compatible, a wildcard character (%) must be added at the end of the value. Example. Smith%.

## Process Flow and Technical Details

Oracle Utilities Customer Care and Billing sends the query information in the form of xml messages which are transformed by the integration and sent to Oracle Utilities Network Management System. Oracle Utilities Network Management System responds with the Job History results based on the input criteria that it received. The response is transformed by the integration layer and sent to Oracle Utilities Customer Care and Billing.



Technical Flow Diagram for Job History Query

### Number of Days to Retrieve

The number of days of job history to retrieve from Oracle Utilities Network Management System is configurable from the integration layer. It is defined in the configuration properties file, service name = OUCCBOUNMSJobHistoryQuery and property name = NMS.NumberOfDaysOfHistory. This value is sent to Oracle Utilities Network Management System for use as the filter when the records are being returned to Oracle Utilities Customer Care and Billing.

## Errors

If the integration layer or Oracle Utilities Network Management System returns a business error while processing the inbound message, an error message is returned in the response message to Oracle Utilities Customer Care and Billing.

## Integration Services and Components

### Integration Service

These values are cross referenced in the [configuration guidelines section](#) for the integration.

Name	Description
OUCCBOUNMSJobHistoryQueryEBF	<p>Query NMS for Job History BPEL Process</p> <p>Synchronous BPEL process to transform incoming CCB message to NMS format and transform the response from NMS back to CCB format.</p> <p>This BPEL process will receive the CCB request messages and invoke NMS stored procedure using DB Adapter. The response from NMS is sent as response back to CCB after appropriate transformations.</p>

### Adapter Services

Name	Description
OUCCBOUNMSJobHistoryQueryAdapterService	<p>Query NMS for Job History Adapter Service</p> <p>This is the DB Adapter Service to invoke NMS stored procedure PK_CCB.JOB_HISTORY.</p>

## Trouble Call History Query

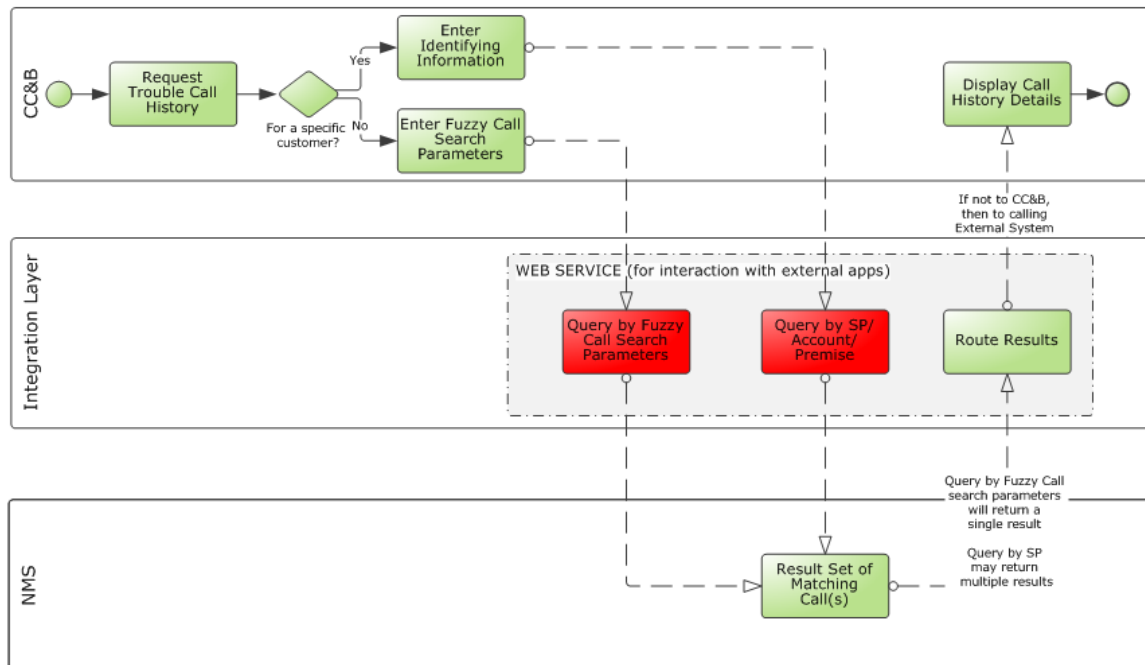
This process is a real-time synchronous interface from Oracle Utilities Customer Care and Billing to retrieve trouble call history information from Oracle Utilities Network Management System for a particular customer, location or call identifier and display the results back in Oracle Utilities Customer Care and Billing.

### Supported Functionality for Trouble Call History Query

The trouble call history query supports the following functionality:

- Query and view call history details from Oracle Utilities Customer Care and Billing using any of the search criteria:
  - Standard Trouble Calls History Query for known customers. Search by Ids:
    - Service Point ID
    - Account ID
    - Premise ID
  - Nearby Outage Job History Query. Search by location:
    - Query by Street Intersection. The possible inputs to the query are:

- Street Intersection (street name and cross street)
- City (optional)
- State (optional)
- Query by Street Segment. The possible inputs to the query are:
  - Street segment (street name and block number)
  - City (optional)
  - State (optional)
- Fuzzy Call Identifier Query. This query can find the trouble call details for a fuzzy call (calls that are not associated with a customer or with a device in Oracle Utilities Network Management System). Search by call identifiers:
  - Caller's Name
  - Caller's Phone Number
  - Call Identifier Number (911 Call Identifier)
  - External ID (Outage Call ID in Oracle Utilities Customer Care and Billing or IVR ID)
- Oracle Utilities Customer Care and Billing default display order of trouble calls with most recent at top.
- Amount of history to be retrieved will be restricted by the number of days defined in the integration layer.



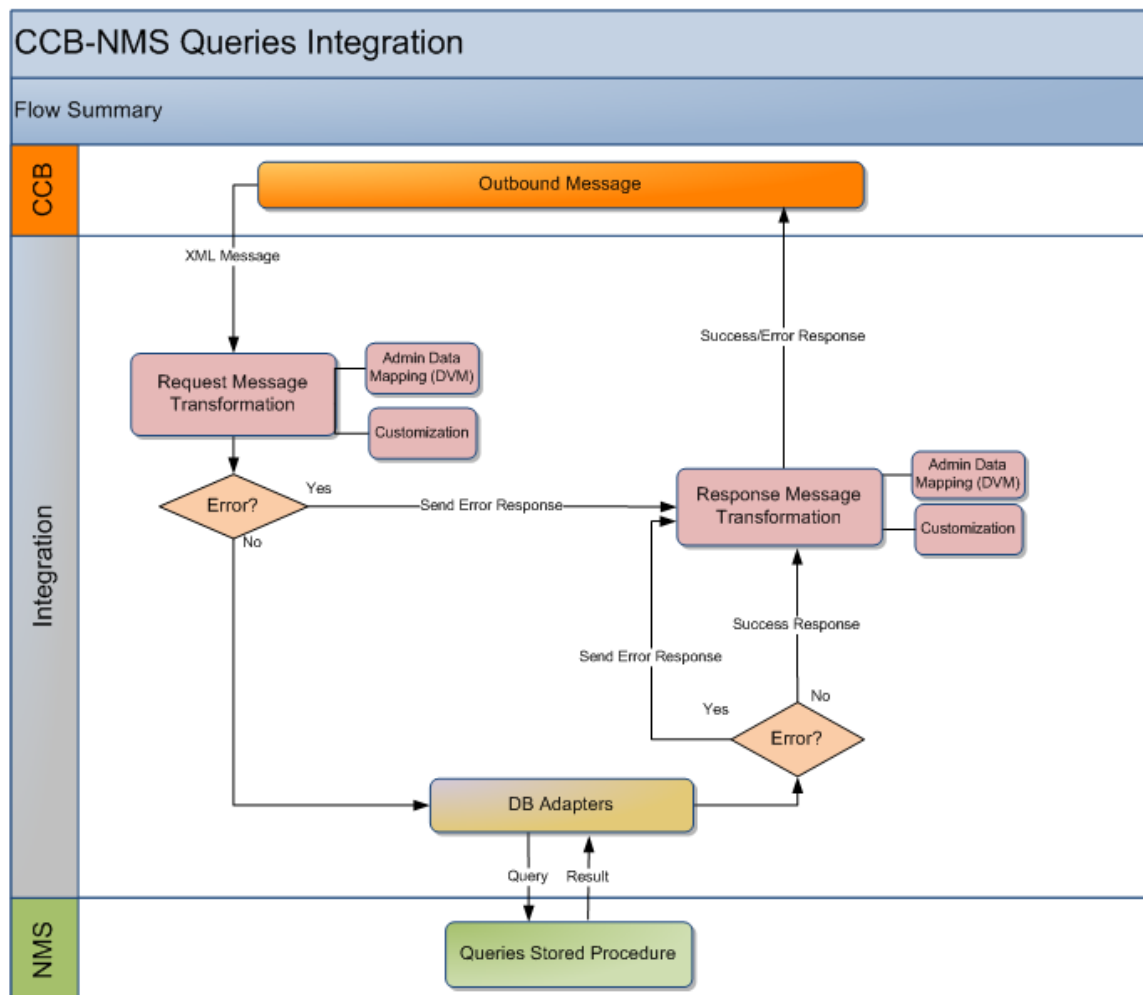
Trouble Call History Query Business Flow Diagram

## Assumptions and Constraints for Trouble Call History Query

- When searching Trouble Call History by contact phone, the integration layer removes all delimiters and characters from the contact phone that Oracle Utilities Customer Care and Billing passes to the integration and only passes the numeric value to Oracle Utilities Network Management System.
- To make the name search and street name search compatible, a wildcard character (%) must be added at the end of the value. Example. Smith%.

## Process Flow and Technical Details

Oracle Utilities Customer Care and Billing sends the query information in the form of xml messages which are transformed by the integration layer and sent to Oracle Utilities Network Management System. Oracle Utilities Network Management System responds with the Trouble Calls History based on the input criteria that it received. The response is transformed by the integration layer and sent to Oracle Utilities Customer Care and Billing.



Technical Flow Diagram for Trouble Call History Query



### Number of Days to Retrieve

The number of days of trouble call history to retrieve from Oracle Utilities Network Management System is configurable from the integration layer. It is defined in the configuration properties file, service name = OUCCBOUNMSTroubleCallsQueryEBF and property name = NMS.NoOfDays. This value is sent to Oracle Utilities Network Management System for use as the filter when the records are being returned to Oracle Utilities Customer Care and Billing.

### Errors

If the integration layer or Oracle Utilities Network Management System returns a business error while processing the inbound message, an error message is returned in the response message to Oracle Utilities Customer Care and Billing.

## Integration Services and Components

### Integration Service

These values are cross referenced in the [configuration guidelines section](#) for the integration.

Name	Description
OUCCBOUNMSCallHistoryQueryEBF	Query NMS for Trouble Calls History BPEL Process  Synchronous BPEL process to transform incoming CCB message to NMS format and transform the response from NMS back to CCB format. This BPEL process will receive the CCB request messages and invoke NMS stored procedure using DB Adapter. The response from NMS is sent as response back to CCB after appropriate transformations.

### Adapter Services

Name	Description
OUCCBOUNMSCallHistoryQueryAdapterService	Query NMS for Trouble Calls History Adapter Service  This is the DB Adapter Service to invoke NMS stored procedure PK_CCB.Call_History.

---

## Planned Outages Query

The primary owner of the planned outage data is Oracle Utilities Network Management System.

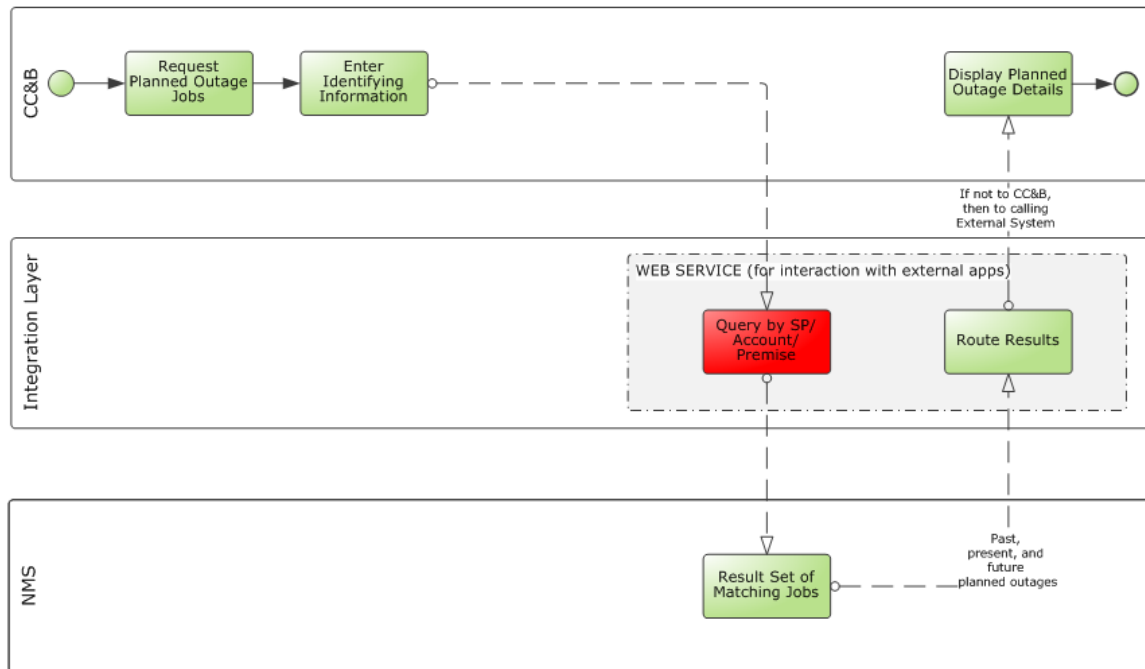
This process is a real-time synchronous interface from Oracle Utilities Customer Care and Billing to retrieve planned outages from Oracle Utilities Network Management System for a particular customer and display the results back in Oracle Utilities Customer Care and Billing.

### Supported Functionality for Planned Outage Query

The planned outage query supports the following functionality:

- Query and view planned outage jobs from Oracle Utilities Customer Care and Billing given a Service Point.

- Retrieve from Oracle Utilities Network Management System for display in Oracle Utilities Customer Care and Billing information for future, current and historical planned outages that will affect or have affected that service point.
  - Whether or not to retrieve all planned outages affecting the customer (past, present, and future), or only “active” ones (current and future, not past) is configurable.
  - Amount of history will be restricted by the number of days defined in the integration layer.

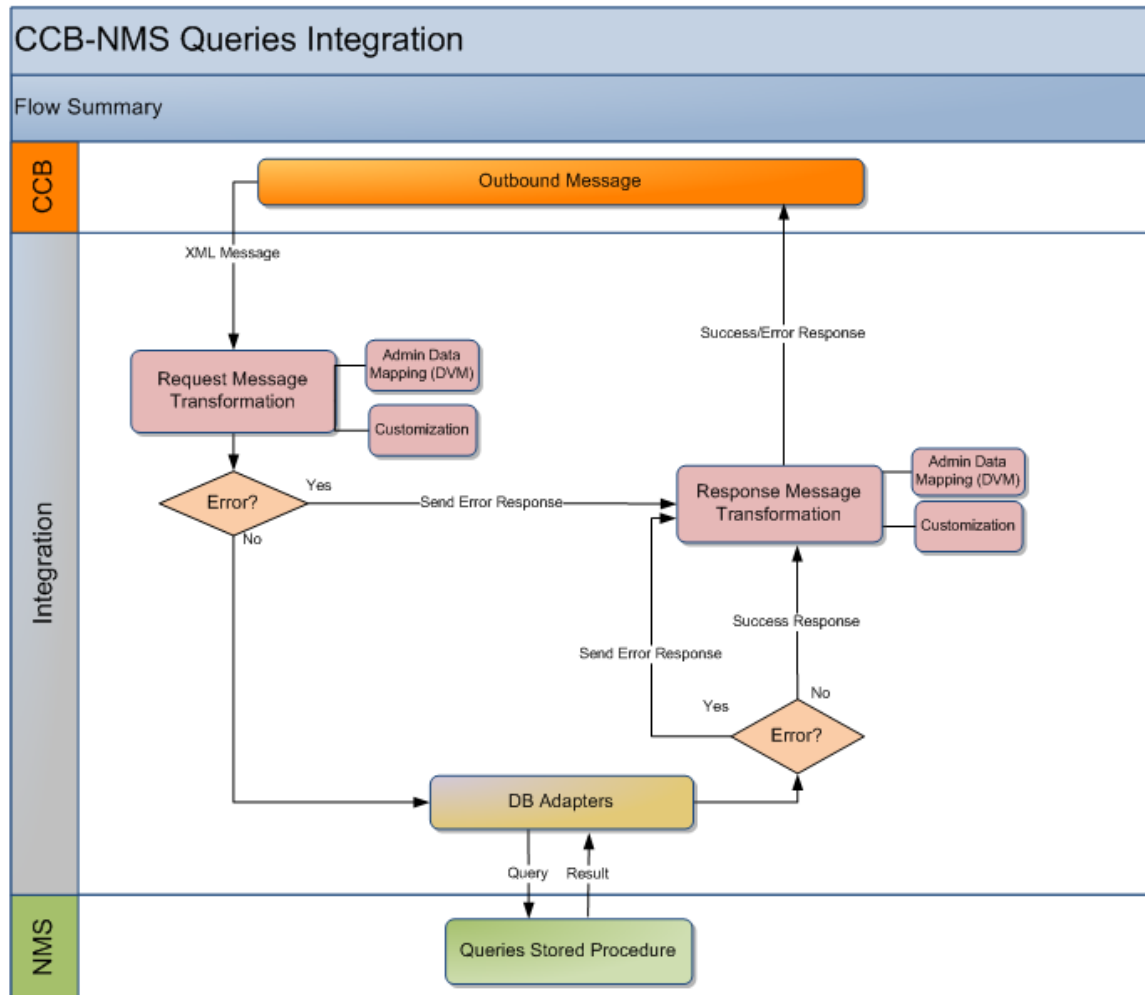


Planned Outages Query Business Flow Diagram

## Process Flow and Technical Details

Oracle Utilities Customer Care and Billing will send the query information in form of xml messages which will be transformed by the integration and sent to Oracle Utilities Network Management System.

Oracle Utilities Network Management System will respond back with the Planned Outage Jobs based on the input criteria that it received. The response will be transformed by the integration layer and sent to Oracle Utilities Customer Care and Billing.



Technical Flow Diagram for Planned Outage Query

### Number of Days to Retrieve

The number of days of past planned outage to retrieve from Oracle Utilities Network Management System is also an input to the Oracle Utilities Network Management System stored procedure and this value is configurable from the integration layer. It is defined in the configuration properties file, service name = OUCCBOUNMSPlannedOutagesQueryEBF and property name = NMS.NoOfDays.

From Oracle Utilities Customer Care and Billing, if the Include Past Planned Outages flag is true, the integration will send to Oracle Utilities Network Management System the defined number of days of past planned outage to retrieve from Oracle Utilities Network Management System obtained from the configuration file. If the Include Past Planned Outages flag is false, a value of zero is passed to Oracle Utilities Network Management System and the integration only retrieves the current and future planned outages.

### Errors

If the integration layer or Oracle Utilities Network Management System returns a business error while processing the inbound message, an error message is returned in the Response message to Oracle Utilities Customer Care and Billing.

## Integration Services and Components

### Integration Service

These values are cross referenced in the [configuration guidelines section](#) for the integration.

Name	Description
OUCCBOUNMSPlannedOutagesQueryEBF	Query NMS for Planned Outage Jobs BPEL Process  Synchronous BPEL process to transform incoming CCB message to NMS format and transform the response from NMS back to CCB format. This BPEL process will receive the CCB request messages and invoke NMS stored procedure using DB Adapter. The response from NMS is sent as response back to CCB after appropriate transformations.

### Adapter Services

Name	Description
OUCCBOUNMSPlannedOutagesQueryAdapterService	Query NMS for Planned Outages Adapter Service  This is the DB Adapter Service to invoke NMS stored procedure PK_CCB.switching_history.

## Part 2: Implementing the Delivered Integrations

This section provides details about how to configure the participating applications and the middle layer for the integration. Information on error handling, monitoring, customization options and data mapping is also included.

### Chapter 3: Configuration Guidelines

This chapter provides detail into the required configuration settings for the integration, and discusses details related to:

- Set up Oracle Utilities Customer Care and Billing
- Set up Oracle Utilities Network Management System
- Set up the direct the integration

---

#### Setting up Oracle Utilities Customer Care and Billing

The following sections provide details into the Oracle Utilities Customer Care and Billing configurations needed to facilitate the integration. Some configurations described may be required for general functionality and do not necessarily relate directly to the integration; however these are called out as particularly significant configuration items. The inclusion of such items does not mean that other general items that are not mention do not need to be configured.

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

At a high-level, you will complete the following steps in Oracle Utilities Customer Care and Billing to support the integration:

1. Configure Administrative Tables
  - Characteristic Type
  - Feature Configuration
  - Service Type
  - Meter Type
2. Configure the Sync Request Process

- Batch Code
  - BO Algorithms
  - MO Algorithms
  - Maintenance Objects
  - Business Object used for the Oracle Utilities Network Management System Data Sync Request
3. Configure JMS
    - Weblogic Server JMS
  4. Configure Outbound Message to pass messages with the integration layer
    - XAI JNDI Server
    - XAI JMS Queue
    - XAI JMS Connection
    - XAI Sender
    - Outbound Message Type
    - External System

## Configure Administrative Tables

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

For more information about configuring Oracle Utilities Customer Care and Billing, see the Oracle Utilities Customer Care and Billing *User Guide* section titled “Setting up the System to Enable FA Integration.”

## Characteristic Types

The following characteristic types must be defined to facilitate the integration.

Characteristic Type	Guideline	Characteristic Entity Collection	Corresponding DVM
<b>Critical Priority Characteristic Type</b>	<ul style="list-style-type: none"> <li>- Premise characteristic used to define the critical priority for the premise.</li> <li>- Pre-defined characteristic type.</li> <li>- The pre-defined values listed here must exactly match values in the DVM for CCB Critical Priority Code.</li> </ul>	Include <b>Premise</b>	OUCCB_OUNMS_Serv_C_Priority
<b>Medical Priority Characteristic Type</b>	<ul style="list-style-type: none"> <li>- Premise characteristic used to define the medical priority for the premise.</li> <li>- Pre-defined characteristic type.</li> </ul>	Include <b>Premise</b>	OUCCB_OUNMS_Serv_D_Priority

Characteristic Type	Guideline	Characteristic Entity Collection	Corresponding DVM
	- The pre-defined values listed here must exactly match values in the DVM for CCB Medical Priority Code.		
<b>Key Priority Characteristic Type</b>	<ul style="list-style-type: none"> <li>- Premise characteristic used to define the key priority for the premise.</li> <li>- Pre-defined characteristic type.</li> <li>- The pre-defined values listed here must exactly match values in the DVM for CCB Key Priority Code.</li> </ul>	Include <b>Premise</b> .	OUCCB_OUNMS_Serv_K_Priority
<b>Location City</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify the location city for an outage without a premise</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_CITY (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Location State</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify the location state for an outage without a premise</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_STATE (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Location 1</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a location used for an outage without a premise. (The location would be either a street name for location type street segment or intersection street1 for location type street intersection)</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_LOCN1 (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Location 2</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a location (intersection street2) used to for an outage without a premise if the location type is a street intersection.</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_LOCN2 (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Block Number</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a block number used for an outage without a premise if the location type is a street segment.</li> <li>- Adhoc characteristic type.</li> <li>- The Block Number adhoc value must be numeric.</li> <li>- CCB Demo Data: CI_BLKNNBR (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Contact Name</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a contact name used for an outage without a premise.</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_CNTNM (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A

Characteristic Type	Guideline	Characteristic Entity Collection	Corresponding DVM
<b>Contact Number</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a contact number used for an outage without a premise.</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_CNTPN (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Call Identifier</b>	<ul style="list-style-type: none"> <li>- Characteristic used to identify a call identifier used for an outage without a premise.</li> <li>- Adhoc characteristic type.</li> <li>- CCB Demo Data: CI_CALL (Sample)</li> </ul>	Include <b>Service Task</b> .	N/A
<b>Outage Codes 1 - N</b>	<ul style="list-style-type: none"> <li>- These characteristics are used to describe the outage problem. <ul style="list-style-type: none"> <li>• Create at least one and up to N pre-defined characteristic type. N being the number of outage codes needed by the implementation.</li> <li>• For each characteristic type, define its list of valid values</li> </ul> </li> <li>- CCB Demo Data: CI_OUT01, CI_OUT02, CI_OUT03, CI_OUT04, CI_OUT05, CI_OUT06, CI_OUT07, CI_OUT08, CI_OUT09 (Samples)</li> </ul>	Include <b>Service Task</b> .	N/A

## Feature Configuration

### Schema Constants

To manage feature configuration:

1. Navigate to Admin Menu > Feature Configuration.
2. Create new feature configuration with **Schema Constants** as the Feature Type or select a feature configuration with **Schema Constants** as the Feature Type if one already exists and enter required option types and values needed for this the integration.
3. Populate entries for the applicable options.

Option	Notes
<b>Home Phone Type</b>	This option type indicates the user defined home phone number type code. For this Option Type, the Option Value must be a valid Phone Number Type defined in the Phone Type Table
<b>Business Phone Type</b>	This option type indicates the user defined business phone number type code. For this Option Type, the Option Value must be a valid Phone Number Type defined in the Phone Type Table.
<b>Device Geographic Type</b>	This option type indicates the user defined device ID geo type code. For this Option



Option	Notes
	Type, the Option Value must be a valid Geographic Type defined in the Geographic Type Table.
<b>Critical Priority Characteristic Type</b>	This option type indicates the user defined critical priority characteristic type code. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Medical Priority Characteristic Type</b>	This option type indicates the user defined medical priority characteristic type code. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Key Priority Characteristic Type</b>	This option type indicates the user defined key priority characteristic type code. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Contact Name Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a contact name on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Contact Number Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a contact number on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Identifier Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a call identifier on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Street Name Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a street name on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Cross Street Name Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a cross street name on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call Block Number Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a block number on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call City Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a city on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.
<b>Outage Call State Characteristic Type</b>	This option type indicates the characteristic type code your implementation uses to capture a state on a trouble call. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.

## NMS Integration

To manage feature configuration:

1. Navigate to Admin Menu > Feature Configuration.

2. Create new feature configuration with **NMS Integration** as the Feature Type or select a feature configuration with **NMS Integration** as the Feature Type if one already exists and enter required option types and values needed for this the integration.
3. Populate entries for the applicable options.

Option	Notes
<b>External System</b>	This option defines the external system used for query outbound messages created from the outage management information portal page. For this Option Type, the Option Value must be a valid External System defined in the External System Table.
<b>Outbound Message Type - Call History</b>	This option defines the outbound message type used for the call history query outbound messages created from the outage management information portal page. For this Option Type, the Option Value must be a valid Outbound Message Type defined in the Outbound Message Type Table.
<b>Outbound Message Type - Job History</b>	This option defines the outbound message type used for the job history query outbound messages created from the outage management information portal page. For this Option Type, the Option Value must be a valid Outbound Message Type defined in the Outbound Message Type Table.
<b>Outbound Message Type - Call History</b>	This option defines the outbound message type used for the planned outage query outbound messages created from the outage management information portal page. For this Option Type, the Option Value must be a valid Outbound Message Type defined in the Outbound Message Type Table.
<b>Outage Group Code Characteristic Type Prefix</b>	This option defines the prefix used for trouble call outage group code characteristic types. The system uses this to build a dropdown of outage group codes during trouble call processing. For this Option Type, the Option Value must be a valid Characteristic Type defined in the Characteristic Type Table.

## Service Type

Every service point type references a service type. The service type defines the type of service you provide to your customer (i.e. electric, water, gas).

The codes defined here must exactly match the values defined in the DVM specified below.

Navigation	Guideline	Corresponding DVM
Admin Menu > Service Type	- Define your service types.	OUCCB_OUNMS_AccountType

## Meter Type

Every meter references a meter type. The meter type defines the type of service and common characteristics shared by its meters.

The codes defined here must match the values defined in the DVM specified below.

Navigation	Guideline	Corresponding DVM
Admin Menu > Meter Type	- Define your meter types.	OUCCB_OUNMS_MeterType

## Configure the Sync Request Process

The Sync Request Process is used to synchronize customer data from Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System.

### Batch Code

This is the batch process to run the sync request. It is a generic batch process that is used for different sync processes. It has a couple of parameters that can be used to control which sync request BOs to process.

Batch Code	Description
F1-SYNRQ	Sync Request Monitor Process

Batch Parameters	Parameter Description	Value
✓maintenanceObject	Sync Request maintenance object.	F1-SYNC REQ (This is the defaulted value.)
✓isRestrictedByBatchCode	The value of true will restrict processing to sync requests whose current state is linked to this batch code.	
✓restrictToBusinessObject	Enter a business object code here to limit the process to sync requests linked to this business object.	C1-NMSSPSyncRequest (To run only the NMS customer sync request, populate this value)_
✓restrictToBOStatus	Enter a status code here to limit the process to sync requests in this state.	PENDING (To only process sync request, in Pending status, populate this value)_

## BO Algorithms

Algorithm Type	Description
C1-CAPNMSSPI	<p>This pre-processing algorithm creates the initial snapshot for the sync request. See the algorithm description in the system for details on how to specify the parameters below.</p> <ol style="list-style-type: none"> <li>1. Define the read BOs the algorithms use to build the initial/final snapshot. The base product provides C1-NMSPerson, C1-NMSAccount, C1-NMSSA, C1-NMSSP, MDMPremise, C1-NMSMeter, and C1-NMSItem for this purpose. If additional elements are needed in the sync request, your implementation may create a child of any of these BOs and add the element under a group called &lt;customElements&gt;. This ensures that the elements are included in the sync request message at the proper group nodes. With this set up any custom translation can be implemented at the integration layer.</li> <li>2. Define the data area that holds the elements needed in the snapshot. The base product provides C1-NMSSPBasedSnapshot for this purpose. Your implementation should not have to create a custom data area as this already provides &lt;customElements&gt; nodes throughout its schema to allow for the addition of any elements not included in the base solution.</li> <li>3. It is possible to filter qualifying SP types for the sync. If your implementation wishes to do so, define these values in the algorithm so that only changes to the desired SP types are communicated across to NMS.</li> <li>4. It is possible to specify custom read BOs by SP Type. To do so, simply define your custom BO along with the SP Type you wish to use it with. The algorithm will use this value instead of those defined in (a).</li> </ol>
C1-MDM-TMOT	<p>This monitor algorithm sets a timeout limit on the receipt of a response from the external system. Define the number of hours your implementation wishes to wait for a response from NMS before transitioning the sync request into the Error state.</p>
F1-TD-CREATE	<p>This algorithm creates a To Do Entry. At a minimum, your implementation will have to define the To Do Type to use in creating the To Do Entry and the Characteristic Type For Log Entry to be used in linking the To Do Entry to the sync request via its logs. The base product provides F1-SYNRQ and F1-TODO, respectively, for this purpose. For details on the other parameters used by this algorithm, see the algorithm type description.</p>

## MO Algorithms

Configure the MO Audit algorithms. MO Audit algorithms contain the logic to instantiate a sync request (as long as one does not already exist in the initial state for the MO-Primary Keys combination). A generic algorithm F1-GCHG-CDCP comes with the base product and is plugged in on MOs that need to instantiate sync requests for the same MO. This algorithm instantiates the BOs defined in the Sync Request BO MO Option (see Maintenance Objects below). For MOs that need a sync request instantiated for a different MO (For example: changes to the Person or Account MO need to be communicated via an SP sync request) will need unique algorithms that contain this logic.

Algorithm Type	Description
C1-PERCDCSP	This algorithm instantiates an SP-based sync request whenever a change to the Person MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-ACCTCDCSP	This algorithm instantiates an SP-based sync request whenever a change to the Account MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-SACDCSP	This algorithm instantiates an SP-based sync request whenever a change to the SA MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-PREMCDCSP	This algorithm instantiates an SP-based sync request whenever a change to the Premise MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-SPICDCSP	This algorithm instantiates an SP-based sync request whenever a change to the SP/Item MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-SPMCDSP	This algorithm instantiates an SP-based sync request whenever a change to the SP/Meter MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-ITEMCDCSP	This algorithm instantiates an SP-based sync request whenever a change to the Item MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.
C1-MTRCDCSP	This algorithm instantiates an SP-based sync request whenever a change to the Meter MO is detected. Define the sync request BO to be instantiated in the algorithm's parameters.

## Maintenance Object

Maintenance Object	Description
PERSON	Specify the MO Audit algorithm configured in the previous section.
ACCOUNT	Specify the MO Audit algorithm configured in the previous section.
SA	Specify the MO Audit algorithm configured in the previous section.

SP	Specify the generic MO Audit algorithm F1-GCHG-CDCP. Also specify the C1-NMSSPSyncRequest BO in the Sync Request BO MO Option.
PREMISE	Specify the MO Audit algorithm configured in the previous section.
SP/ITEM	Specify the MO Audit algorithm configured in the previous section.
SP/METER	Specify the MO Audit algorithm configured in the previous section.
ITEM	Specify the MO Audit algorithm configured in the previous section.
METER	Specify the MO Audit algorithm configured in the previous section.

## Business Object

Business Object	Description
C1-NMSSPSyncRequest	<p>This business object defines the behavior of the outbound sync request for NMS. It contains the schema elements monitored and synchronized to NMS.</p> <p>The following BO Options must be configured to create the outbound sync request:</p> <ul style="list-style-type: none"> <li>- Outbound Message Type: This contains a reference to the outbound message BO to use. The base package includes BO C1-NMSSPSyncReqOutMsg for the NMS SP Sync. Refer to Defining Outbound Message Types in the user documentation for more information.</li> </ul> <p>External System: This contains the reference to the outbound message type and its corresponding configuration for communicating with the external system. The base package includes the message XSL C1-CCBJMSQAddNamespace.xsl. Please refer to External Systems in the user documentation for more information.</p> <p>Specify the pre-processing algorithm configured in the previous section.</p> <p>Specify the time out algorithm as a monitor algorithm on the Awaiting Acknowledgement state for this BO.</p> <p>Specify the To Do creation algorithm on the Error state for this BO</p> <p>Depending on the technology used to communicate the sync request to the external system, you may need to create your own enter algorithm and plug it into the Send Request state. The base package comes with an algorithm that creates a message and drops it into a JMS Queue. If your implementation uses this algorithm (C1-CR-OUTMSG), you must define the BO Options for External System and Outbound Message Type.</p>

For more information about the sync request process, the business objects, maintenance objects and other components use for this process, see the Oracle Utilities Framework *User Guide* section titled "Data Synchronization"

## Configure JMS

This section describes the JMS configuration to be done in the Oracle Utilities Customer Care and Billing weblogic server and in the Oracle Utilities Customer Care and Billing deployment xml files. The configuration in this section will be used for receiving JMS messages from the integration layer.

This configuration is only needed in the Customer Synchronization Integration process which uses jms queues.

### Weblogic Server JMS

For the JMS configuration in the Oracle Utilities Customer Care and Billing Weblogic server login to the console using the URL `http://<server_name>:<port_number>/console`

For example: `http://ccbserver:7001/console`

### JMS Module

To create a JMS Module to be used for remote queue configuration:

Create a new JMS Module in the weblogic console.

1. Enter the weblogic console and create a new JMS module.
2. Enter a meaningful name for the JMS module. This JMS Module is used to create configurations which consume messages from remote weblogic queues.

For example: `NMSIntegrationModule`

### Foreign Server

To create the Foreign Server to be used for remote queue configuration:

Create a new Foreign Server under the JMS Module in the weblogic console.

1. Enter the weblogic console and select the JMS module created for the integration.
2. Create a Foreign server under the JMS Module.
3. Add the following for the Foreign Server
  - **Name** – Name for the Foreign Server  
For example: `CCBNMSForeignServer`
  - **JNDI Initial Context Factory** – *`weblogic.jndi.WLInitialContextFactory`*
  - **JNDI Connection URL** – Add the URL for the Integration SOA Server  
For example: *`t3://sosserv.com:8002`*
  - **JNDI Properties Credential** – Password for the SOA Server user
  - **JNDI Properties** - *`java.naming.security.principal=<SOA Server user>`*  
*For example: `weblogic`*

4. Under the Foreign server create a Foreign Destination for each remote queue.

- **Name** – Name of foreign destination
- **Local JNDI Name** – Add a local JNDI name for the Integration Queue. Local JNDI name will later be added manually as part of configuration in the weblogic-ejb-jar.xml → <weblogic-enterprise-bean> → <message-driven-descriptor> → <destination-jndi-name>
- **Remote JNDI Name** – JNDI name of the queue on the Integration SOA Server

For example: the Customer Sync the integration point that uses a queue, one destination is created.

Destination Name	Local JNDI Name	Remote JNDI Name
OUCCBCustomerDataSyncResponse	jms/LocalOUCCBCustomerDataSyncResponse	jms/OUCCBCustomerDataSyncResponse

5. Under the Foreign server create a Remote Connection Factory.

- **Name** – Name of remote connection factory
- **Local JNDI Name** – Add a local JNDI name for the Integration Connection Factory. This JNDI name will later be added manually as part of configuration in the weblogic-ejb-jar.xml → <weblogic-enterprise-bean> → <message-driven-descriptor> → <connection-factory-jndi-name>
- **Remote JNDI Name** – JNDI name of the JMS Connection factory on the Integration SOA Server

Connection Factory Name	Local JNDI Name	Remote JNDI Name
OUCCBOUNMSConnectionFactory	jms/LocalOUCCBOUNMSConnectionFactory	jms/ OUCCBOUNMSConnectionFactory

### Oracle Utilities Customer Care and Billing Configuration File Changes

The Oracle Utilities Customer Care and Billing configuration files ejb-jar.xml and ejb-weblogic-jar.xml need to be modified to configure Message Driven Beans (MDB) which will receive messages from the Integration queues. These files are part of the Oracle Utilities Customer Care and Billing Enterprise Archive (EAR) file and the Oracle Utilities Customer Care and Billing application needs to be redeployed after these changes are made.

It is recommended that you use the Oracle Utilities Customer Care and Billing template and CM (Customer Modification) feature to make changes to these configuration files. This will ensure that your modifications are not overwritten by any future application patches.

Configuration files such as config.xml, ejb-jar.xml and ejb-weblogic-jar.xml are managed through template configuration files which reside in the environment's etc directory. When the initialSetup.sh script is executed environment specific information is combined with the template to create the target file which is then deployed to the correct location. When the environment is started up (spl.sh start) the changes will be automatically deployed to WebLogic.

For example, the template that is used to derive ejb-jar.xml is etc/ejb-jar.xml.template



It is possible to override the existing template by providing another template file in the same location, with the same name but prefixed with ".cm". For example, if etc/cm.ejb-jar.xml.template is found when initialSetup is run, the cm.ejb-jar.xml.template will be used to generate the target ejb-jar.xml file.

To enable your changes for Oracle Utilities Network Management System the integration it is recommended that you first make a "CM" copy of the existing template and make your changes to the CM version. If there are any problems with starting the application it is a simple process to delete the CM versions of the files and rerun initialSetup to regenerate and redeploy the original versions.

Working example configuration files are available for download from My Oracle Support in Patch number 9974118 - CCB - MDM/NMS INTEGRATION CONFIGURATION EXAMPLES. Before installing the examples please read the Product Fix Design document included in the patch.

For simplicity, in the following examples we refer to the names of the target configuration files, but you should make your changes in the etc/cm.<target file>.template version of the file and then execute initialSetup.sh (Unix) or initialSetup.cmd (Windows) to deploy the generated file.

PLEASE NOTE: If you make CM versions of the template files and later install a patch which updates the base template, the CM version will not be updated.

To create MDB to receive messages from the Oracle Utilities Customer Care and Billing Inbound queue:

Create a new MDB to receive messages from the customer sync inbound queue.

1. Create an MDB for the Oracle Utilities Customer Care and Billing inbound queue to receive messages and invoke Oracle Utilities Customer Care and Billing service.
2. Modify the ejb-jar.xml and weblogic-ejb-jar.xml to configure the MDBs.
3. Add the <message-driven> and <container-transaction> tag for each inbound queue in the ejb-jar.xml. Add a security role with role cisusers in the ejb-jar.xml

For example:

```
<?xml version="1.0" encoding="UTF-8"?>
<ejb-jar>
  <display-name>ServiceBean</display-name>
  <enterprise-beans>
    <!--Customer Sync Integration Point -->
    <message-driven>
      <description>MDB for OUCCBCustomerDataSyncResponse </description>
      <display-name>OUCCBCustomerDataSyncResponse</display-name>
      <ejb-name> OUCCBCustomerDataSyncResponse</ejb-name>
      <ejb-class>com.splwg.ejb.mdb.MessageProcessor</ejb-class>
      <messaging-type>javax.jms.MessageListener</messaging-type>
      <transaction-type>Bean</transaction-type>
      <message-destination-type>javax.jms.Queue</message-destination-
type>
    </message-driven>
  </enterprise-beans>
  <assembly-descriptor>
    <security-role>
      <role-name>cisusers</role-name>
```

```

        </security-role>
        <!--Customer Sync Integration Point -->
        <container-transaction>
            <method>
                <ejb-name> OUCCBCustomerDataSyncResponse</ejb-name>
                <method-name>onMessage</method-name>
            </method>
            <trans-attribute>NotSupported</trans-attribute>
        </container-transaction>
    </assembly-descriptor>
</ejb-jar>

```

4. Modify the weblogic-ejb-jar.xml. Add the <weblogic-enterprise-bean> tag for each inbound queue. Add a security role with role cisusers.
5. The following are references in the <weblogic-enterprise-bean> tag.
  - a. <ejb-name> - MDB Name given in ejb-jar.xml
  - b. <destination-jndi-name> - JNDI name provided in JMS Module→ Foreign Server →Foreign Destination →Local JNDI Name
  - c. <connection-factory-jndi-name> - JNDI name provided in JMS Module→ Foreign Server→Remote Connection Factory→Local JNDI Name

For example:

```

<?xml version="1.0" encoding="UTF-8"?>
<weblogic-ejb-jar xmlns="http://www.bea.com/ns/weblogic/90"
xmlns:j2ee="http://java.sun.com/xml/ns/j2ee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.bea.com/ns/weblogic/90
http://www.bea.com/ns/weblogic/90/weblogic-ejb-jar.xsd">
    <weblogic-enterprise-bean>
        <ejb-name>SPLServiceBean</ejb-name>
        <jndi-name>spl/servicebean</jndi-name>
    </weblogic-enterprise-bean>
    <!--Customer Sync Integration Point -->
    <weblogic-enterprise-bean>
        <ejb-name> OUCCBCustomerDataSyncResponse</ejb-name>
        <message-driven-descriptor>
            <pool>
                <max-beans-in-free-pool>5</max-beans-in-free-pool>
                <initial-beans-in-free-pool>1</initial-beans-in-free-pool>
            </pool>
            <destination-jndi-name>jms/LocalOUCCBCustomerDataSyncResponse
        </destination-jndi-name>
            <connection-factory-jndi-
name>jms/LocalOUCCBOUNMSConnectionFactory</connection-factory-jndi-name>
        </message-driven-descriptor>
    </weblogic-enterprise-bean>
    <security-role-assignment>
        <role-name>cisusers</role-name>
        <principal-name>cisusers</principal-name>
    </security-role-assignment>
</weblogic-ejb-jar>

```

## Outbound Configuration

The following sections identify settings required to pass messages with the integration layer.

### XAI JNDI Server

To create XAI JNDI Server configured to communicate with the integration layer:

Create a new XAI JNDI Server which points to the Integration SOA server.

1. Navigate to Admin Menu, XAI JNDI Server. .  
For example: CI\_NMS\_JNDI
2. Enter XAI JNDI Server name and description.  
For example: CCB-NMS Integration server
3. Populate the Provider URL in the format **t3//<SOA Server>: <SOA Port>**  
For example: t3://soaserver.us.oracle.com:8002

### XAI JMS Queue

To create XAI JMS Queue

Create a new XAI JMS Queue for the Integration Queue which Oracle Utilities Customer Care and Billing will send messages to.

1. Navigate to Admin Menu, XAI JMS Queue.
2. Enter the following

**XAI JMS Queue** – Queue name in Oracle Utilities Customer Care and Billing

**Description** – Queue description

**Queue Name** – JNDI name of the queue on the Integration server

For example: jms/OUCCBNMSCustomerDataRequest

**Target Client Flag** – **JMS**

**XAI JNDI Server** – Select XAI JNDI Server created for this the integration as described in the XAI JNDI Server section

For example: Customer Data Sync Integration Point

XAI JMS Queue	Description	Queue Name	Target Client Flag	XAI JNDI Server
CI_NMSCDSync	Customer Data Sync Request	jms/OUCCBCustomerDataSyncRequest	JMS	CI_NMS_JNDI

## XAI JMS Connection

To create XAI JMS Connection

Create a new XAI JMS Connection which will be used to connect to the Integration Queues.

1. Navigate to Admin Menu, XAI JMS Connection.
2. Enter the following

**XAI JMS Connection** – Connection name in Oracle Utilities Customer Care and Billing

**Description** – Connection description

**XAI JNDI Server** – Select XAI JNDI Server created for this the integration as described in the XAI JNDI Server section

**JNDI ConnectionFactory** – JNDI name of the Connection factory on the Integration server  
For example: jms/OUCCOUNMSConnectionFactory

For example:

XAI JMS Connection	Description	XAI JNDI Server	JNDI Connection Factory
CI_NMS_CF	CCB NMS Integration Connection	CI_NMS_JNDI	cms/OUCCOUNMSConnectionFactory

## XAI Sender

To create a Realtime XAI sender configured to communicate with the integration layer:

### XAI Sender for Customer Data Synchronization

Create a new XAI Sender for each Oracle Utilities Customer Care and Billing Outbound Integration Queue.

1. Navigate to Admin Menu, XAI Sender.
2. Enter a unique XAI Sender and Description.
3. Populate values as follow:

**XAI Sender** – Sender name in Oracle Utilities Customer Care and Billing

**Description** – Sender description

**Invocation Type** – *Real-time*

**XAI Class** – *RTJMSQSNDR* (Realtime JMS Queue Sender)

**Active** - Select the checkbox

**MSG Encoding** – *UTF-8 message encoding*

**XAI JMS Connection** – XAI JMS connection created for the integration

**XAI JMS Queue** – XAI JMS Queue created for the Oracle Utilities Customer Care and Billing Outbound Queue.

4. Select the Context tab and set values for the following Context Types:

**JMS Message Type (Bytes(Y)/Text(N))** – *N*

**JMS User Name** – user for the SOA server to be accessed

**JMS User Password** – Password for the SOA server to be accessed

For example:

XAI Sender	Description	XAI JMS Connection	XAI JMS Queue
CI_NMS_CDSYN	NMS Customer Data Sync Request Sender	CI_NMS_CF	CI_NMSCDSync

#### **XAI Sender for Trouble Call Interface**

Create a new XAI Sender which points to the Trouble Call Interface EBF endpoint URL for the Trouble Call Interface the integration point.

1. Navigate to Admin Menu, XAI Sender.
2. Enter a unique XAI Sender and Description.
3. Populate values as follow:

Invocation Type = *Real-time*

XAI Class = *RTHTTPSNDR*. (Real Time Sender to route messages via HTTP)

4. Select the **Active** check box.

MSG Encoding = *UTF-8 message encoding*

5. 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 Trouble Call Interface

EBF.

**For example:**

`http://demoenv/soa-infra/services/CCB-NMS/OUCCBOUNMSTroubleCallInterfaceEBF/ouccbounmstroublecallinterfaceebf_client_ep`

### XAI Sender for Job History Query

To create an HTTP sender configured to communicate with the integration layer:

Create a new XAI Sender which points to the Job History EBF endpoint URL for Job History the integration point.

1. Navigate to Admin Menu, XAI Sender.
2. Enter a unique XAI Sender and Description.
3. Populate values as follow:

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 Query Job History EBF.

**For example:**

`http://demoenv/soa-infra/services/CCB-NMS/OUCCBOUNMSJobHistoryQueryEBF/ouccbounmsjobhistoryquery_client_ep`

### XAI Sender for Trouble Call History Query

To create an HTTP sender configured to communicate with the integration layer:

Create a new XAI Sender which points to the Trouble Call History EBF endpoint URL for Trouble Call History the integration point.

1. Navigate to Admin Menu, XAI Sender.
2. Enter a unique XAI Sender and Description.
3. Populate values as follow:

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 Query Trouble Call History EBF.

**For example:**

http://demoenv/soa-infra/services/CCB-NMS/OUCCBOUNMSTroubleCallsQueryEBF/OU  
CCBOUNMSTroubleCallsQueryEBF\_ep

### XAI Sender for Planned Outages Query

To create an HTTP sender configured to communicate with the integration layer:

Create a new XAI Sender which points to the Planned Outages Query EBF endpoint URL for Planned Outages Query the integration point.

1. Navigate to Admin Menu, XAI Sender.

2. Enter a unique XAI Sender and Description.

3. Populate values as follow:

Invocation Type = **Real-time**

XAI Class = **RTHTTPSNDP**. (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 Query Planned Outage EBF.

**For example:**

http://demoenv/soa-infra/services/CCB-NMS/ OUCCBOUNMSOutageHistoryQueryEBF/  
OUCCBOUNMSOutageHistoryQueryEBF\_ep

## Outbound Message Type

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

### Customer Data Synchronization

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique **Outbound Message Type** and **Description**.
3. Populate values as follow:

**Business Object** = **C1-NMSSPSyncReqOutMsg** (Customer Sync Outbound Message BO)

**Priority** = (choose from the selection)



### Trouble Calls Interface

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique **Outbound Message Type** and **Description**.
3. Populate values as follow:

**Business Object** = **C1-OutageCallOutboundMsg** (Outage Call Outbound Message BO)

**Priority** = (choose from the selection)

### Query Job History

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique **Outbound Message Type** and **Description**.
3. Populate values as follow:

**Business Object** = **C1-OutageJobHistory** (Job History Query Outbound Message BO)

**Priority** = (choose from the selection)

### Query Trouble Call History

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique **Outbound Message Type** and **Description**.
3. Populate values as follow:

**Business Object** = **C1-OutageCallHistory** (Call History Query Outbound Message BO)

**Priority** = (choose from the selection)

### Query Planned Outages

1. Navigate to Admin Menu, Outbound Message Type.
2. Enter a unique **Outbound Message Type** and **Description**.
3. Populate values as follow:

**Business Object** = **C1-PlannedOutages** (Planned Outages Query Outbound Message BO)

**Priority** = (choose from the selection)

## External System

Create a new External System for direct the integration.

1. Navigate to Admin Menu, External System.

2. Enter a unique **External System** and **Description**.
3. Set **Our Name in Their System** to **CC&B**
4. Define the Outbound Message Types associated to the integration.
  - For Customer Data Synchronization Outbound Message Type
    - Populate values as follow:  
**Outbound Message Type** = (Outbound Message Type for Customer Data Synchronization)  
**Processing Method** = *Real-time*  
**XAI Sender** = (XAI Sender for Customer Data Synchronization)  
**Message XSL** = *CDxAddEnvelope-SOAP1-2.xsl*  
**Response XSL** =
  - For Trouble Calls Interface Outbound Message Type
    - Populate values as follow:  
**Outbound Message Type** = (Outbound Message Type for Trouble Call Interface)  
**Processing Method** = *Real-time*  
**XAI Sender** = (XAI Sender for Trouble Call Interface)  
**Message XSL** = *CDxAddEnvelope-SOAP1-2.xsl*  
**Response XSL** =
  - For Query Job History Outbound Message Type
    - Populate values as follow:  
**Outbound Message Type** = (Outbound Message Type for Job History Query)  
**Processing Method** = *Real-time*  
**XAI Sender** = (XAI Sender for Job History Query)  
**Message XSL** = *CDxAddEnvelope-SOAP1-2.xsl*  
**Response XSL** = *C1-NMSRemoveEnvelopeJobHist.xsl*
  - For Query Trouble Call History Outbound Message Type
    - Populate values as follow:  
**Outbound Message Type** = (Outbound Message Type for Trouble Call History Query)  
**Processing Method** = *Real-time*  
**XAI Sender** = (XAI Sender for Trouble Call History Query)  
**Message XSL** = *CDxAddEnvelope-SOAP1-2.xsl*  
**Response XSL** = *C1-NMSRemoveEnvelopeCallHist.xsl*
  - For Query Planned Outages Outbound Message Type

- Populate values as follow:

**Outbound Message Type** = (Outbound Message Type for Planned Outages Query)

**Processing Method** = *Real-time*

**XAI Sender** = (XAI Sender for Planned Outages Query)

**Message XSL** = *CDxAddEnvelope-SOAP1-2.xsl*

**Response XSL** = *C1-NMSRemoveEnvelopePlanOut.xsl*

---

## Setting Up Oracle Utilities Network Management System

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

There are two components of the Oracle Utilities Network Management System which are involved in the integration:

- **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 Standard Edition.
- **Generic IVR Adapter** - processes trouble calls received from Oracle Utilities Customer Care and Billing. It is part of Oracle Utilities Outage Management Standard Edition.

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 Care and Billing system.

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

### 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 Care and Billing). 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.

## Setting up the Integration Pack

The following sections describe how to configure the integration pack to meet the requirements for the integration.

To configure Oracle Integration Pack for Oracle Utilities CCB - NMS Direct Integration you must complete the following configurations:

- CCB-NMS Configuration Properties
- System Properties
- Domain Value Maps
- Error Handling

### Setting CCB-NMS Configuration Properties

Various configurations that apply to the entire Oracle Utilities CCB - NMS Integration and specific processes for the integration services are stored in the OUCCBOUNMSConfigurationProperties.xml file located under the apps/CCB-NMS/AIAMetaData/config directory.

This section lists the configurations in this file that are used by the Oracle Utilities CCB - NMS Integration. These configurations hold several configurable values that are picked up by the integration at runtime to:

- Set Default values to be used in the integration
- Activate custom implemented extension points available inside the processes. By default these properties are set to false, not to invoke any of the extension points.
- Activate error handling

Whenever the OUCCBOUNMSConfigurationProperties.xml file is updated, the file must be reloaded for updates to be reflected in the applications or services that use the updated properties. You can perform the reload by restarting the soa server.

### Settings for System Properties

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

- Module Configurations are the properties that are shared by multiple the integration flows within this Oracle Integration Pack for Oracle Utilities CCB - NMS Direct Integration.
- Service Configurations are the properties that are used by a specific the integration process.

#### Module Configurations

Module Name	Property Name	Default / Shipped Value	Description
CCB-NMS	CCB.Generic.MessageCategory	11114	This is the Message category that the integration uses

Module Name	Property Name	Default / Shipped Value	Description
			for CCB Error messages
CCB-NMS	CCB.GenericBusinessException.MessageNumber	11001	This is the Message number the integration uses for Generic CCB error
CCB-NMS	CCB.GenericDVMException.MessageNumber	11401	This is the Message number the integration uses for DVM error.
CCB-NMS	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 this is set to OFF the Error Handling will not use the composite and component instance ids to log the error message.
CCB-NMS	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.
CCB-NMS	DVM.OUCCB_OUNMS_ETR_SOURCE.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM OUCCB_OUNMS_ETR_SOURCE.</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_Job_History_Status.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM OUCCB_OUNMS_Job_History_Status.</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_Alarm_State.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM OUCCB_OUNMS_Alarm_State.</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_CALL_STATUS.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM OUCCB_OUNMS_CALL_STATUS.</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p>

Module Name	Property Name	Default / Shipped Value	Description
			If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application
CCB-NMS	DVM.OUCCB_OUNMS_CallBackIndicator.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM OUCCB_OUNMS_CallBackIndicator.</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_MeetType.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_NewCallIndicator.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_CallCancelIndicator.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_AccountType.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_MeterType.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception</p>

Module Name	Property Name	Default / Shipped Value	Description
			<p>error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_Serv_C_Priority.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_Serv_D_Priority.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_Serv_K_Priority.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_LifeSupportIndicator.ThrowException	false	<p>This flag will indicate if an error is triggered when the DVM lookup value is not found for DVM</p> <p>If set to true, the integration triggers a DVM exception error to the initiating application</p> <p>If set to false, the integration does not trigger any exception and instead passes the source application value as the default value to the target application</p>
CCB-NMS	DVM.OUCCB_OUNMS_ErrorCode.ThrowException	false	This flag is not used by this the integration.
CCB-NMS	DVM.OUCCB_OUNMS_LifeSupportIndicator_FALSE_VALUE	0	This is the NMS value for life support if it is false.
CCB-NMS	DVM.OUCCB_OUNMS_LifeSupportIndicator_TRUE_VALUE	1	This is the NMS value for life support if it is true. This value is used to check if the CCB life support value in the premise info or person info is true then the NMS value has to be set to true.

### Service Configurations

Service Name	Property Name	Default / Shipped Value	Description
OUCCBOUNMSCustomerDataSyncReqEBF	Default.SystemID	OU_CCB_01	Initiating system ID.
	Extension.PreXformCCBtoNMS	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformCCBtoNMS	false	If set to true the Post transformation extension service will be invoked.
	Extension.CustomXformItemCCBtoNMS	false	If set to true the Custom transformation for Item will be invoked.
	BusinessError.NotificationFlag	true	If set to true Business error notification is sent via email.
	TechnicalError.NotificationFlag	true	If set to true Technical error notification is sent via email.
	Comments.NoItemInfoMapping	Synchronization of item information is not supported.	These comments are used to store a clearer error message in the integration error table when an error is encountered when a message with item information is sent from CCB but no customization is enabled in the integration layer.
	Comments.NoMeterandItemMapping	Meter or item information is missing.	These comments are used to store a clearer error message in the integration error table when an error is encountered when a message with no meter or item information is sent from CCB but no customization is enabled in the integration layer.
OUCCBOUNMSTroubleCallInterfaceEBF	Default.SystemID	OU_CCB_01	Initiating system ID.
	Extension.PreXformCCBtoNMS	False	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformCCBtoNMS	False	If set to true the Post transformation extension service will be invoked.
	NMS.CCBSourceID	2	Define unique call source ID for CCB - This CCB call source ID will indicate that the trouble call was created in CCB. - This will also ensure that the external ID in the trouble calls table will have a unique value. NMS will prefix the call source ID to the external ID coming from CCB.



Service Name	Property Name	Default / Shipped Value	Description
	CCB.LocationType.StreetIntersection	C1IS	CCB use the location type to determine whether a street intersection or street segment was populated for the location of a fuzzy call.  This is the CCB Location Type value for Street Segment.  Location Type coming from CCB is not mapped to NMS.  It is only used by the integration for the mapping of the cross street or block number.
	CCB.LocationType.StreetSegment	C1SS	CCB use the location type to determine whether a street intersection or street segment was populated for the location of a fuzzy call.  This is the CCB Location Type value for Street Intersection.  Location Type coming from CCB is not mapped to NMS.  It is only used by the integration for the mapping of the cross street or block number.
	BusinessError.Notification Flag	false	If set to true Business error notification is sent via email.
	TechnicalError.Notification Flag	false	If set to true Technical error notification is sent via email.
OUCCBOUNMSJobHistory Query	Default.SystemID	OU_CCB_01	Initiating system ID.
	Extension.PreXformCCBtoNMS	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformCCBtoNMS	false	If set to true the Post transformation extension service will be invoked.
	Extension.PreXformNMStoCCB	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformNMStoCCB	false	If set to true the Post transformation extension service will be invoked.
	NMS.NumberOfDaysOfHistory	60	Define the Number of Days Of Job History to return
OUCCBOUNMSTroubleCallsQueryEBF	Default.SystemID	OU_CCB_01	Initiating system ID.

Service Name	Property Name	Default / Shipped Value	Description
	Extension.PreXformCCBtoNMS	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformCCBtoNMS	false	If set to true the Post transformation extension service will be invoked.
	Extension.PreXformNMStoCCB	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformNMStoCCB	false	If set to true the Post transformation extension service will be invoked.
	NMS.NumberOfDaysOfHistory	60	Define the Number of Days Of Call History to return
OUCCBOUNMSPlannedOutagesQueryEBF	Default.SystemID	OU_CCB_01	Initiating system ID.
	Extension.PreXformCCBtoNMS	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformCCBtoNMS	false	If set to true the Post transformation extension service will be invoked.
	Extension.PreXformNMStoCCB	false	If set to true the Pre transformation extension service will be invoked.
	Extension.PostXformNMStoCCB	false	If set to true the Post transformation extension service will be invoked.
	NMS.NumberOfDaysOfHistory	10	Define the Number of Days Of Planned Outages to return

## 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: “FOOT” and “FT” or “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 the information within the domain value maps:

1. Open a browser and access the SOA Composer Application
2. On the SOA composer click on the “Open” dropdown and select “Open DVM”. This will display a list of all the DVM files in MDS.
3. Search and Select the relevant DVM you want to maintain.
4. The Edit button in the top navigation bar enables editing the DVM.

5. Once the DVM has been edited click on the “Save” button in the top navigation bar. This will save the DVM data for that session.
6. Click on “Commit” after updating each DVM this will save the DVM data in MDS.

These are the DVMs for the Oracle Utilities Customer Care and Billing to Oracle Utilities Network Management System Integration:

DVM	Integration Points	Description
OUCCB_OUNMS_AccountType	Customer Data Synchronization	DVM mapping for CCB Service Type Code and NMS Account Type Code
OUCCB_OUNMS_Alarm_State	Job History Query	DVM mapping for Alarm State Code
OUCCB_OUNMS_Call_Status	Trouble Call History Query	DVM mapping for Call Status Code
OUCCB_OUNMS_CallBackIndicator	Trouble Call Interface	DVM mapping for Call Back Indicator
OUCCB_OUNMS_CallCancelIndicator	Trouble Call Interface	DVM mapping for Call Cancel Indicator
OUCCB_OUNMS_Cause	Job History Query	DVM mapping for Cause Code
OUCCB_OUNMS_ETR_Source	Job History Query	DVM mapping for ETR Source Code
OUCCB_OUNMS_Job_History_Status	Job History Query	DVM mapping for Job History Status Code
OUCCB_OUNMS_LifeSupportIndicator	Customer Data Synchronization	DVM mapping for Life Support Indicator
OUCCB_OUNMS_MeetType	Trouble Call Interface	DVM mapping for Meet Type Code
OUCCB_OUNMS_MeterType	Customer Data Synchronization	DVM mapping for Meter Type Code
OUCCB_OUNMS_NewCallIndicator	Trouble Call Interface	DVM mapping for New Call Indicator
OUCCB_OUNMS_Serv_C_Priority	Customer Data Synchronization	DVM mapping for Critical Priority Code
OUCCB_OUNMS_Serv_D_Priority	Customer Data Synchronization	DVM mapping for Medical Priority Code
OUCCB_OUNMS_Serv_K_Priority	Customer Data Synchronization	DVM mapping for Key Priority Code

For more information on matching values for each DVM, refer to the applicable section in [Setting up Oracle Utilities Customer Care and Billing](#) for OUCCB values and [Setting up Oracle Utilities Network Management System](#) for OUNMS values.

#### **OUCCB\_OUNMS\_AccountType**

This DVM is used to map the CCB Service Type to NMS Account Type.

OUCCB_ServiceType	OUNMS_AccountType
This contains the valid CCB values for ‘Service Type’.	This contains the valid NMS values for ‘Account Type’.

#### **OUCCB\_OUNMS\_Alarm\_State**

<b>OUCCB_Alarm_State</b>	<b>OUNMS_Alarm_State</b>
This contains the valid CCB values for 'Alarm State'. These values are obtained from CCB lookup C1_ALARM_STATE_FLG.	This contains the valid NMS values for 'Alarm State'.

#### **OUCCB\_OUNMS\_Call\_Status**

<b>OUCCB_Call_Status</b>	<b>OUNMS_Call_Status</b>
This contains the valid CCB values for 'Call Status'. These values are obtained from CCB lookup C1_CALL_STATUS_FLG.	This contains the valid NMS values for 'Call Status'.

#### **OUCCB\_OUNMS\_CallbackIndicator**

<b>OUCCB_CallbackRequested</b>	<b>OUNMS_CallbackFlag</b>
This contains the valid CCB values for 'Callback Requested'.	This contains the valid NMS values for 'Callback Flag'.

#### **OUCCB\_OUNMS\_CallCancelIndicator**

<b>OUCCB_Status</b>	<b>OUNMS_CallCancelFlag</b>
This contains the valid CCB values for 'Call Cancel'.	This contains the valid NMS values for 'Call Cancel Flag'.

#### **OUCCB\_OUNMS\_ETR\_Source**

<b>OUCCB_ETR_Source</b>	<b>OUNMS_ETR_Source</b>
This contains the valid CCB values for 'ETR Source'. These values are obtained from CCB lookup C1_ETR_SOURCE_FLG	This contains the valid NMS values for 'ETR Source'.

#### **OUCCB\_OUNMS\_Job\_History\_Status**

<b>OUCCB_Job_History_Status</b>	<b>OUNMS_Job_History_Status</b>
This contains the valid CCB values for 'ETR Source'. These values are obtained from CCB lookup C1_ETR_SOURCE_FLG	This contains the valid NMS values for 'Job History Status'.

#### **OUCCB\_OUNMS\_LifeSupportIndicator**

<b>OUCCB_LifeSupportIndicator</b>	<b>OUNMS_LifeSupportIndicator</b>
This contains the valid CCB values for 'Life Support Indicator'.	This contains the valid NMS values for 'Life Support Indicator'.

#### **OUCCB\_OUNMS\_MeetType**

<b>OUCCB_MeetType</b>	<b>OUNMS_MeetType</b>
This contains the valid CCB values for 'Meet Type'.	This contains the valid NMS values for 'Meet Type'.

#### OUCCB\_OUNMS\_MeterType

OUCCB_MeterType	OUNMS_MeterType
This contains the valid CCB values for 'Meter Type'.	This contains the valid NMS values for 'Meter Type'.

#### OUCCB\_OUNMS\_NewCallIndicator

OUCCB_OutageCallAction	OUNMS_UpdateExistingFlag
This contains the valid CCB values for 'Outage Call Action Flag'. These values are obtained from CCB lookup C1_OUTCALL_ACT_FLG	This contains the valid NMS values for 'Update Existing Flag'.

#### OUCCB\_OUNMS\_Serv\_C\_Priority

OUCCB_Serv_C_Priority	OUNMS_Serv_C_Priority
This contains the valid CCB values for 'Serv C Priority'.	This contains the valid NMS values for 'Serv C Priority'

#### OUCCB\_OUNMS\_Serv\_D\_Priority

OUCCB_Serv_D_Priority	OUNMS_Serv_D_Priority
This contains the valid CCB values for 'Serv D Priority'.	This contains the valid NMS values for 'Serv D Priority'

#### OUCCB\_OUNMS\_Serv\_K\_Priority

OUCCB_Serv_K_Priority	OUNMS_Serv_K_Priority
This contains the valid CCB values for 'Serv K Priority'.	This contains the valid NMS values for 'Serv K Priority'

### Error Notification Setup

To enable EMAIL notification for CCB-NMS Error Handling module.

1. Log in to the enterprise manager console.
2. Click on > Expand SOA → right click on soa-infra → SOA Administration → Workflow Notification Properties
3. On this screen select EMAIL from the drop down.
4. Provide the email IDs for From address.

### INTEGRATION\_ERR\_LOOKUP

Error handling module configuration is governed by the Integration\_err\_lookup table. This table contains processing instructions for each composite. The Error\_Processing\_Parent composite picks data for one composite and calls error\_Processing\_Detail for processing.

S. No.	Column Name	Description	Default / Suggested values
1	LookUp_ID	Sequence ID of entry in this table. This is auto generated.	Auto generated

S. No.	Column Name	Description	Default / Suggested values
2	IP_Name	The name of the composite that is processed.  Example OUCCBOUNMSCustomerSyncEBF	This column is pre populated with the individual enterprise business flow name.  Do not modify: Modifying this value will break the code.
3	Processing_Status	Current status of processing it has to be one of the following.  HALTED (waiting for manual intervention),  NOT REQUIRED  ALIVE	NOT REQUIRED
4	Run_Flag	Processing flag status, Y or N. Unread value = N, read value =Y	N
5	Next_Runtime	Next runtime when the error record should be processed for this composite.	SYSDATE+200
6	Halt_For_Error	Allowed values Y or N.  When set to Y manual intervention is required after one successful error record processing.  When set to N processing continues without halting.	N
7	RunTime_Interval	Runtime in minutes after which the next error processing should be done.  Example : P10Y0M0DT0H0M0S  Next processing is done after 10 years 0 months 0 days 0 hours 0 minutes and 0 seconds  This value must be updated based on the business requirement. Setting fewer intervals may have impact on performance.	Default : P10Y0M0DT0H0M0S
8	Email_ID	Email-ID where error notifications are sent.  This value can be different or same for all the composites.	Default : email@email.com

S. No.	Column Name	Description	Default / Suggested values
9	Email_Content_Type	<p>GENERIC – One email is sent for all errors. No detail information is included.</p> <p>SINGLE – One email is sent for all errors with all details included in the attachment.</p> <p>MULTIPLE – Multiple emails are sent &amp; each email has information equal to the value specified in the Error_Count_Per_Notification column.</p> <p>Values are case sensitive and must always be given in upper case.</p>	Default : GENERIC
10	Email_XSL	<p>XSL to be applied for creating email Content which includes subject/body and attachment. Look and feel can be modified here.</p>	<p>Default file is provided for all the composites and present under the xsl folder of composite.</p> <p>Example: xsl/Transformation_Create_Email.xsl</p> <p>Copy this to the mds folder and enter the mds path in this column for additional configuration.</p>
11	Error_Count_Per_Processing	<p>A notification is sent after the number of records set here is processed.</p> <p>For example: If this is set to 50 then an email notification containing 50 records is sent after 50 records are created in the error store.</p>	Default : 100
12	Email_Attachment_Location	<p>Location where the email attachment is created on the server.</p> <p>This value should point to the location / folder where the attachment should be stored.</p> <p>This is used to create the attachment file in the following format.</p> <p>INTEGRATION_ERR_LOOKUP.Email_Attachment_Location + IP_Name + Date (in YYYYMMDDHH24MMSS)</p>	

S. No.	Column Name	Description	Default / Suggested values
13	Email_Attachment_Flag	<p>Y – Send email with attachment. In this case it is not mandatory to have Email_Attachment_Location specified.</p> <p>N – Send email without attachment but send the attachment location. In this case Email_Attachment_Location has to be specified.</p> <p>ServerName</p> <p>+INTEGRATION_ERR_LOOKUP.Email_Attachment_Location + IntegrationPoint_Name + Date in YYYYMMDDHH24MMSS</p>	N
14	Publish_Human_Task_Flag	<p>Y – Publish human task</p> <p>N – Don't publish human task</p> <p>If Halt_For_Error value is set to Y and Publish_Human_Task_Flag is also Y then human task will be published and the user can take action from worklist application.</p>	N
15	ID_Human_Task	<p>User/Group ID to which human task should be published in case Halt_For_Error is set to Y</p> <p>This ID must be present in the weblogic realm pointed by fusion middleware.</p>	weblogic
16	Last_Updated_Date	Last updated date time	SYSDATE
17	Purge_Error_Store_Flag	<p>Y – Purge data</p> <p>N – No purge require</p> <p>The process PurgeIntegrationErrorStore will only be deployed when following flag purge.process.deploy=true in deploy.properties file is set to true during installation.</p> <p>If flag.purge.process = false then value of this column Purge_Error_Store_Flag will always be N.</p>	Default : N



S. No.	Column Name	Description	Default / Suggested values
18	Purge_Processing_Status_Flag	<p>Y – Purge Processing in process N – Purge processing not happening</p> <p>The process PurgeIntegrationErrorStore will only be deployed when following flag purge.process.deploy=true in deploy.properties file is set to true during installation.</p> <p>If flag.purge.process = false then value of this column Purge_Error_Store_Flag will always be N.</p>	Default : N
19	Purge_Frequency	<p>No of days after which data should be purged. This will be in picture format</p> <p>Example : P10Y0M0DT0H0M0S</p> <p>Next processing will be done after 10 years 0 months 0 days 0 hours 0 minutes and 0 seconds</p> <p>This value has to be updated based on the business requirement. Setting fewer intervals may have impact on performance.</p> <p>Need to set this value appropriately.</p> <p>Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationErrorStore ID deployed.</p>	Default : P10Y0M0DT0H0M0S
20	Next_Purge_Date	<p>Next purge date. Format: Next_Purge_date + Purge_Frequency</p> <p>Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationErrorStore ID deployed.</p>	SYSDATE+100

S. No.	Column Name	Description	Default / Suggested values
21	Purge_File_Name	Directory Name where purge file should be stored.  Applicable only when flag.purge.process = true in deploy.properties file during installation and the process PurgeIntegrationErrorStore ID deployed.	'location on server where purge record should be persisted'

Primary Key - LookUp \_ID, IP\_Name

To customize error email notifications for individual the integration points:

1. Use the composite: UpdateIntegrationErrorLookupTable .
2. Enter the following URL into a browser to get the screen that will provide options of updating the contents of the table.  
http://<hostname>:<soa server port>/soa-infra/services/CCB-NMS/  
UpdateIntegrationErrorLookupTable/updates the integrationerrorlookuptablebpel\_client\_ep?
3. Expand WS-Security and provide authentication information.  
This username and password is going to be same as that used for login to weblogic enterprise manager console.
4. Expand the payload section.  
This displays several editable text fields.  
Only the ipName field is mandatory and should be entered as one of the values from  
INTEGRATION\_ERR\_LOOKUP.IP\_NAME field.  
  
By default all the checkboxes appearing next to the text fields are checked.
5. Provide values in the text field.  
If you do not want to have a particular value updated then uncheck the box.

## Chapter 4: Error Handling, Monitoring and Troubleshooting

This chapter provides detail into monitoring, error handling, and troubleshooting, and discusses how to:

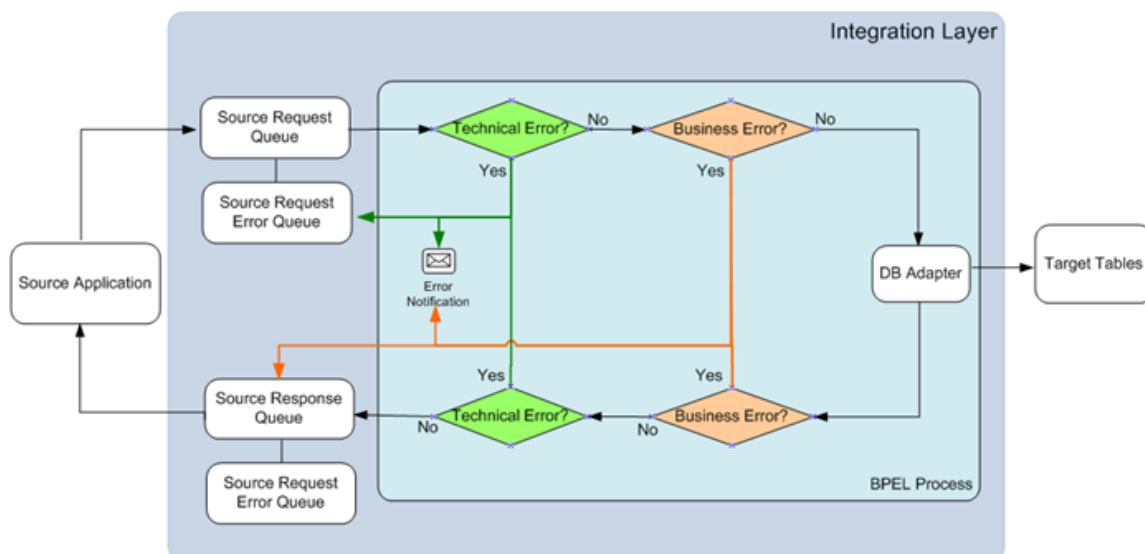
- Monitor from Oracle Utilities Customer Care and Billing
- Monitor from Oracle Utilities Network Management System
- Monitor from the Integration Layer
- Manage Failure Scenarios

---

### Error Handling by Integration Point

---

#### Customer Synchronization



Customer Data Synchronization – Error Handling

There are two types of error that can be encountered in the integration:

- **Business Errors** are triggered when the DVM exception flag is set to true and there is a DVM conversion error or business errors are triggered by the application. Business errors are sent back to the source application and can be re-tried from there.

- **Technical Errors** are triggered when there are connectivity issues (database is down, queue is not reachable). Technical errors are sent to the error queue and can be re-tried from the integration layer.

S. No	Type of error	Action	Notification Type	Retry
1	CCB cannot reach the CCB Customer Data Sync Request Queue	Sync error response to CCB	CCB will create a To Do.	User after ensuring that the source of the issue had been fixed can resend the message from CCB.
2	Internal failure in BPEL Process (Technical Error)	Message will go to the request error queue	Email (optional)	From Weblogic Admin Console, move message to the CCB request queue
3	BPEL cannot reach NMS DB (Technical Error)	Message will go to the request error queue	Email (optional)	From Weblogic Admin Console, move message to the CCB request queue
4	Error response from NMS DB (Business Error)	Insert a negative acknowledgement message to the CCB response queue	Email (optional) and CCB To Do.	User after ensuring that the source of the issue had been fixed can resend the message from CCB.
5	Empty DVM values in incoming message or the DVM lookup values are not found (Business Error)	Insert a negative acknowledgement message to the CCB response queue	- Email (optional) and CCB To Do.	User after ensuring that the source of the issue had been fixed can resend the message from CCB.
6	Any other Technical Error	Message will go to the request error queue	Email (optional)	From Weblogic Admin Console, move message to the CCB request queue

A create To Do entry algorithm must be set up in the error status of the Sync Request BO for CCB to create a To Do entry on error.

To retry the technical error failure messages:

1. Open a browser and access the Weblogic console for your installation
2. Navigate to Services-> Messaging -> JMS Modules.
3. Select the CCB-NMS Integration JMS Module and it will show all the queues related to this the integration.
4. Select the appropriate Error queue and click on the Monitoring tab.

5. This tab will show the details about the messages in the queue in a table.
6. Select the checkbox in the details table and click on the Show Messages button. This will display all the messages in the Error queue.
7. Click on the Move button and select the Move All option.
8. Select the CCB-NMS JMS Server to move the messages and click on the Next button.
9. Now select the correct parent queue for the error queue from the dropdown and click on the Finish button.
10. This action will move all the messages to the source queue and the integration will process all the messages again.

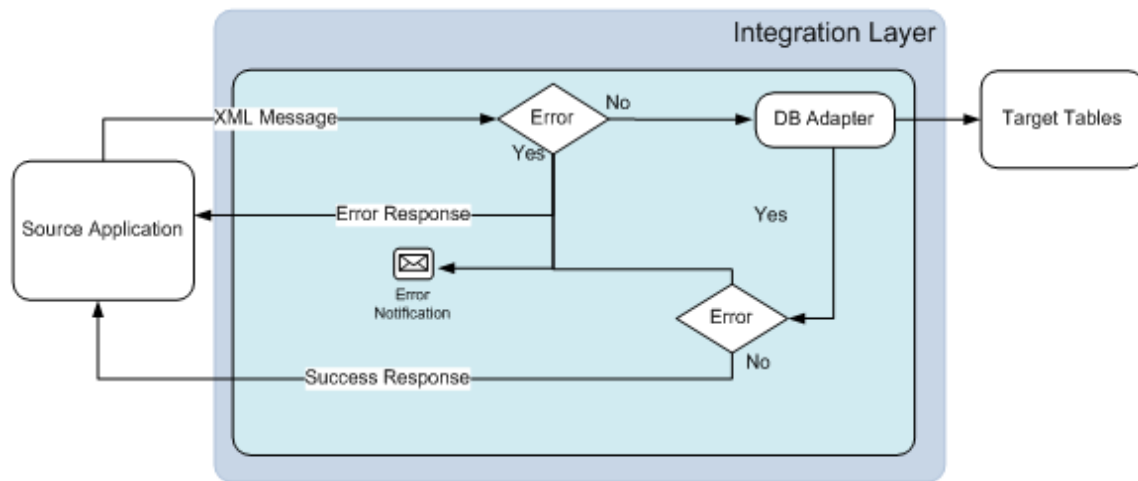
When resending from Oracle Utilities Customer Care and Billing, the user can either:

Change the status of the existing sync request in error from Error to Pending Sync Request or

Change data and create a new sync request but also change the status of the existing sync request in error from Error to Discarded.

---

## Trouble Call Interface

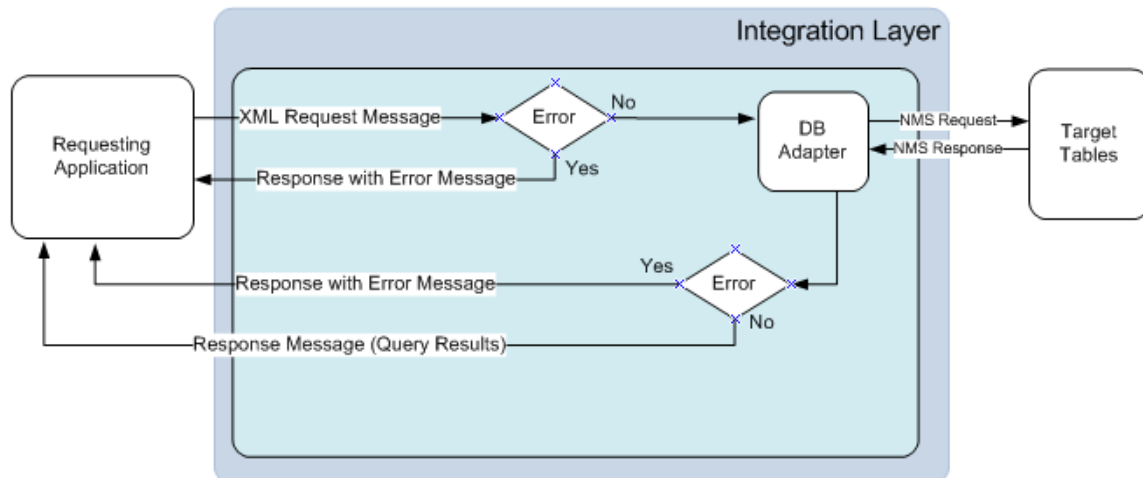


Trouble Calls Interface – Error Handling

Any errors encountered in the integration layer respond with an error to the Oracle Utilities Customer Care and Billing which will then show an outbound message in 'Error' status. Oracle Utilities Customer Care and Billing re-sends the message when the error is fixed.

S. No	Type of error	Action	Notification Type	Retry
A1	CCB cannot reach BPEL (Technical Error)	Sync error response to CCB	- CCB Display Error in the Outage Call UI	User can resend the message from the Outage Call UI after ensuring that the source of the issue had been fixed.
A2	Internal failure in BPEL (Technical Error)	Sync error response to CCB	- CCB Display Error in the Outage Call UI - Email (optional)	
A3	BPEL cannot reach NMS database (Technical Error)	Sync error response to CCB	- CCB Display Error in the Outage Call UI - Email (optional)	
A4	Error response from NMS (Business Error)	Sync error response to CCB	- CCB Display Error in the Outage Call UI - Email (optional)	
A5	DVM exception (Business Error)	Sync error response to CCB	- CCB Display Error in the Outage Call UI - Email (optional)	

## Online Queries (Job History / Call History / Planned Outages)



Any errors encountered in the integration layer reported back to the Oracle Utilities Customer Care and Billing synchronously and an error message is displayed in the UI to inform the user that an error was encountered by the external system. No error notification is needed.

## Error Notification Setup

To enable EMAIL notification for CCB-NMS Error Handling module.

1. Log in to the enterprise manager console.
2. Click on 1> Expand SOA → right click on soa-infra → SOA Administration → Workflow Notification Properties
3. On this screen select EMAIL from the drop down.
4. Provide the email ids for from address / Actionable address and Reply To address.

## Error Handling Module Configuration

Error Handling module configuration are governed by the integration\_err\_lookup table following is description of this table and the possible values and how it affects the Error Handling.

### INTEGRATION\_ERR\_LOOKUP

This table will contain processing instruction for each composite & Error\_Processing\_Parent composite will pick data for one composite & call the error\_Processing\_Detail for processing

S. No.	Column Name	Description	Default / Suggested values
1	LookUp_ID	Sequence ID of entry in this table. This is auto generated.	Auto generated
2	IP_Name	Composite name for which processing should be done, Example OUCCBOUNMSCustomerSyncEBF Note: This is case sensitive and modifying these values will result in breaking the code.	This column will be pre populated with the individual enterprise business flow name.
3	Processing_Status	Current status of processing it has to be one of the following. HALTED (waiting for manual intervention), NOT REQUIRED ALIVE	NOT REQUIRED
4	Run_Flag	Processing flag status, Y or N. Unread value will be N while read be Y	N
5	Next_Runtime	Next runtime when the error record should be processed for this composite.	SYSDATE+200
6	Halt_For_Error	Allowed values Y or N.	N

S. No.	Column Name	Description	Default / Suggested values
		<p>When set to Y - Manual intervention is required after one successful error record processing.</p> <p>When set to N – processing continues without halting.</p>	
7	RunTime_Interval	<p>Runtime in minutes after which the next error processing should be done. This will be picture format.</p> <p>Example : P10Y0M0DT0H0M0S</p> <p>Next processing will be done after 10 years 0 months 0 days 0 hours 0 minutes and 0 seconds</p> <p>This value has to be updated based on the business requirement. Setting fewer intervals may have impact on performance.</p> <p>Need to set this value appropriately.</p>	Default : P10Y0M0DT0H0M0S
8	Email_ID	<p>Email-ID to which error notification should be sent for this the integration point.</p> <p>This value can be different or same for all the composites.</p>	Default : email@email.com
9	Email_Content_Type	<p>GENERIC – One email be send for all error without detail information</p> <p>SINGLE – One email be send for all error with all information dump into attachment</p> <p>MULTIPLE – Multiple emails will be sent &amp; each email will have information equal to the value specified in Error_Count_Per_Notification column.</p> <p>Values are case sensitive and must always be given in upper case.</p> <p>Example: GENERIC, SINGLE,MULTIPLE</p>	Default : GENERIC
10	Email_XSL	<p>XSL to be applied for creating email subject &amp; e-mail body. This will be stored in MDS so that customer can update its look &amp; feel.</p>	<p>Default file is provided for all the composites and present under xsl folder of composite.</p> <p>Example: xsl/Transformation_Create_Email.xsl</p> <p>If the client wants then can put it under the mds and provide mds path in this column.</p>



S. No.	Column Name	Description	Default / Suggested values
11	Error_Count_Per_Processing	No of records for which a notification to be sent.  Example: If this is 50 then email notification will be sent after every 50 records for this composite are created in error store and each email will contain 50 records.	Default : 100
12	Email_Attachment_Location	Location where the email attachment be created on server hosting SOA suite.  This value will point to the location / folder where attachment needs to be stored.  This will be use to create the attachment file in following format.  INTEGRATION_ERR_LOOKUP.Email_Attachment_Location + IP_Name + Date (in YYYYMMDDHH24MMSS)	'location on server where email attachment be persisted'
13	Email_Attachment_Flag	Y – Send email with attachment. In this case it is not mandatory to have Email_Attachment_Location specified.  N – Send email without attachment but send the attachment location which will be. In this case Email_Attachment_Location has to be specified.  ServerName  +INTEGRATION_ERR_LOOKUP.Email_Attachment_Location + IntegrationPoint_Name + Date in YYYYMMDDHH24MMSS	N
14	Publish_Human_Task_Flag	Y – Publish human task  N – Don't publish human task  If Halt_For_Error value is set to Y and Publish_Human_Task_Flag is also Y then human task will be published and user can take action from worklist application.	N
15	ID_Human_Task	User/Group ID to which human task should be published in case Halt_For_Error is set to Y	weblogic
16	Last_Updated_Date	Last updated date time	SYSDATE
17	Purge_Error_Store_Flag	Y – Purge data  N – No purge require	Default : N
18	Purge_Processing_Status_Flag	Y – Purge Processing in process  N – Purge processing not happening	Default : N
19	Purge_Frequency	No of days after which data should be	Default : P10Y0M0DT0H0M0S

S. No.	Column Name	Description	Default / Suggested values
		<p>purged. This will be in picture format</p> <p>Example : P10Y0M0DT0H0M0S</p> <p>Next processing will be done after</p> <p>10 years 0 months 0 days 0 hours 0 minutes and 0 seconds</p> <p>This value has to be updated based on the business requirement. Setting fewer intervals may have impact on performance.</p> <p>Need to set this value appropriately.</p>	
20	Next_Purge_Date	Next purge date. It will be populated based on Next_Purge_date + Purge_Frequency	SYSDATE+100
21	Purge_File_Name	Directory Name where purge file should be stored.	'location on server where purge record should be persisted'

Primary Key: LookUp \_ID, IP\_Name

## Monitoring from Oracle Utilities Customer Care and Billing

### Oracle Utilities Customer Care and Billing Error Logs

Errors related to the online the integration invocation from CCB are stored in the CCB\_ENVIRONMENT\_NAME/system/logs folder.

For example: V231\_CCB\_PERF\_BLD10\_LIN\_ORA\_WLS/logs/system

Errors related to Batch the integration invocation from CCB are stored in the \$SPLOUTPUT/CCB\_ENVIRONMENT\_NAME folder.

For example: /spl/sploutput/V231\_CCB\_PERF\_BLD10\_LIN\_ORA\_WLS

For more information about errors and notifications see the Oracle Utilities Customer Care and Billing documentation.

---

## Notifications

Any errors encountered during the trouble call entry or online outage info query are displayed to the end-user in real time on the user portal.

Errors in the synchronization process are communicated back to CCB by the integration layer. An XAI Inbound Service invokes processing to transition the sync request to either the Synchronized state (if it is a positive acknowledgement) or the Error state (if it is a negative acknowledgement).

---

## Connection Errors

Information can be found in the log files in the folders described above.

---

## 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 chapter Troubleshooting and Support in the Oracle Utilities Network Management System Configuration Guide.

---

## Monitoring from the Integration

The integration flow can be monitored using the following:

1. Monitoring the composite instances using Weblogic SOA Enterprise Manager
2. Monitoring the Weblogic logs

---

## Weblogic SOA Enterprise Manager

To configure the weblogic SOA enterprise manager

1. Login into the Weblogic SOA server Enterprise Manager
2. In the left menu navigate to SOA → soa-infra → CCB-NMS.
3. All the composite processes deployed for the integration are available under the partition CCB-NMS.
4. Select the appropriate process to list all the instances for the processes sorted by time of execution.

5. The instances also have the request ID as part of the display name.
6. Click on the appropriate process instance and it will display the flow for the process.
7. The composite flow will list all the activities in the process instance.

Note: If the Audit Level is Off, no flow process is shown.

---

## Weblogic Logs

### To configure 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

The Oracle Fusion Middleware maintains state of the instances and the database needs to be purged periodically. Refer to Note 815896.1 on <https://support.oracle.com/> for details about purging data from Fusion Middleware database.

## Chapter 5: Extensibility Options

The Integration Process allows extensibility of transaction messages using the following methods:

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

### Pre Transformation Extension Point

---

The pre transformation extension point is invoked before the main transformation is executed. This transformation will help in transforming the source xml coming as an input to the integration process.

The integration layer defines an external call from the pre-transformation extension point which accepts the source xml as input and gives the source xml as output. The wsdl the integration layer points to an abstract wsdl and can be plugged in by a concrete wsdl by the implementation team.

This helps the implementation to invoke 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 will help in transforming the target xml going as an input to the Target queue.

The Integration layer will define an external call from the Post transformation extension point which accepts the target xml as input and gives the target xml as output. The wsdl the integration layer points to an abstract wsdl and can be plugged in by a concrete wsdl by the implementation team.

This will help the implementation to invoke 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 will 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 will enable the implementation to define and pass additional data from source system to the target system.

## Extension Points

### To implement extension points

1. Each process in the integration has a pre and post transformation extension point which can be used to invoke webservices and transform the payload.
2. The desired extension point can be triggered from the process by enabling it using the OUCCBOUNMSConfigurationProperties.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.
4. These concrete wsdl files are located in MDS under the directories /apps/CCB-NMS/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCCB and /apps/CCB-NMS/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUNMS
5. The binding, port type and soap address for the extension service can be added in the concrete wsdl in the installation folder and the wsdl moved to MDS.
6. To move the wsdl to MDS update the appropriate wsdl in the product install home. The directories in the product install home are CCB\_NMS\_INSTALL\_HOME/MDS-Artifacts/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUCCB and CCB\_NMS\_INSTALL\_HOME/MDS-Artifacts/AIAMetaData/AIAComponents/ExtensionServiceLibrary/OUNMS.
7. The OUCCBOUNMSConfigurationProperties.xml and the concrete wsdl need to be checked-in into MDS and the soa server has to be restarted.
8. Refer to command for MDS update in the installation guide to update MDS.
9. Restart the SOA server. On restart, the extension point will invoke the web service in the concrete wsdl.

For example: To enable the extension points for OUCCBOUNMSCustomerSyncReqExtension add the binding and service elements to the OUCCBOUNMSCustomerSyncReqExtensionConcrete.wsdl

```
<binding name="OUCCBOUNMSCustomerSyncReqV1ExtensionServiceSOAP11Binding"
        type="ccbext: OUCCBOUNMSCustomerSyncReqV1ExtensionService">
  <soap:binding style="document"
    transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="PreXformCCBtoNMSMsg">
    <soap:operation style="document"
      soapAction="http://xmlns.oracle.com/OUCCBOUNMSCustomerSyncEBF/OUCCBOUNMSCustomerSyncReqExtension/V1/ PreXformCCBtoNMSMsg "/>
    <input>
      <soap:body use="literal" parts="CCBToNMS"/>
    </input>
    <output>
      <soap:body use="literal" parts="CCBToNMS"/>
    </output>
    <fault name="fault">
      <soap:fault name="fault" use="literal"/>
    </fault>
  </operation>
</binding>
```

```
        </fault>
    </operation>
    <operation name=" PostXformCCBtoNMSMsg">
        <soap:operation style="document"
            soapAction="http://xmlns.oracle.com/
OUCCBOUNMSCustomerSyncEBF/OUCCBOUNMSCustomerSyncReqExtension/V1
PostXformCCBtoNMSMsg"/>
        <input>
            <soap:body use="literal" parts=" NMSEnqueue "/>
        </input>
        <output>
            <soap:body use="literal" parts=" NMSEnqueue "/>
        </output>
        <fault name="fault">
            <soap:fault name="fault" use="literal"/>
        </fault>
    </operation>
</binding>
<service name="OUCCBOUNMSCustomerSyncReqV1ExtensionService">
<!-- Port name must match the port name used for the Extension service in
the composite.xml for the process - -->
    <port name="OUCCBOUNMSCustomerSyncReqV1ExtensionService"
        binding="ccbext:
OUCCBOUNMSCustomerSyncReqV1ExtensionServiceSOAP11Binding">
        <soap:address location="<b>endpoint url of the Extension
server">/>
    </port>
</service>
```

---

## Custom Transformations

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 xsd has a corresponding CustomType xsd in which the complexType elements for each customElements tag are defined.
3. Some process which does not expect a response back only uses one xsd files while some process that expects a response uses two xsd files, one for the request message and one for the response message.
4. To pass additional elements in the customElements tag, the corresponding complexType needs to be modified in the customType xsd. Add the additional elements required in the complexType elements i.e. xsd for CCB application.
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.
6. These custom templates are invoked at the location where each customElements tag is present.
7. The Custom xsl can be modified to add transformation for the newly added elements in the custom xsd files.

8. The custom xsd files for the CCB application are located in the product install home under the directory CCB\_NMS\_INSTALL\_HOME/MDS-Artifacts/AIAMetaData/AIAComponents/ApplicationObjectLibrary/OUCCB/V1/schemas. The NMS application does not have custom xsds. NMS has defined user defined field in there stored procedure to be used for extension.
9. The custom xsl files are located in the product install home under the directory CCB\_NMS\_INSTALL\_HOME/services/industry/Utilities/EnterpriseBusinessFlow/<Process Name>/xsl
10. After updating the xsd and xsl files in the product install home, update MDS using the ant scripts and restart the SOA server. Refer to command for MDS update in the installation guide to update MDS.

Example:

To modify the Customer Data Synchronization process to pass email from CCB to NMS.

sendDetails→finalSnapshot→personInfo→customElements→email element in CCB to CUST\_UDF1 element in NMS.

Complete the following changes:

**a. Modify OUCCBCustomerSyncReqCustomType.xsd**

```
<xsd:complexType name="personInfoCustomType">
  <xsd:sequence>
    <xsd:element name="email" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

**b. Modify Transform\_CCB\_to\_NMS\_CustomerSyncMeterInfo\_Custom.xsl**

```
<xsl:template name="InputParameters_customElements">
  <!-- this template is use for Xformation of
  //InputParameters/customElements in Request Message-->
  <CUST_UDF1>
    <xsl:value-of
select="/tnsl:sendDetails/tnsl:syncRequestDetails/tnsl:finalSnapshot/tnsl:pe
rsonInfo/tnsl:customElements/tnsl:email"/>
  </CUST_UDF1>
</xsl:template>
```



## Appendix A: Data Mapping

This section provides mapping details for each the integration point.

### Customer Data Synchronization

---

---

#### Mapping Details

NMS Field	CCB Element Name	Notes
Service Point serv_point_ID	spld	From CCB SP Table (CI_SP)
Device ID device_ID	deviceId	<ul style="list-style-type: none"><li>- Device ID provide must exist in the NMS supply nodes table</li><li>- From CCB, obtain from geo value that matches with Geo Type set up for Device ID in CCB Feature configuration schema constants</li><li>- From CCB SP Geo Table (CI_SP_GEO).</li></ul>
Account Type account_type	serviceType	<ul style="list-style-type: none"><li>- Account Type provide must exist in the Account Type Table in NMS</li><li>- Use <a href="#">DVM OUCCB_OUNMS_AccountType</a></li></ul>
Creation Timestamp creation_time		<ul style="list-style-type: none"><li>- Timestamp for the record's creation. NMS stored proc will put the current date time on for new records inserted.</li><li>- Not mapped by the integration</li></ul>
Active Flag active	Y or N (populated by the integration)	<ul style="list-style-type: none"><li>- Flag identifying currently active records</li><li>Y – For Current Customers</li><li>N – For Inactive customers</li><li>- If the customer ID is not populated in the message, the integration set it to N, otherwise it is always Y.</li></ul>
Customer ID cust_ID	accountId	<ul style="list-style-type: none"><li>- Unique external system generated primary key</li><li>- CCB Account must be linked to a metered service point via active service agreement.</li><li>- From CCB Account Table (CI_ACCT)</li></ul>
Customer Account Number cust_account_number	accountId	<ul style="list-style-type: none"><li>- From CCB Account Table (CI_ACCT)</li></ul>
Customer Name cust_name	name	<ul style="list-style-type: none"><li>- This is the primary person linked to the account</li><li>- If it is a Person, it contains the concatenation of last, first and middle names</li></ul>

NMS Field	CCB Element Name	Notes
		<ul style="list-style-type: none"> <li>- If it is a business, contains the business name</li> <li>- From CCB Person Name Table (CI_PER_NAME)</li> </ul>
Customer Home Phone Area Code cust_home_area_code	First 3 number of homePhone	<ul style="list-style-type: none"> <li>- Get Area Code from CCB homePhone element</li> <li>- homePhone retrieved from CCB depends on the home phone type defined in CCB feature configuration schema constants.</li> <li>- From CCB Person Phone Table (CI_PER_PHONE)</li> </ul>
Customer Home Phone cust_home_phone	From fourth to tenth number of homePhone	<ul style="list-style-type: none"> <li>- Get Phone Number from CCB homePhone element</li> <li>- homePhone retrieved from CCB depends on the home phone type defined in CCB feature configuration schema constants.</li> <li>- From CCB Person Phone Table (CI_PER_PHONE)</li> </ul>
Customer Day Phone Area Code cust_day_area_code	First 3 number of businessPhone	<ul style="list-style-type: none"> <li>- Get Area Code from CCB businessPhone</li> <li>- businessPhone retrieved from CCB depends on the home phone type defined in CCB feature configuration schema constants.</li> <li>- From CCB Person Phone Table (CI_PER_PHONE)</li> </ul>
Customer Day Phone cust_day_phone	From fourth to tenth number of businessPhone	<ul style="list-style-type: none"> <li>- Get Phone Number from CCB businessPhone</li> <li>- businessPhone retrieved from CCB depends on the home phone type defined in CCB feature configuration schema constants.</li> <li>- From CCB Person Phone Table (CI_PER_PHONE)</li> </ul>
Customer Day Phone Extension cust_day_phone_ext	extension	<ul style="list-style-type: none"> <li>- This is the extension for the CCB businessPhone</li> <li>- From CCB Person Phone Table (CI_PER_PHONE)</li> </ul>
Customer Address 1 cust_address1	address 1	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer Address 2 cust_address2	address 2	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer Address 3 cust_address3	address 3	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer Address 4 cust_address4	address 4	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer City cust_city	city	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer State cust_state	state	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>
Customer Postcode 1 cust_postcode1	postal	<ul style="list-style-type: none"> <li>- From CCB Premise Table (CI_PREM)</li> </ul>

NMS Field	CCB Element Name	Notes
Premise ID serv_premise_ID	premiseId	- Unique external system generated primary key - From CCB Premise Table (CI_PREM)
Service Account Number serv_account_number	accountId	- Account number linked to the premise ID - From CCB Account Table (CI_ACCT)
Service Life Support serv_life_support	lifeSupport	- If the CCB premise or person life support/sensitive load flag is true, this will be also be set to true. - Use DVM <a href="#">OUCCB_OUNMS_LifeSupportInd</a>
Service Address 1 serv_address1	address 1	- From CCB Premise Table (CI_PREM)
Service Address 2 serv_address2	address 2	- From CCB Premise Table (CI_PREM)
Service Address 3 serv_address3	address 3	- From CCB Premise Table (CI_PREM)
Service Address 4 serv_address4	address 4	- From CCB Premise Table (CI_PREM)
Service City serv_city	city	- From CCB Premise Table (CI_PREM)
Service State serv_state	state	- From CCB Premise Table (CI_PREM)
Service City State serv_city_state	Concatenate city, state	- The delimiter that the integration uses for concatenation will always be (,) comma.
Service Postal serv_postcode1	postal	- From CCB Premise Table (CI_PREM)
Service D Priority serv_d_priority	medicalPriority	- D (critical) customers flag values in NMS, 0 or 1 - D Priority retrieved from premise characteristics depends on the char type defined from CCB feature configuration schema constants - From Premise Characteristic Table (CI_PREM_CHAR) - Use DVM OUCCB_OUNMS_Serv_D_Priority
Service C Priority serv_c_priority	criticalPriority	- C customers flag values in NMS, 0 or 1 - C Priority retrieved from premise characteristics depends on the char type defined from CCB feature configuration - From Premise Characteristic Table (CI_PREM_CHAR) - Use DVM OUCCB_OUNMS_Serv_C_Priority
Service K Priority	keyPriority	- K customers flag values in NMS, 0 or 1

NMS Field	CCB Element Name	Notes
serv_k_priority		<ul style="list-style-type: none"> <li>- K Priority retrieved from premise characteristics depends on the char type defined from CCB feature configuration</li> <li>- From Premise Characteristic Table (CI_PREM_CHAR)</li> <li>- Use DVM OUCCB_OUNMS_Serv_K_Priority</li> </ul>
Meter ID meter_ID	meterId	- From CCB Meter Table (CI_MTR)
Meter Number meter_number	badgeNumber	- From CCB Meter Table (CI_MTR)
Meter Type meter_type	meterType	<ul style="list-style-type: none"> <li>- From CCB Meter Table (CI_MTR)</li> <li>- Use DVM OUCCB_OUNMS_MeterType</li> </ul>
Meter Manufacturer meter_manufacturer	Manufacturer	- From CCB Meter Table (CI_MTR)

## Message Mapping

CCB SP Sync Request Message			NMS Stored Procedure Inputs	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
C1-NMSSPSyncRequest		OutermostTag		
syncRequestId	C1-NMSSPSyncRequest	Field		
bo	C1-NMSSPSyncRequest	Field		
boStatus	C1-NMSSPSyncRequest	Field		
createDateTime	C1-NMSSPSyncRequest	Field		
statusDateTime	C1-NMSSPSyncRequest	Field		
version	C1-NMSSPSyncRequest	Field		
syncRequired	C1-NMSSPSyncRequest	Field		
forceSync	C1-NMSSPSyncRequest	Field		
discardReason	C1-NMSSPSyncRequest	Field		
cancelReason	C1-NMSSPSyncRequest	Field		
message	C1-NMSSPSyncRequest	List		
sequence	message	Field		
messageCategory	message	Field		
messageNumber	message	Field		
comment	message	Field		

CCB SP Sync Request Message			NMS Stored Procedure Inputs	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
messageParameters	message	List		
parameterSequence	messageParameters	Field		
messageParameterValue	messageParameters	Field		
mo	C1- NMSSPSyncRequest	Field		
pkValue1	C1- NMSSPSyncRequest	Field		
pkValue2	C1- NMSSPSyncRequest	Field		
pkValue3	C1- NMSSPSyncRequest	Field		
pkValue4	C1- NMSSPSyncRequest	Field		
pkValue5	C1- NMSSPSyncRequest	Field		
personBO	C1- NMSSPSyncRequest	Field		
accountBO	C1- NMSSPSyncRequest	Field		
saBO	C1- NMSSPSyncRequest	Field		
spBO	C1- NMSSPSyncRequest	Field		
premiseBO	C1- NMSSPSyncRequest	Field		
meterBO	C1- NMSSPSyncRequest	Field		
itemBO	C1- NMSSPSyncRequest	Field		
spTypeBO	C1- NMSSPSyncRequest	Field		
snapshotDA	C1- MDM1SPSyncRequest	Field		
postScript	C1- MDM1SPSyncRequest	Field		
syncRequestDetails	C1- MDM1SPSyncRequest	Group		
initialSnapshot	syncRequestDetails	Group		
personInfo	initialSnapshot	Group		
personId	personInfo	Field		
name	personInfo	Field		
homePhone	personInfo	Field		
businessPhone	personInfo	Field		
extension	personInfo	Field		
lifeSupport	personInfo	Group		
customElements	personInfo	Group		
accountInfo	initialSnapshot	Group		
accountId	accountInfo	Field		
mainPerson	accountInfo	Field		
customElements	accountInfo	Group		
saInfo	initialSnapshot	Group		
sald	saInfo	Field		
accountId	saInfo	Field		
customElements	saInfo	Group		
spInfo	initialSnapshot	Group		

CCB SP Sync Request Message			NMS Stored Procedure Inputs	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
spld	splInfo	Field		
premiseId	splInfo	Field		
spType	splInfo	Field		
deviceId	splInfo	Field		
serviceType	splInfo	Field		
customElements	splInfo	Group		
premiseInfo	initialSnapshot	Group		
premiseId	premiseInfo	Field		
country	premiseInfo	Field		
address1	premiseInfo	Field		
address2	premiseInfo	Field		
address3	premiseInfo	Field		
address4	premiseInfo	Field		
houseType	premiseInfo	Field		
number1	premiseInfo	Field		
number2	premiseInfo	Field		
inCityLimit	premiseInfo	Field		
city	premiseInfo	Field		
geographic	premiseInfo	Field		
county	premiseInfo	Field		
state	premiseInfo	Field		
postal	premiseInfo	Field		
criticalPriority	premiseInfo	Field		
medicalPriority	premiseInfo	Field		
keyPriority	premiseInfo	Field		
lifeSupport	premiseInfo	Field		
customElements	premiseInfo	Group		
meterInfo	initialSnapshot	Group		
meterId	meterInfo	Field		
badgeNumber	meterInfo	Field		
meterType	meterInfo	Field		
manufacturer	meterInfo	Field		
customElements	meterInfo	Group		
itemInfo	initialSnapshot	Group		
itemId	itemInfo	Field		
badgeNumber	itemInfo	Field		
itemType	itemInfo	Field		
manufacturer	itemInfo	Field		
customElements	itemInfo	Group		
customElements	initialSnapshot	Group		
finalSnapshot	syncRequestDetails	Group		
personInfo	finalSnapshot	Group		
personId	personInfo	Field		
name	personInfo	Field	cust_name	
homePhone	personInfo	Field	cust_home_area_code, cust_home_phone	

CCB SP Sync Request Message			NMS Stored Procedure Inputs	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
businessPhone	personInfo	Field	cust_day_area_code, cust_day_phone	
extension	personInfo	Field	cust_day_phone_ext	
lifeSupport	personInfo	Group	serv_life_support	OUCCB_OUNMS_ LifeSupportIndicator
customElements	personInfo	Group		
accountInfo	finalSnapshot	Group		
accountId	accountInfo	Field		
mainPerson	accountInfo	Field		
customElements	accountInfo	Group		
salInfo	finalSnapshot	Group		
sald	salInfo	Field		
accountId	salInfo	Field	cust_ID , cust_account_number, serv_account_number	
customElements	salInfo	Group		
splInfo	finalSnapshot	Group		
spld	splInfo	Field	service_point_ID	
premiseId	splInfo	Field		
spType	splInfo	Field		
deviceId	splInfo	Field	device_ID	
serviceType	splInfo	Field	account_type	OUCCB_OUNMS_ AccountType
customElements	splInfo	Group		
premiseInfo	finalSnapshot	Group		
premiseId	premiseInfo	Field	serv_premise_ID	
country	premiseInfo	Field		
address1	premiseInfo	Field	cust_address1, serv_address1	
address2	premiseInfo	Field	cust_address2, serv_address2	
address3	premiseInfo	Field	cust_address3, serv_address3	
address4	premiseInfo	Field	cust_address4, serv_address4	
houseType	premiseInfo	Field		
number1	premiseInfo	Field		
number2	premiseInfo	Field		
inCityLimit	premiseInfo	Field		
city	premiseInfo	Field	cust_city, serv_city, serv_city_state	
geographic	premiseInfo	Field		
county	premiseInfo	Field		
state	premiseInfo	Field	cust_state, serv_state, serv_city_state	
postal	premiseInfo	Field	cust_postcode1	
criticalPriority	premiseInfo	Field	serv_c_priority	OUCCB_OUNMS_Serv_C_Priority
medicalPriority	premiseInfo	Field	serv_d_priority	OUCCB_OUNMS_Serv_D_Priority

CCB SP Sync Request Message			NMS Stored Procedure Inputs	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
keyPriority	premiseInfo	Field	serv_k_priority	OUCCB_OUNMS_Serv_K_Priority
lifeSupport	premiseInfo	Field	serv_life_support	
customElements	premiseInfo	Group		
meterInfo	finalSnapshot	Group		
meterId	meterInfo	Field	meter_ID	
badgeNumber	meterInfo	Field	meter_number	
meterType	meterInfo	Field	meter_type	OUCCB_OUNMS_MeterType
manufacturer	meterInfo	Field	meter_manufacturer	
customElements	meterInfo	Group		
itemInfo	finalSnapshot	Group		
itemId	itemInfo	Field		
badgeNumber	itemInfo	Field		
itemType	itemInfo	Field		
manufacturer	itemInfo	Field		
customElements	itemInfo	Group		
customElements	finalSnapshot	Group		

## Trouble Call Interface

### Mapping Details

NMS Field	CCB Message Element	Notes
Call Source ID <code>call_source_ID</code>		<ul style="list-style-type: none"> <li>- Integration will get the default value from the configuration properties file and map it to the NMS Call Source ID.</li> <li>- Trouble calls can be created from different external systems like CCB, IVR or web call entry. Each external system sending trouble calls to NMS will have a unique call_source_ID. This determines where the trouble call originated and makes sure the external ID passed to NMS is unique.</li> <li>- NMS prefixes this value to the external ID to make it unique.</li> </ul>
Service Point ID <code>service_point_ID</code>	spld	<ul style="list-style-type: none"> <li>- This is the outage call's SP ID.</li> <li>- For fuzzy calls, CCB will pass an empty value.</li> </ul>
External ID <code>external_ID</code>	outageCallId	<ul style="list-style-type: none"> <li>- CCB passes the Outage Call ID.</li> </ul>



NMS Field	CCB Message Element	Notes
Account Number <i>account_number</i>	accountId	This is the customer's account ID
Trouble Code <i>trouble_code</i>	outageCodes	<ul style="list-style-type: none"> <li>- Trouble code mapping setup between CCB and NMS must be the same.</li> <li>- In NMS the total length of the string is the total number of distinct groups in the SRS_TROUBLE_CODES table.</li> <li>- In CCB, it is the Number of Outage Code Characteristic.</li> </ul>
Call Time <i>call_time</i>	callDateTime	
CallbackFlag <i>callback_flag</i>	callbackRequested	<ul style="list-style-type: none"> <li>-In NMS, the possible values are as follows:  '0' - callback not requested  '1' - callback requested</li> <li>- Defaults to '1' if no value are supplied.</li> <li>- If Y or N are passed to NMS, it will be translated to:  'Y' is translated to '1'.  'N' is translated to '0'</li> <li>- use DVM OUCCB_OUNMS_CallBackIndicator to translate CCB value to NMS value</li> </ul>
Callback Before Time <i>callback_before_time</i>	callbackDateTime	
Alternate Phone Number <i>alt_phone</i>	callbackNumber	<ul style="list-style-type: none"> <li>- When the integration populates this field, it should strip off all delimiters and only pass the numeric values.</li> </ul>
Customer Phone <i>phone</i>	contactNumber	<ul style="list-style-type: none"> <li>- If trouble call is related to an SP, CCB pass customer phone</li> <li>- If fuzzy call, CCB pass the caller's phone</li> <li>- When the integration populates this field, it should strip off all delimiters and only pass the numeric values.</li> </ul>
Customer Name <i>first_name</i>	contactName	<ul style="list-style-type: none"> <li>- If trouble call is related to an SP, the integration maps it to the main person on the account</li> <li>- If fuzzy call, the integration maps it to the caller's name.</li> </ul>

NMS Field	CCB Message Element	Notes
Address Street <i>addr_street</i>	address1 or location1	<ul style="list-style-type: none"> <li>- If trouble call is related to an SP, the integration maps it to the customer's premise address1.</li> <li>- If fuzzy call, the integration maps it to location1 (It must contain a street name or free format location description).</li> </ul>
Address Street 2 <i>addr_cross_street</i>	location2	<ul style="list-style-type: none"> <li>- If fuzzy call, check locationType value</li> <li>--If Street Intersection, populate with location2 (It must contain a cross street)</li> </ul>
Address Building <i>addr_building</i>	blockNumber	<ul style="list-style-type: none"> <li>- If fuzzy call, check locationType value</li> <li>-- If Street Segment, populate with blockNumber (it must contain a number).</li> </ul>
Address City State <i>addr_city_state</i>	city state or locationCity locationState	<ul style="list-style-type: none"> <li>- If trouble call is related to an SP, CCB pass the premise city and state. Concatenation of City, State.</li> <li>- If fuzzy call, CCB pass the location city and location state. Concatenation of Location City, Location State.</li> <li>- The delimiter provided by the integration is always comma (,). If city or state is blank, no delimiter (,) is needed.</li> </ul>
Call ID <i>call_ID</i>	callIdentifier	<ul style="list-style-type: none"> <li>- This is the call identifier's external reference ID (i.e. 911 Reference Number)</li> </ul>
Call Taker <i>call_taker</i>	userFirstName userLastName	<ul style="list-style-type: none"> <li>- The name of the user who created the outage call also known as trouble call.</li> <li>- Integration will concatenate First name &lt;space&gt; Last Name</li> </ul>
Call Comment <i>call_comment</i>	comments	
Meet Type <i>meet_type</i>	meetType	<ul style="list-style-type: none"> <li>- In NMS, 1 if new appointment is set and '0' – default value for all other type jobs</li> <li>Valid Values:</li> <li>0 – for non-meet calls</li> <li>1 - create new meet</li> <li>2 - reschedule existing meet</li> <li>3- cancel existing meet</li> <li>- CCB gets the value from FA Char</li> <li>- use DVM OUCCB_OUNMS_MeetType to translate CCB value to NMS value</li> </ul>

NMS Field	CCB Message Element	Notes
Meet Time <i>meet_time</i>	meetDateTime	- CCB will populate meetDateTime only if meet type = 1
Call Cancel Flag <i>cancel_flag</i>	Y or null (base on Status)	- If Status is Canceled, the integration populate this field with Y, otherwise it is null - use DVM OUCCB_OUNMS_CallCancelIndicator to translate CCB value to NMS value
Update Existing Record Flag <i>update_flag</i>	outageCallAction	- The possible values in NMS are: 0 - insert new call 1 - update existing call - Use DVM OUCC_OUNMS_NewCallIndicator to translate CCB value to NMS value
Device ID <i>device_ID</i>	transformerId	May be null if trouble call is not related to a specific SP In CCB from SP Geo Code for Device

For Fuzzy Calls,

If Location is a **street intersection**, the mapping of the location from CCB to NMS will be  
p\_customer\_address = location1 (street name) and addr\_cross\_street (cross street).

If Location is a **street segment**, the mapping of the location from CCB to NMS will be  
p\_customer\_address = location1 (street name) and addr\_building (block number, this must only be a numeric value).

If Location is **other (free format description)**, the mapping of the location from CCB to NMS will be customer address = location1 (location description)

All DateTime coming from CCB will be converted to the ISO8601 format which is YYYY-MM-DDHH:MM:SS

## Message Mapping

CCB Trouble Call Request Message			NMS Trouble Call Stored Procedure Input	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
requestMessage		OutermostTag		
outageCallId	requestMessage	Field	external_ID	
spId	requestMessage	Field	service_point_ID	
premiseId	requestMessage	Field		

CCB Trouble Call Request Message			NMS Trouble Call Stored Procedure Input	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
accountId	requestMessage	Field	account_ID	
contactName	requestMessage	Field	first_name	
contactNumber	requestMessage	Field	phone	
callIdentifier	requestMessage	Field	call_ID	
callDateTime	requestMessage	Field	call_time	
OutageCallAction	requestMessage	Field	update_flag	OUCC_OUNMS_NewCallIndicator
Status	requestMessage	Field	cancel_flag	OUCCB_OUNMS_CallCancelIndicator
faComments	requestMessage	Field	call_comment	
userId	requestMessage	Field		
userFirstName	requestMessage	Field	call_taker	
userLastName	requestMessage	Field	call_taker	
country	requestMessage	Field		
addressLine1	requestMessage	Field	addr_street	
addressLine2	requestMessage	Field		
addressLine3	requestMessage	Field		
addressLine4	requestMessage	Field		
houseType	requestMessage	Field		
number1	requestMessage	Field		
number2	requestMessage	Field		
inCityLimit	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	meet_time	
meetType	requestMessage	Field	meet_type	OUCCB_OUNMS_MeetType
outageCodes	requestMessage	Field	trouble_code	
transformerId	requestMessage	Field	device_ID	

CCB Trouble Call Request Message			NMS Trouble Call Stored Procedure Input	DVM Mapping
Element Name	Parent Element	Type	Element Name	DVM
callbackRequested	requestMessage	Field	callback_flag	OUCCB_OUNMS_CallBackIndicator
callbackDateTime	requestMessage	Field	callback_before_time	
callbackNumber	requestMessage	Field	alt_phone	
customElements	requestMessage			

## Job History Query

### Message Mapping

#### Request Message

CCB Request Message Elements		NMS FIELD	DVM
<jobHistoryRequest>			
	spld	CID	
	premiseId	PREMISEID	
	accountId	ACCOUNTNUMBER	
	location		
	cityStreet	CITY	
	state	STATE	
	intersection		
	street1	STREET1	
	street2	STREET2	
	intersection		
	segment		
	street	STREET1	
	number	BLOCKNUMBER	
	segment		
	location		
	externalId	EXTERNALID	
	callIdentifierId	CALLID	
	callerName	CALLERNAME	
	callerPhoneNumber	CALLERPHONE	
		P_CMP_DAYS	
</jobHistoryRequest>			

## Response Message

Response Message	NMS FIELD	DVM
<jobHistoryResponse>		
<jobHistory>		
spld	CID	
jobStartDateTime	BEGIN_TIME	
troubleLocation	TROUBLE_LOCATION	
estimatedRestorationDateTime	EST_REST_TIME	
actualRestorationDateTime	RESTORE_TIME	
ertType	EST_SOURCE	OUCCB_OUNMS_ETR_Source
eventStatus	STATUS	OUCCB_OUNMS_Status
alarmState	ALARM_STATE	OUCCB_OUNMS_Alarm_State
customersAffected	NUM_CUST_OUT	
jobId	EVENT_IDX	
operatorComment	OPERATOR_COMMENT	
deviceClass	DEVCLS_NAME	
troubleDescription	TROUBLE_CODE	
feeder	FEEDER_NAME	
primaryCause	CAUSE	OUCCB_OUNMS_Cause
actionTakenDescription	DESCRIPTION	
referralGroup	REFERRAL_GROUP	
	P_ERR_NO	OUCCB_OUNMS_Error_Code
	P_ERR_MSG	
	UDF1	
	UDF2	
	UDF3	
	UDF4	
	UDF5	
</jobHistory>		
<exceptionInfo>	<messageCategory/>	
	<messageNumber/>	
	<comments/>	
	<messageParms>	
	<parmSequence/>	
	<messageParmValue/>	
	</messageParms>	
</exceptionInfo>		
</jobHistoryResponse>		

## Trouble Call History Query

### Message Mapping

#### Request Message

CCB Request Message		NMS FIELD	DVM
<callHistoryRequest>			
	spld	CID	
	premiseId	PREMISEID	
	accountId	ACCOUNTNUMBER	
	location		
	cityStreet	CITY	
	state	STATE	
	intersection		
	street1	STREET1	
	street2	STREET2	
	intersection		
	segment		
	street	STREET1	
	number	BLOCKNUMBER	
	segment		
	location		
	externalId	EXTERNALID	
	callIdentifierId	CALLID	
	callerName	CALLERNAME	
	callerPhoneNumber	CALLERPHONE	
		P_NUM_DAYS	
</callHistoryRequest>			

#### Response Message

Response Message		NMS FIELD	DVM
<callHistoryResponse>			
<callHistory>			
	seq	NUMB	
	spld	CID	
	callDate	INPUT_TIME	
	location	ADDRESS	
	complaintDescription	SHORT_DESC	
	callComments	OP_COMMENT	
	issuer	USER_NAME	
	referenceNumber	EXTERNALID	
	contactName	CUSTOMER_NAME,	

Response Message		NMS FIELD	DVM
	callStatus	ACTIVE	OUCCB_OUNMS_Call_Status
	callId	GENERAL_AREA	
		P_ERR_NO	
		P_ERR_MSG	
		UDF1	
		UDF2	
		UDF3	
		UDF4	
		UDF5	
</callHistory>			
<exceptionInfo>	<messageCategory/>		
	<messageNumber/>		
	<comments/>		
	<messageParms>		
	<parmSequence/>		
	<messageParmValue/>		
	</messageParms>		
</exceptionInfo>			
</callHistoryResponse>			

## Planned Outages Query

### Message Mapping

#### Request Message

CCB Request Message Elements		NMS FIELD	DVM
< plannedOutages Request>			
	spId	CID	
	showAllPlannedOutages		
		P_NUM_DAYS	
</ plannedOutages Request>			



## Response Message

Response Message			NMS FIELD	DVM
< plannedOutages Response>				
<plannedOutages>				
		planClass	PLANCLASS	
	outageNumber	planNumber	PLANNUMBER	
	plannedStartDateTime	plannedStartDateTime	START_DATE	
	plannedEndDateTime	plannedEndDateTime	FINISH_DATE	
	plannedOutageSatus	state	STATE	
	workDistrict	workDistrict	WORKDISTRICT	
	workLocation	workLocation	WORKLOCATION	
	workDescription	workDescription	WORKDESCRIPTION	
		errorCode	P_ERR_NO	
		errorMessage	P_ERR_MSG	
		userDefinedField1	UDF1	
		userDefinedField2	UDF2	
		userDefinedField3	UDF3	
		userDefinedField4	UDF4	
		userDefinedField5	UDF5	
		userDefinedField6	UDF6	
		userDefinedField7	UDF7	
		userDefinedField8	UDF8	
		userDefinedField9	UDF9	
		userDefinedField10	UDF10	
<plannedOutages>				
<exceptionInfo>	<messageCategory/>	<messageCategory/>		
	<messageNumber/>	<messageNumber/>		
	<comments/>	<comments/>		
	<messageParms>	<messageParms>		
	<parmSequence/>	<parmSequence/>		
	<messageParmValue/>	<messageParmValue/>		
	</messageParms>	</messageParms>		
</exceptionInfo>				
</ plannedOutages Response>				

## Appendix B: Cross References

The following sections provide references for where you can find more information on some of the terms and entities related to this the integration.

### **Domain Value Maps**

For more information refer the chapters [Working with Domain Value Maps](#) and [Using SOA Composer with Domain Value Maps](#) in the Oracle® Fusion Middleware Developer's Guide for Oracle SOA Suite.

### **JMS Adapter**

For more information please refer to [Oracle Fusion Middleware User's Guide for Technology Adapters/](#).

### **Process Flow Diagrams**

The following supplemental process flow diagrams are included with the documentation package.

- 050601 Maintain Customer Information CCB-NMS (Reusable Subprocess).pdfs
- 050000 Customer Service.pdf
- 050603 Fulfill Simple Request.pdf