Oracle Utilities Transaction Management

Installation and Configuration Guide

Release 1.6.1.23 for Windows

E18219-24

December 2018

(Revised April 2020)

Oracle Utilities Transaction Management/Transaction Management Installation and Configuration Guide, Release 1.6.1.23 for Windows

E18219-24

Primary Author: Lou Prosperi

Contributor: Steve Pratt

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 is software or related documentation that 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 that 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.

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 loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

NOTIFICATION OF THIRD-PARTY LICENSES

Oracle Utilities software contains third party, open source components as identified below. Third- party license terms and other third-party required notices are provided below.

License: Apache 1.1

Module: xercesImpl.jar, xalan.jar

Use of xercesImpl and xalan within the product is governed by the following (Apache 1.1):

wherever such third-party acknowledgments normally appear. (4) Neither the component name nor Apache Software Foundation may be used to endorse or promote products derived from the software without specific prior written permission. (5) Products derived from the software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: Paul Johnston

Modules: md5.js

Copyright (C) Paul Johnston 1999 - 2002

Use of these modules within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor the names of the copyright holders and contributors may be used to endorse or promote products derived from the software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

License: Tom Wu

Module: jsbn library

Copyright © 2003-2005 Tom Wu. All rights reserved

Use of this module within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Contents

Contents

Chapter 1

Chapter 2

Chapter 3

Chapter 4

Chapter 5

Chapter 6

Appendix A

Appendix B

Index

Chapter 1

Overview

This chapter provides an overview of the installation and configuration of the Oracle Utilities Transaction Management, including:

- **Installation and Configuration Overview**
- **What is this book?**
- **Oracle Utilities Transaction Management Glossary**

Installation and Configuration Overview

Installing and configuring Oracle Utilities Transaction Management involves the following steps:

- Setup the Oracle Utilities Transaction Management network environment as described in **Chapter 2**: **Oracle Utilities Transaction Management Network Environment and Database Installation**.
- Install the Oracle Utilities Transaction Management Database as described in **Chapter Two: Oracle Utilities Transaction Management Database and Server Component Installation** and **Chapter Three**: **Oracle Utilities Data Repository Schema Creation** of the *Oracle Utilities Energy Information Platform Installation and Configuration Guide*.
- Install server components on application server as described in **Chapter Two: Oracle Utilities Transaction Management Database and Server Component Installation**.
- Install Oracle Utilities Transaction Management server and runtime components on application server as described in **Chapter Three: Installing the Oracle Utilities Transaction Management Application Software**.
- Install Oracle Utilities Transaction Management Configuration Workbench on client machines as described in **Chapter Three: Installing the Oracle Utilities Transaction Management Application Software.**
- Install Oracle Utilities Transaction Management Web components on web server as described in **Chapter Three: Installing the Oracle Utilities Transaction Management Application Software**
- Configure Oracle Utilities Transaction Management documents, transactions, and processes using the Oracle Utilities Transaction Management Configuration Workbench as described in **Chapter Four: Configuring Oracle Utilities Transaction Management using the Configuration Workbench**.

What is this book?

The *Oracle Utilities Transaction Management Installation and Configuration Guide* describes how to install and configure Oracle Utilities Transaction Management, including the following:

- **Chapter 1**: **Overview** provides an overview of the Oracle Utilities Transaction Management installation and configuration process
- **Chapter 2**: **Oracle Utilities Transaction Management Network Environment and Database Installation** describes how to install the Oracle Utilities Transaction Management Database and application server components needed to run Oracle Utilities Transaction Management.
- **Chapter 3**: **Installing the Oracle Utilities Transaction Management Application Software** describes how to install Oracle Utilities Transaction Management application software on an application server, on a web server, and on client machines.
- **Chapter 4**: **Configuring Oracle Utilities Transaction Management** describes how to set up and configure Oracle Utilities Transaction Management to function within a specific market.
- **Chapter 5**: **Oracle Utilities Transaction Management Server and Monitor** describes the Oracle Utilities Transaction Management Server and Oracle Utilities Transaction Management Monitor used by Oracle Utilities Transaction Management administrators.
- **Chapter 6**: **Configuring Oracle Utilities Transaction Management Security** describes how to configure security for use with Oracle Utilities Transaction Management.
- **Appendix A**: **Oracle Utilities Data Repository Transaction Management Database Schema** describes a number of validation scripts used by Oracle Utilities Transaction Management.
- **Appendix B**: **Oracle Utilities Transaction Management Validation Scripts** describes a number of validation scripts used by Oracle Utilities Transaction Management.

Oracle Utilities Transaction Management Glossary

Auto Response Transactions

A response transaction that is automatically generated by Oracle Utilities Transaction Management. This is typically a response transaction to illustrate acceptance or rejection of a sending transaction.

Business Rule

A Business Rule is a “program” registered within Oracle Utilities Transaction Management to run as part of a business process. These custom business rules can be used to perform numerous functions but are typically used to lookup values in a database or to post transaction data. Custom business rules can be Oracle Utilities Rules Language, PL SQL, Java, Visual Basic or other.

Business Rule Override

The business rule override function allows the user to ignore a business rule that caused a transaction to fail. Please note this function can be dangerous and could allow a user to accept invalid transactional data.

Coupling Exceptions

The Oracle Utilities Transaction Management Transaction Coupling service identifies each new transaction and relates it to an associated business process. Any transactions that cannot be either identified as the initial transaction for a business process, or coupled to an associated business process that has already been started, are logged as an exception indicating an unsolicited or **Orphaned Transaction**. The exception will remain until all of the required transactions have been identified.

Coupling Service

In order to manage individual transactions in the context of a business process, each of the transactions must be identified and associated with other transactions that constitute the business process. For example, in the Texas ERCOT market place, an enrollment process consists of an 814_01 enrollment request, 814_02 enrollment response, and 867_02 historical usage request.

Data Validation

The Data Validation Engine is an XML-based processor that leverages XML and Xpath concepts to perform market specific validations on transactions. For example, although an EDI 814 transaction may be used for customer enrollment in several markets or jurisdictions, the validation rules for the Texas market are different than the validations for the Pennsylvania market. The Data Validation Engine allows the user to configure validation rules, and associate these rules with one or more transactions. In the event that validation failures require a response transaction be sent, the Data Validation Engine can be configured to invoke the Oracle Utilities Transaction Management Auto Response Service to initiate the response transaction to the originating party.

Please note, the Data Validations Engine goes beyond standard XML Schema validations. It provides additional validation capability as well as the ability record transaction failures.

Document

XML Documents used by Oracle Utilities Transaction Management.

Exception Service

The Oracle Utilities Transaction Management Exception processing service provides the user with a way to quickly and easily identify exceptions and errors that occur. All exceptions are logged to work queues, which can then be viewed from the Web-based user interface.

Expirations Service

In many business processes, there are market rules related to response times. For example, market rules often dictate that for a customer enrollment request, an enrollment response transaction must be delivered within a certain number of business days. The Oracle Utilities

Transaction Management Expiration Service allows the user to configure expiration timers around the business processes according to the market rules. In the enrollment example, if the enrollment request transaction is sent out and the enrollment response is not received in the user-specified number of business days, an exception is logged indicating that there is a problem with that particular enrollment process. Expirations can be set to one of two levels, either soft or hard expirations. Soft expirations log an exception, but allow the business process to continue. Hard expirations log an exception and terminate the associated business process.

Mapping Service

The Oracle Utilities Transaction Management mapping service converts XML from one format to another using standard XSLT scripts.

Message Identification Service

The Oracle Utilities Transaction Management Transaction Identification service compares the received transactions with the configuration library to determine the type and context of the transaction. Once this has been established, the appropriate validations and business process rules can be applied.

Notify Transactions

Within Oracle Utilities Transaction Management several processes can be joined through the use of a notify transaction. This is where one process notifies another process that a transaction has been received. For example, a final usage transaction may be used by a usage and billing process to provide a final bill to the client as well as to end a drop process.

Optional Transactions

A transaction that is not required by a process.

Orphaned Transaction

A transaction that cannot be identified or coupled to a business process.

Process

A process is a comprehensive set or electronic transactions that are combined to form a business process. For example, a switch process can consist of multiple transactions: a switch request, switch response, historical usage, switch cancel and final usage.

Resubmit

The resubmit feature allows the user to resubmit and “retry” a failed transaction. This feature is very useful when working exceptions. When the resubmit function is used, the transaction is retried at the point of failure. If the error condition no longer exist, the transaction will be processed normally. If the error condition still exist, the transaction will remain in an error state.

Schema Validations

Schema validations are used to ensure transactions conform to the schema rules for each transaction. See **Schemas** below.

Schemas

Per the W3C, The purpose of the XML schema language is to provide an inventory of XML markup constructs with which to write schemas. The purpose of a schema is to define and describe a class of XML documents by using these constructs to constrain and document the meaning, usage and relationships of their constituent parts: datatypes, elements and their content, attributes and their values, entities and their contents and notations. Schema constructs may also provide for the specification of implicit information such as default values. Schemas document their own meaning, usage, and function. Thus, the XML schema language can be used to define, describe and catalogue XML vocabularies for classes of XML documents.

Task

An activity performed on a transaction. For example, transform the transaction from one format to another.

Terminate

The terminate feature is used to manually end a transaction. This is useful when a transaction is no longer expected.

Transaction

An electronic transaction sent from a trading or backend system to another trading partner or backend system. Examples include invoice or usage transactions.

XML (eXtensible Markup Language)

A language that specifies how data is to “marked-up” to identify the specific data elements included in a transmission. Very similar to HTML but also includes data dictionaries for the data transmitted.

Chapter 2

Oracle Utilities Transaction Management Network Environment and Database Installation

This chapter describes the Oracle Utilities Transaction Management network environment and how to install the Oracle Utilities Date Repository used by Oracle Utilities Transaction Management, including:

- **The Oracle Utilities Transaction Management Installation Package**
- **Oracle Utilities Transaction Management Network Environment**
- **Oracle Utilities Transaction Management Database Installation**
- **Oracle Utilities Transaction Management Message Service Installation**

The Oracle Utilities Transaction Management Installation Package

There is a separate installation package for the Oracle Utilities Energy Information Platform and for each related product. The table below provides the installation package file names for the Oracle Utilities Energy Information Platform and Oracle Utilities Transaction Management.

Product	Installation File Name
Oracle Utilities Energy Information Platform	1.6.1.xx.0.EIP.zip
Oracle Utilities Transaction Management	1.6.1.xx.0.TM.zip

The Oracle Utilities Transaction Management installation package contains the following folders:

- **Install**: Contains the installation program for Oracle Utilities Transaction Management, including:
  - Oracle Utilities TM 1.6.1.xx.0.msi
- **DBScripts**: Contains the following database scripts for Oracle Utilities Transaction Management:
  - Install: scripts used to create a new Transaction Management schema
  - Upgrade: scripts used to update the Transaction Management schema

  See **Oracle Utilities Transaction Management Database Installation** on page 2-9 for more information about installing and upgrading the Oracle Utilities Transaction Management database schema.
- **Documentation**: Contains documentation for Oracle Utilities Transaction Management, including:
  - Oracle Utilities Transaction Management Installation and Configuration Guide
  - Oracle Utilities Transaction Management User's Guide

    Note: Oracle Utilities Transaction Management documentation can also be downloaded separately.

Oracle Utilities Transaction Management Network Environment

This section contains options for hardware and software specifications for implementation of the Oracle Utilities Energy Information Platform and related products, including:

- Oracle Utilities Billing Component
- Oracle Utilities Load Analysis
- Oracle Utilities Load Profiling and Settlement
- Oracle Utilities Meter Data Management
- Oracle Utilities Quotations Management
- Oracle Utilities Rate Management
- Oracle Utilities Transaction Management

The Energy Information Platform application architecture comprises all of the components typically found in any n-tier enterprise architecture, including client machines, servers, and the supporting network infrastructure.

The Oracle Utilities Energy Information Platform uses five primary types of systems components. Suggested hardware specifications and required software for each component are listed below.

Before installing Oracle Utilities software you should consult with Oracle Utilities Consulting. They will make suggestions for your hardware and network architecture based on your business requirements. You should also consult with Oracle Utilities Consulting if your business requirements change, if your database grows significantly, or if you plan on migrating to a different version of Oracle Utilities software.

> **Note:** Supported software versions are subject to change. Please refer to the Release Notes shipped with the software for the latest supported versions.

Client Workstation - C/S

These are workstations used when running the client/server (C/S) versions of Oracle Utilities applications, such as Data Manager.

Software for Client Workstation - C/S:

- Supported Operating Systems
    - Windows 8.1
    - Windows 10
- Supported Web Browsers
    - Microsoft Internet Explorer 11 with latest security updates
    - Microsoft Edge (Windows 10 only)
- Relational Database Management System (RDBMS) client with applicable ODBC driver and Data Provider for .NET

  Note: When using 64-bit operating systems, the 32-bit version of the database client is required.
- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher

Client Workstation - Web

These are workstations used when running the web-enabled versions of Oracle Utilities applications only.

Software for Client Workstation - Web:

- Supported Operating Systems
  - Windows 8.1
  - Windows 10
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge (Windows 10 only)

Note: For users that require access to both client/server and web browser-based applications, please refer to the requirements listed for "Client Workstations - Thick Client."

Database Server

The database server houses the Oracle Utilities Data Repository. The hardware platform for the database server may be Windows/Intel or UNIX/RISC.

Software for Database Server:

- Supported Operating Systems
    - Microsoft Windows Server 2012
    - Microsoft Windows Server 2016
    - LINUX (UNIX/Intel)
    - Sun Microsystems Solaris
    - HP-UX
    - IBM AIX (UNIX/RISC) software

  Note: Refer to the *Oracle Utilities Energy Information Platform Quick Install Guide* for additional details about supported operating systems.
- RDBMS Server: Oracle

Database Platforms:

- Oracle 11g, Release 2 (11.2.0.4)
- Oracle 12c, Release 1 (12.1.0.2 or 12.2.0.1)
- Oracle 19c, Release 1 (19.3.0.0.0)

Note: Supported database versions and drivers are subject to change. Please refer to the Oracle Release Notes shipped with the software for the latest compatibility matrix.

Application/Batch Processing Server

Application servers are used when running Windows Services, such as those employed by the Reporting Framework and Adapter components. The application server(s) can also perform batch processes such as meter data upload and validation as well as batch billing and settlements.

Software for the Application/Batch Server:

- Supported Operating Systems
  - Windows 2012 Server R2
  - Windows 2016 Server
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge with latest security updates (Windows 2016 Server only)
- RDBMS client with applicable ODBC driver and Data Provider for .NET

  Note: When using 64-bit operating systems, the 32-bit version of the database client is required.
- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher
- Java Development Kit (JDK) 1.8 or later (for application servers that will be running the Adapter)

Web Server(s)

Web servers run Microsoft Internet Information Service (IIS). These servers may be combined with the application server(s) for combination Web and Application servers.

Software for Web Server(s):

- Supported Operating Systems
  - Windows 2012 R2 Enterprise Server with Microsoft Internet Information Service (IIS) 8.5
  - Windows 2016 Enterprise Server with Microsoft Internet Information Service (IIS) 10.0
- Supported Web Browsers
  - Microsoft Internet Explorer 11 with latest security updates
  - Microsoft Edge with latest security updates (Windows 2016 Server only)
- RDBMS client with applicable ODBC driver and Data Provider for .NET

  Note: When using 64-bit operating systems, the 32-bit version of the database client is required.
- Oracle Utilities Energy Information Platform software
- Microsoft .NET Framework 4.5
- Microsoft Visual C++ Redistributable Packages for Visual Studio 2013 (32-bit)
- Microsoft Data Access Component (MDAC) 2.8 or higher
- Microsoft XML Core Services (MSXML) 6.0 or higher

Oracle Utilities Transaction Management Database Installation

This section explains how to install Oracle Utilities Transaction Management database tables and data in the Oracle Utilities Data Repository, including:

- **Installation Requirements**
- **Installing the Database**
- **Upgrading the Transaction Management Database Schema**
- **Updating the Reporting Framework for Oracle Business Intelligence Publisher**
- **Verifying the Database**

> **Note**: This section assumes that you have created the Oracle Utilities Data Repository schema as described in **Chapter 3**: **Oracle Utilities Data Repository Schema Creation** in the *Oracle Utilities Energy Information Platform Installation Guide.*

Installation Requirements

The following are required in order to install the Oracle Utilities Transaction Management tables and data into the Oracle Utilities Data Repository:

- The Oracle Utilities Data Repository schema (v1.6.1.0.0) must have been installed on the same database instance on which you plan to install the Oracle Utilities Transaction Management database.
- The **addTM.cmd** file. This database script adds the Oracle Utilities Transaction Management database tables and data to the Oracle Utilities Data Repository.
- The **updateTM.cmd** file. This database script updates the tables and data used by Oracle Utilities Transaction Management in the Oracle Utilities Data Repository. See **Upgrading the Transaction Management Database Schema** on page 2-10 for more information.
- The **BIPTM.cmd** file. This database script changes the pre-defined Transaction Management reports from Crystal Reports to Oracle Business Intelligence Publisher. See **Updating the Reporting Framework for Oracle Business Intelligence Publisher** on page 2-10 for more information.

Installing the Database

After determining which database scripts you need to run (consult release notes) open a command prompt and run the addTM.CMD script. This script uses the following syntax:

```
addTM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own the database objects. This parameter is optional. If not specified, the default user PWRLINE will own database objects.
<owner password>	The chosen password for the PWRLINE schema owner.

Like the base schema database tables and indexes, the Oracle Utilities Transaction Management objects are created in the default tablespace of the PWRLINE user with default sizing parameters. If these defaults are required to be changed then the scripts may be edited.

Upgrading the Transaction Management Database Schema

If you are upgrading Oracle Utilities Transaction Management from a previous version, you must upgrade Transaction Management database schema. The following database upgrade scripts are included in the Transaction Management installation package.

- The **updateTM.cmd** file. This database script updates the tables and data used by Oracle Utilities Transaction Management in the Oracle Utilities Data Repository. This script can only be used when upgrading the following schema version.

v1.6 Schema Version	v1.6.1 Schema Version
v1.6.0.0.0	v1.6.1.0.0

If you are upgrading from a different schema version, contact Oracle Global Customer Support.

Upgrading Oracle Utilities Transaction Management Schema

Open a command prompt and run the updateTM.CMD script. This script uses the following syntax:

```
updateTM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own Oracle Utilities database objects. This parameter is optional. If not specified, the default Oracle Utilities user PWRLINE will own database objects.
<owner password>	The chosen password for the PWRLINE schema owner.

Updating the Reporting Framework for Oracle Business Intelligence Publisher

If you are using Oracle Business Intelligence Publisher for reporting, you must update the pre-defined reports in Reporting Framework database tables. The following database scripts are included in the Transaction Management installation package for this purpose.

- The **BIPTM.cmd** file. This script changes the pre-defined Transaction Management reports from Crystal Reports to Oracle Business Intelligence Publisher.

Oracle Utilities Transaction Management Reports

Open a command prompt and run the BIPTM.CMD script. This script uses the following syntax:

```
BIPTM [-d <database>] [-own <owner name>] -opw <owner password>
```

Parameter	Description
<database>	The name given to the instance as specified in the TNSNAMES.ORA file. This parameter is optional and if not specified, the script will connect to the default Oracle database.
<owner name>	The name of the user which will own Oracle Utilities database objects. This parameter is optional. If not specified, the default Oracle Utilities user PWRLINE will own database objects.
<owner password>	The chosen password for the PWRLINE schema owner.

Verifying the Database

Verifying the Oracle Utilities Transaction Management database involves verifying the Oracle Utilities Transaction Management database schema in the Oracle Utilities Date Repository.

Verification - Transaction Management Tables

To verify that the Oracle Utilities Transaction Management schema tables are in place use the following procedure:

1. Log into the database using the Oracle Utilities Transaction Management user (Password =password).
2. Verify that data exists in the following tables:
   - DataType
   - ColumnMetaData
   - HubException
   - HubQueue
   - HubService
   - HubServiceType
   - HubSeverity
   - HubStatus
   - HubSystem
   - HubTokenSchema
   - Jurisdiction
   - XMLAttribute
   - XMLElement
   - HubTransException
   - HubTransExpiration
   - HubTaskInstance
   - HubTokenException

- IceRequestException
- HubInboundException
- HubOutboundException
- HubToken
- HubPayload
- HubProcessTrans_RT
- HubProcTransInstance
- HubTransInstance
- HubProcInstance
- HubProcInstance_Audit
- HubTransInstance_Audit
- HubTaskInstance_Audit
- IceResponse
- HubInbound
- IceRequest
- HubOutbound
- IceFa
- IcePayload
- IceDocDownload
- HubAdapterExcep
- HubAdapterInter
- LsBusinessUnit
- LsReportUnit
- LsReport

Appendix A: **Oracle Utilities Data Repository Transaction Management Database Schema** includes a diagram of the Oracle Utilities Load Profiling and Settlement database schema (v1.6.1.0.0) that provides details regarding the table and columns in the Oracle Utilities Load Profiling and Settlement schema, as well as the relationships between these tables in the Oracle Utilities Data Repository.

Verification - Transaction Management Reports

To verify that the Transaction Management reports have been updated, use the following procedure:

1. Log into the database using the PWRLINE user (Password =password).
2. Verify that the following records exist in the Report Instance table:

Report Name	Description	Report Type
TransactionControl	Transaction Control Summary	BIPublisher
TransactionDetail	Transaction Detail	BIPublisher
ExceptionDetail	Exception Detail	BIPublisher

Report Name	Description	Report Type
LTMHTransSummaryByType	Transaction Summary by Transaction Type	BIPublisher
LTMHExcepSummByType	Exception Summary by Exception Type	BIPublisher
LTMHExcepSummByJuris	Exception Summary by Jurisdiction	BIPublisher
LTMHExcepSummByTrans	Exception Summary by Transaction Type	BIPublisher
LTMHTransSummByJuris	Transaction Summary by Jurisdiction	BIPublisher
LTMHTransSummByStatus	Transaction Summary by Status	BIPublisher

Oracle Utilities Transaction Management Message Service Installation

The architecture of Oracle Utilities Transaction Management requires that Java Message Service (JMS) compliant messaging software be installed on at least one machine that is accessible via your Local Area Network (LAN). The server where the messaging software and its components will reside **must** be on the same domain and subnet as the Oracle Utilities Transaction Management server.

Message Queue Setup and Configuration

After installing the message service, you must setup and configure the message queues used by Oracle Utilities Transaction Management. The required queues are as follows:

Queue Name	Queue Type*	Cluster Name*
QAUORESPONSE	Local	DEFAULT_CLUSTER
QCOUPLER	Local	DEFAULT_CLUSTER
QEXCEPTION	Local	DEFAULT_CLUSTER
QINPUT	Local	DEFAULT_CLUSTER
QMAP	Local	DEFAULT_CLUSTER
QOUTPUT	Local	DEFAULT_CLUSTER
QPROCESSBROKER	Local	DEFAULT_CLUSTER
QSERVICEBROKER	Local	DEFAULT_CLUSTER
QTRANSID	Local	DEFAULT_CLUSTER
QVALIDATION	Local	DEFAULT_CLUSTER
QRESUBMIT	Local	DEFAULT _CLUSTER
QBUSINESSBROKER	Local	DEFAULT_CLUSTER
QEBTIN	Local	DEFAULT_CLUSTER
QEBTOUT	Local	DEFAULT_CLUSTER
QHTTPS	Local	DEFAULT_CLUSTER

*The Queue Type and Cluster Name are based on using IBM MQ Series as the message service. These may differ if you're using different message service software.

Message Server Setup

The last step is to setup the message servers that will be used by Oracle Utilities Transaction Management. This includes:

- **Bindings File Setup**
- **JMS Server Setup (on the database)**

Bindings File Setup

Note: At a minimum, the following IBM MQSeries items **MUST BE INSTALLED**:

- Classes for Java V5.1.2
- Classes for JAVA Messages Service V1.1.

1. From a DOS prompt, go to the following directory (the drive where you installed MQSeries Java Components):

   …../**MQSeries/Java/bin/**

2. Open the file “JMSAdmin.config” and comment out the following lines:

   ```
   INITIAL_CONTEXT_FACTORY=com.sun.jndi.ldap.LdapCtxFactory
   PROVIDER_URL=ldap://polaris/o=ibm_us,c=us
   ```

3. Uncomment the following lines in the same file:

   ```
   INITIAL_CONTEXT_FACTORY=com.sun.jndi.fscontext.RefFSContextFactory
   PROVIDER_URL=file:/C:/JNDI-Directory
   ```

4. From the DOS prompt, run “JMSAdmin.bat” (installdir/MQSeries/java/bin/JMSAdmin –v)

5. Make sure your Java Virtual Machine (JVM) is in your classpath.

   From the InitCtx> prompt, type in the following, using your machine variables:

   ```
   DEFINE QCF(JMSCONNECTON NAME) HOST(HOSTNAME) CHANNEL(JAVA.CHANNEL)
   TRAN(CLIENT) QMGR(QM) PORT(1414)
   ```

   Example:

   ```
   DEFINE QCF(JMSCONN-DELTA) HOST(DELTA) CHANNEL(JAVA.CHANNEL)
   TRAN(CLIENT) QMGR(QM_delta) PORT(1414)
   ```

   Remember that the syntax of the QMGR is case-sensitive; it must be exactly the same as in MQSeries Explorer (i.e., “QM_Delta”, “QM_DELTA”, and “QM_delta” are all different queue manager names).

6. The JNDI entry will be created in a “.binding” file under the “PROVIDER_URL” (C:/JNDI-Directory) defined above. Copy this file to:

   ```
   ../Lodestar/LTMH/runtime/jndi/
   ```

This completes the bindings file setup for the JMS Server that Oracle Utilities Transaction Management 2.0 will use.

Note

The above is based on using IBM MQ Series as the message service. The specific steps may differ if you’re using different message service software.

JMS Server Setup (on the database)

After you install the message service software, follow these steps to set up JMS Servers on the database. These steps assume that the database portion of the Oracle Utilities Transaction Management setup has been completed.

- Add the JMS Server (the machine where the JMS components are installed) to JMSSERVER Table with the name field set to its appropriate JNDI lookup name (see Step 4 of **Bindings File Setup** for lookup name "JMSCONN-DELTA"). Leave the remaining fields blank.

Multiple Oracle Utilities Transaction Management Runtime Services and their combinations can be set up to run for a Oracle Utilities Transaction Management Server instance on a given host machine. Only one instance of Oracle Utilities Transaction Management Server is allowed on a host. The configuration needs to be set up beforehand following these steps:

1. Add your Oracle Utilities Transaction Management Server record to the HUBSERVER Table; SERVERNAME is the same as the machine name where the Oracle Utilities Transaction Management Server will be running.
2. Populate the HUBQUEUE and HUBSERVICE tables, if necessary. *(This should already have been done during database initialization.) Verify that the queues on page 2-14 are present in the HUBQUEUE table.*
3. Configure runtime services for this Oracle Utilities Transaction Management Server in the HUBSERVERSERVICE Table, if necessary. *(This should already have been done during database initialization.)*
4. The ENABLED flag in this table should be set to "Y" if the service is part of the Oracle Utilities Transaction Management Server. Setting the RUNATINIT column to "Y" indicates this service will be automatically started when LTMHSERVER is launched.
5. Set up the HUBSYSPROPS Table for this Oracle Utilities Transaction Management Server:
    - Set the JMS Server ID (UIDJMSSERVER from JMSSERVER Table)
    - Set the jar file class path as follows:

```
   .\ltmh20.jar .\lib\jms\jms.jar .\lib\ibm\com.ibm.mq.jar
.\lib\ibm\com.ibm.mqjms.jar .\lib\xml\xerces.jar
.\lib\xml\xalan.jar .\lib\xml\xml-apis.jar
.\lib\jdbc\classes12.zip .\lib\jndi\fscontext.jar
.\lib\jndi\providerutil.jar
```

    Verify the path of these jar files by going to the machine where MQSeries resides and verify the path to each jar file. Any error in these paths will result in numerous "path" specific errors from the JMS Server.

    Set the JDK Path like this:

    ..\jre\bin\java –server

    - MAXRETRY: maximum number of re-tries before a runtime service is shut down, if tokens get rolled back. Default is 5.
    - VMAXERRORS: maximum number of errors returned from validation service. Default is 20.

After this, all pieces of the JMS Server set should be configured properly.

Chapter 3

Installing the Oracle Utilities Transaction Management Application Software

This chapter describes how you install the Oracle Utilities Transaction Management application software, including:

- **Installing the Oracle Utilities Transaction Management Software**
- **Setting Up Configuration Files on the Web Server**
- **Setting Up Configuration Files on Application Server and Workstations**

Installing the Oracle Utilities Transaction Management Software

This section describes how to install the Oracle Utilities Transaction Management software including:

- **Installing in Conjunction with the Energy Information Platform**
- **Installing After Installation of the Energy Information Platform**

Installing in Conjunction with the Energy Information Platform

To install the Oracle Utilities Transaction Management software in conjunction with the Oracle Utilities Energy Information Platform, use the following procedure:

1. Navigate to the Install folder created by the Energy Information Platform installation package (1.6.1.xx.0.EIP.zip). This folder contains the following files:
    - Oracle Utilities EIP 1.6.1.xx.0.msi
    - setup.exe
2. Navigate to the Install folder created by the Oracle Utilities Transaction Management installation package (1.6.1.xx.0.TM.zip). This folders contains the following files:
    - Oracle Utilities TM 1.6.1.xx.0.msi
3. Move the "Oracle Utilities TM 1.6.1.xx.0.msi" file into the same directory as the Energy Information Platform files.

    The directory should now contain following:
    - Oracle Utilities EIP 1.6.1.xx.0.msi
    - setup.exe
    - Oracle Utilities TM 1.6.1.xx.0.msi
4. Double-click the setup.exe file.

    A dialog opens asking you to confirm the products you wish to install. Click **Yes** to continue with the installation. Click **No** to cancel the installation.
5. Proceed with the installation (starting at Step 3) as outlined in **Chapter 4**: **Installing the Oracle Utilities Application Software** in the *Oracle Utilities Energy Information Platform Installation Guide.*

Installing After Installation of the Energy Information Platform

To install the Oracle Utilities Transaction Management software after the Oracle Utilities Energy Information Platform software has been installed, use the following procedure:

1. Navigate to the Install folder created by the Oracle Utilities Transaction Management installation package (1.6.1.xx.0.TM.zip). This folders contains the following files:
    - Oracle Utilities TM 1.6.1.xx.0.msi
2. Double-click the Oracle Utilities TM 1.6.1.xx.0.msi file.

    A dialog opens asking you to confirm the products you wish to install. Click **Yes** to continue with the installation. Click **No** to cancel the installation.
3. Proceed with the installation (starting at Step 3) as outlined in **Chapter 4**: **Installing the Oracle Utilities Application Software** in the *Oracle Utilities Energy Information Platform Installation Guide.*

Setting Up Configuration Files on the Web Server

Oracle Utilities web applications use the following configuration files.

LODESTAR.CFG

The LODESTAR.CFG file is a text file used to customize the working environment of the Oracle Utilities application software. See **LODESTAR.CFG** on page 2-2 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

> **Note:** This file **MUST** be named LODESTAR.CFG on the web server.

LSSECURE.CFG.XML

The LSSECURE.CFG.XML file specifies the data source used by the Security functionality and is required in order to run Oracle Utilities web-enabled applications. The LSSECURE.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSSECURE.CFG.XML** on page 2-42 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

> **Note:** If an application server and web server are installed on different machines, the LSSECURE.CFG.XML file is installed in the C:\LODESTAR\CFG directory with the web server components.

LSREPORTMONITOR.CFG.XML (optional)

The LSREPORTMONITOR.CFG.XML file specifies where report data is stored and how reports are processed through the web-enabled Oracle Utilities Energy Information Platform. The LSREPORTMONITOR.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSREPORTMONITOR.CFG.XML** on page 2-36 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

> **Note:** If an application server and web server are installed on different machines, the LSREPORTMONITOR.CFG.XML file is installed in the C:\LODESTAR\CFG directory on both server.

LSLOGGER.CFG.XML (optional)

The LSLOGGER.CFG.XML file defines how log files are generated by Oracle Utilities applications. Each Oracle Utilities component can be configured to write messages to a specified log file or an Event Log. The LSLOGGER.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSLOGGER.CFG.XML** on page 2-27 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

> **Note:** If an application server and web server are installed on different machines, the LSLOGGER.CFG.XML file is installed in the C:\LODESTAR\CFG directory on both machines.

LSSCHDLR.CFG.XML (optional)

The LSSCHDLR.CFG.XML file defines the data source(s) monitored by the Oracle Utilities Schedule Service (LSSCHDLR.exe). This service monitors the Scheduled Message (SCHEDULEDMESSAGE) table in each data source specified in this file. This file is required if the Oracle Utilities Schedule Service (LSSCHDLR.exe) is used. The LSSCHDLR.CFG.XML must be installed in the **C:\LODESTAR\CFG** directory. See **LSSCHDLR.CFG.XML** on page 2-41 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

> **Note:** If an application server and web server are installed on different machines, the LSSCHDLR.CFG.XML file is installed in the C:\LODESTAR\CFG directory on both machines.

LSRELAY.CFG.XML (optional)

The LSRELAY.CFG.XML file identifies the machine running an SMTP service used to send email messages from the Oracle Utilities Energy Information Platform. This file is required only if the email functions are used. The LSRELAY.CFG.XML file must be installed in the **C:\LODESTAR\CFG** directory. See **LSRELAY.CFG.XML** on page 2-34 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

LOCALES.CFG.XML

The LOCALES.CFG.XML file specifies the regional locales available to Oracle Utilities web-enabled applications. The LOCALES.CFG.XML file must be installed in the **C:\LODESTAR\Web\CCS** directory. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server. See **LOCALES.CFG.XML** on page 2-22 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about this file.

> **Note:** If an application server and web server are installed on different machines, the LOCALES.CFG.XML file is installed in the C:\LODESTAR\Web\ccs directory with the web server components.

LTMH.CFG.XML

The LTMH.CFG.XML file is used to specify the database connection used by the Oracle Utilities Adapter and Oracle Utilities Transaction Management. See **LTMH.CFG.XML** on page 2-46 of the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file.

Setting Up Configuration Files on Application Server and Workstations

You also need to set up the configuration files needed by application server and workstation applications such as Data Manager, batch executables, and the Transaction Management Configuration Workbench.

LODESTAR.CFG

The LODESTAR.CFG file is a text file used to customize the working environment of the Oracle Utilities application software. See **LODESTAR.CFG** on page 2-2 of the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file.

LSRELAY.CFG.XML (optional)

The LSRELAY.CFG.XML file identifies the machine running an SMTP service used to send email messages from the Oracle Utilities Energy Information Platform. This file is required only if the email functions are used. The LSRELAY.CFG.XML file must be installed in the **C:\LODESTAR\CFG** directory on any application server or workstation used to process Rules Language configuration that uses the EMAILCLIENT function. See **LSRELAY.CFG.XML** on page 2-34 in the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file. A sample of this file can be found in the **C:\LODESTAR\CFG\Examples\CFG** directory on the web server.

LTMH.CFG.XML

The LTMH.CFG.XML file is used to specify the database connection used by the Oracle Utilities Adapter and Oracle Utilities Transaction Management. See **LTMH.CFG.XML** on page 2-46 of the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about setting up this file.

Chapter 4

Configuring Oracle Utilities Transaction Management

This chapter describes how you configure Oracle Utilities Transaction Management, including a description of the Oracle Utilities Transaction Management Configuration Workbench application. This application is used to configure the documents, transactions, processes, and business rules used by Oracle Utilities Transaction Management. This chapter includes:

- **Overview**
- **Defining Your Oracle Utilities Transaction Management Configuration**
- **Configuring Oracle Utilities Transaction Management Components**
- **Configuring Oracle Utilities Transaction Management Services**
- **Configuring Oracle Utilities Transaction Management Business Rules**
- **Working with the Configuration Workbench**
- **Other Process and Transaction Configuration Tasks**

Overview

This section describes the process by which you configure Oracle Utilities Transaction Management. The Oracle Utilities Transaction Management Configuration Workbench enables you to create the building blocks necessary to configure an Oracle Utilities Transaction Management process, which can then be used to monitor transactional data flow among your company and its trading partners.

Before you create your first process you must understand the Oracle Utilities Transaction Management vocabulary and the Oracle Utilities Transaction Management Hierarchy. The following is a brief description of the key terms and objects involved when configuring Oracle Utilities Transaction Management.

Process

A process is an Oracle Utilities Transaction Management representation of a real-world business flow. A process is a stored configuration of transactions expected for a particular business process. For example, the enrollment process is a common process in the utility industry. This process may have two transactions, an enrollment request and an enrollment response. An Oracle Utilities Transaction Management process will be visually represented in the Oracle Utilities Transaction Management Web product so the end user will now how many processes are currently open, closed, or in error, as well as at what point (transaction) in the process the each instance is currently in.

Process Instance

A process instance is an actual occurrence of a process. For example, your company may have one enrollment process, but 500 enrollment process instances, one for each account that is actually being enrolled. The Oracle Utilities Transaction Management Web product offers the user a snapshot view of all the process instance information.

Transaction

A transaction is the main building block of a process. In the enrollment process example, the enrollment request and enrollment response are examples of two transactions.

Document

A document defines the format of a given transaction. The source document is the transaction in the format in which it is received from the trading partner. However, to post this transaction to an internal system, the transaction may need to be mapped to another format. For example, to post a monthly usage transaction to Oracle Utilities Billing Component, the transaction must be mapped to the monthly usage document.

Task

A task is processing that occurs against a transaction. Examples of tasks include validation, mapping, or a business rule. There are several Oracle Utilities Transaction Management system tasks that cannot be removed from a transactions flow. These include tasks associated with writing the transaction to the database, identifying the transaction, and attempting to couple the transaction with another transaction. These three tasks occur on all transactions that Oracle Utilities Transaction Management processes.

Map

Maps are XSLT scripts used to transform one XML document into another. A map is associated with a map task for a particular transaction.

Validation Script

Validation scripts are XSLT script used to validate XML documents. Validations can be done through XSLT scripts associated to a Validation task for a transaction, or through a more complex business rule and associated business rule task. Validations done via XSLT scripts usually include valid values, required fields, etc.

Business Rule

An Oracle Utilities Transaction Management business rule is customer-specific operation, performed either through the Oracle Utilities Rules Language, a stored procedure in the Oracle Utilities Data Repository, or another technology that does more complex validations of an XML document that normally cannot be done via an XSL Script. In general, if any database lookups or validations need to occur, a business rule is the preferred approach. Once the business rule is created (following the Oracle Utilities Transaction Management custom business rule standards), the rule is registered via the Oracle Utilities Transaction Management Configuration Workbench and can be added to any XML document processing by associating the business rule task that is created during the registration process to the XML document.

The Oracle Utilities Transaction Management Hierarchy

In a sentence, the Oracle Utilities Transaction Management hierarchy can be described as this: A process is made up of a number of transactions. A transaction can be transformed to any number of documents. A document is processed (validated, mapped, or posted to another system) through the execution of its tasks.

Defining Your Oracle Utilities Transaction Management Configuration

Before you can start setting up the Oracle Utilities Transaction Management processes, transactions, and documents, you should first take some time to define your Oracle Utilities Transaction Management configuration.

Identify Objects

The first step is to define the specific objects, namely processes, transactions, documents, and business rules you need to set up. At this step, you need to identify and define the following:

Required Business Processes: What specific business processes do you need to manage using Oracle Utilities Transaction Management. Some examples include Customer Enrollment, Usage Request, and Dropping a Customer.

Required Transactions to Support Required Business Processes: For each business process you identify, you also need to determine the specific transactions needed to support that process. For example, an Enrollment process might require an Enrollment Request transaction and an Enrollment Response transaction.

Required Documents and Business Rules to Support Required Transactions: For each transaction you identify, you also need to determine the specific documents and/or business rules you need to support that transaction. In addition, you may need to define any validation scripts used to validate transactions and/or maps used to transform content from one transaction format to another. For example, you would need a document that defines the Enrollment Request and Enrollment Request transactions. In addition, you might require a validation script that validates the data in the Request transaction and automatically rejects it if the validation fails. Lastly, you might need to create maps that transform the incoming transaction document into the format used by Oracle Utilities Transaction Management.

Create Documents and Business Rules

After you have defined the required components of your Oracle Utilities Transaction Management configuration, the next step is to create the documents and/or business rules that will be used by the transactions you've defined.

Document Schemas: *.xsd files that define the format of a transaction.

Validation Scripts: *.xsl files used to validate transactions.

Map Scripts: *.xsl files used to transform one document format to another.

Business Rules: custom functionality, implemented as a Oracle Utilities Rules Language configuration, a stored procedure in the Oracle Utilities Data Repository, or some external system.

Configure Objects using the Oracle Utilities Transaction Management Configuration Workbench

The next step is to set up the documents/business rules, transactions, and processes required for your configuration using the Oracle Utilities Transaction Management Configuration Workbench application. The following is an outline of the steps you need to follow to set up these objects up. The section entitled **Working with the Configuration Workbench** on page 4-11 and the sections that follow it provide detailed instructions for performing each step.

Configure External Systems, Queues, and Servers

First, set up any external systems, queues, and/or servers needed for your configuration. This includes:

- **External Systems**: External entities that Oracle Utilities Transaction Management interact (i.e. trading partners)
- **Queues**: Java Messaging Service (JMS) data queues used as middleware between outside systems and Oracle Utilities Transaction Management.
- **Servers**: Oracle Utilities Transaction Management Servers, JMS Servers that interact with Oracle Utilities Transaction Management, and JMS Properties.

See the **Document Menu** on page 4-12 for more information.

Configure Documents

Next, set up the documents required for your configuration, including:

XML Documents: XML schemas used to define transactions in Oracle Utilities Transaction Management.

Validation Scripts: allow you to enforce rules and constraints on the XML data coming into Oracle Utilities Transaction Management. Validation scripts also allow you to can define your own error codes.

Maps: used to transform XML documents from one format to another using XSLT scripts.

See the **Document Menu** on page 4-12 for more information.

Configure Transactions

Next, set up the transactions required for your configuration, including:

Transaction Types: the types of transactions needed.

Transactions: individual transactions. This includes associating an XML document, as well as maps and/or validation scripts to each transaction.

See the **Transactions Menu** on page 4-18 for more information.

Configure Processes

After the documents and transactions are in place, set up the processes required for your configuration, including:

Process Types: the types of transactions needed.

Processes: individual processes. This includes associating one or more transactions to each process, and defining process transaction details and transaction coupling.

See the **Process Menu** on page 4-22 for more information.

Configure Business Rules

Lastly, set up the business rules required for your configuration, including:

Business Rules: individual business rules needed for your configuration. This includes specifying the class name and optional parameters for each.

Business Rule Services: business rule services needed for your configuration. This includes defining specific business rules associated with each service.

See the **Business Rules Menu** on page 4-26 for more information.

Configuring Oracle Utilities Transaction Management Components

Oracle Utilities Transaction Management utilize several components such as servers, Java Messaging Service (JMS) servers, and other components. This section describes how to set up records that define these components in the Oracle Utilities Data Repository, including:

- **Servers**
- **JMS Servers**
- **JMS Queues**
- **Payload Types**
- **Exception Types**
- **Origination Systems**
- **System Properties**
- **Trading Partners**
- **External Transaction Types**
- **External Hubs**
- **External Hub Activation**

Servers

Servers are the application servers which run the Oracle Utilities Transaction Management application. Only application servers with corresponding Server records can run the Oracle Utilities Transaction Management/Oracle Utilities Adapter software.

See **Servers** on page 11-2 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with servers.

JMS Servers

JMS Servers are Java Messaging Service (JMS) compliant components used by Oracle Utilities Transaction Management. JMS Servers are linked to Servers in **System Properties** records. Note that even if JMS Servers are not being used, a placeholder record is required.

See **JMS Servers** on page 11-4 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with JMS servers.

JMS Queues

JMS Queues are message queues used by JMS compliant components used by Oracle Utilities Transaction Management. JMS Queues are associated to services in Runtime Service records.

See **JMS Queues** on page 11-5 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with JMS queues.

Payload Types

Payload types represent different payloads used by Oracle Utilities Transaction Management transactions. Payload types are used by the Oracle Utilities Transaction Management Database Portal to indicate which payload type to monitor.

See **Payload Types** on page 11-7 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with payload types.

Exception Types

Exception types represent different types of custom exceptions that can be generated by Oracle Utilities Transaction Management. Exception types (and their corresponding severity types) are used for grouping and reporting purposes.

See **Exception Types** on page 11-9 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with exception types.

Origination Systems

Origination Systems define the external data sources used by Oracle Utilities Transaction Management, and indicate the system the transaction originated from. Origination Systems are linked to services in Runtime Service records.

See **Origination Systems** on page 11-10 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with origination systems.

System Properties

System Properties represent specific Oracle Utilities Transaction Management servers at runtime, defining how the Oracle Utilities Transaction Management server will perform. System property records also link **Servers** with **JMS Servers** and provide basic information which the server needs to run properly.

See **System Properties** on page 11-12 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with system properties records.

Trading Partners

Trading Partners represent any party to whom Oracle Utilities Transaction Management will send and/or receive transactions.

See **Trading Partners** on page 5-4 in the *Oracle Utilities Transaction Management User's Guide* for information on working with trading partner records.

External Transaction Types

External Transaction Types represent different transaction types used by Oracle Utilities Transaction Management. External Transaction Types tie to the External transaction type that is associated to each Oracle Utilities Transaction Management transaction. The Transaction Identification service will generate an exception if the External Transaction Type associated with an Oracle Utilities Transaction Management transaction is not defined in this table.

See **External Transaction Types** on page 5-7 in the *Oracle Utilities Transaction Management User's Guide* for information on working with external transaction type records.

External Hubs

External Hubs represent external systems to/from which Oracle Utilities Transaction Management sends/receives transactions.

See **External Hubs** on page 5-9 in the *Oracle Utilities Transaction Management User's Guide* for information on working with external hub records.

External Hub Activation

External Hub Activation records provide specifics surrounding the activation of external hubs accessed Oracle Utilities Transaction Management.

See **External Hub Activation** on page 5-10 in the *Oracle Utilities Transaction Management User's Guide* for information on working with external hub activation records.

Configuring Oracle Utilities Transaction Management Services

Services are the basic working components of Oracle Utilities Transaction Management. A service does a single unit of work, a series of which can be chained together in processing transactions. This section describes how to set up records that define these services in the Oracle Utilities Data Repository, including:

- **Runtime Services**
- **Service Activation**
- **XSLT Map Names**
- **XSLT Map Versions**
- **XSLT Maps**

Runtime Services

Runtime Services represent individual Oracle Utilities Transaction Management services. When used with JMS, Runtime Services take input from **JMS Queues**.

See **Runtime Services** on page 11-16 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with runtime services.

Service Activation

Service Activation records define how individual Oracle Utilities Transaction Management services will run on each server. Service Activation records link specific **Servers** to **Runtime Services**.

See **Service Activation** on page 11-20 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with service activation records.

XSLT Map Names

XSLT Map Names are names and descriptions of XML stylesheets (or maps) used by Oracle Utilities Transaction Management. XSLT Maps Names are used to define **XSLT Map Versions**.

See **XSLT Map Names** on page 11-22 in the *Oracle Utilities Energy Information Platform User's Guide* or **Maps** on page 4-16 for information on working with XSLT map names.

XSLT Map Versions

XSLT Map Versions represent specific versions of XML stylesheets (or maps) used by Oracle Utilities Transaction Management. Adapter. XSLT Maps Versions are used to define **XSLT Maps**.

See **XSLT Map Versions** on page 11-23 in the *Oracle Utilities Energy Information Platform User's Guide* or **Maps** on page 4-16 for information on working with XSLT map versions.

XSLT Maps

XSLT Maps are records which store the actual XML stylesheets (or maps) used by Oracle Utilities Transaction Management. XSLT Maps are defined by **XSLT Map Versions**.

See **XSLT Maps** on page 11-26 in the *Oracle Utilities Energy Information Platform User's Guide* or **Maps** on page 4-16 for information on working with XSLT maps.

Configuring Oracle Utilities Transaction Management Business Rules

This section describes how you configure business rules used by Oracle Utilities Transaction Management, including:

- **Global Functions**
- **Rule Description Language**
- **Business Rules**

Global Functions

Global functions are snippets of application code that can be used with multiple Rule Description Language scripts. Each global function is related to a particular Business Rule Type (Import and Import Interval), and is available during configuration of an RDL of the same Business Rule Type.

See **Global Functions** on page 11-50 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with global functions.

Rule Description Language

Rule Description Language (RDL) scripts define specific units of work for business rules. RDL scripts describe what actions will be taken with an incoming payload, such as whether or not to insert particular pieces of data into the Oracle Utilities Data Repository based on a set of validations. RDL scripts are used only with Import, ImportInterval, and Interval Business Rule Types.

See **Rule Description Language** on page 11-53 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with global functions.

Business Rules

Business rules are individual business rules that are available to Oracle Utilities Transaction Management. In specific, they define the business rules that will be applied to data being processed Oracle Utilities Transaction Management.

See **Business Rules** on page 11-58 in the *Oracle Utilities Energy Information Platform User's Guide* for information on working with global functions.

Working with the Configuration Workbench

This section describes how you work with the Oracle Utilities Transaction Management Configuration Workbench, including:

- **Starting the Application**
- **Menus**
- **Common Functions**
- **Document Menu**
- **Transactions Menu**
- **Process Menu**
- **Business Rules Menu**

Starting the Application

To start the Oracle Utilities Transaction Management Configuration Workbench application:

1. Select **Start->Programs->Oracle Utilities->LTMHConfig**.

   The login dialog opens on the desktop.

2. Enter the Data Source, User Name, and Password, and click **OK**.

   The application opens on the desktop.

Menus

You access the Oracle Utilities Transaction Management Configuration Workbench functions and options using the following menus:

- **Document Menu**: used for configuring XML documents, validation scripts, and maps.
- **Transactions Menu**: used for creating transaction types and setting up transactions.
- **Process Menu**: used for creating process types and setting up processes.
- **Business Rules Menu**: used for creating and setting up business rules.

Common Functions

The Oracle Utilities Transaction Management Configuration Workbench screens use a common set of functions, described in the following table.

Button	Description
⚡	**Search** Used to search for records.
➕	**Add** Used to add a new record.
	Edit Used to edit an existing record.

Button	Description
	Save Used to save a new record, or to save changes made to an existing record.
	View Used to view xml document schemas, validation scripts, and map scripts.
	Delete Used to delete records.
	Export Used to export validation scripts and maps.
	Import Used to import validation scripts and maps.

Document Menu

The Document menu allows users to setup and configure documents used by transactions, including:

- **XML Documents**
- **Validation Scripts**
- **Maps**

XML Documents

XML documents are XML schemas used to define transactions in Oracle Utilities Transaction Management.

The XML Documents screen is used to search, add, and edit XML document records, create and edit document version records, and to view document schemas.

How to search XML document records:

1. Select **Documents->XML Documents**. The XML Documents screen opens.
2. Click the **Search** button.

   You can filter the search based on existing XML document names by entering the document name (or the first few letters of a document name) in the XML Document Name field in the Search box before clicking the **Search** button. For instance, if you want to search for all documents whose name begins with the letter ON, you could enter "ON" in the XML Document Name field.

   You can also use the percent sign ("%") to search for any document that contains a letter or string of letters. For instance, if you want to search for all documents whose name contains the letters INV (as in Invoice), you could enter "%INV" in the XML Document Name field.
3. The search results appear in the XML Documents box.

How to add new XML document records:

1. Click the **Add** button in the XML Documents box. The **Add XML Document** dialog opens.
2. Enter the **XML Document Name** and a **Document Description**.
3. Click the **Save** button. The new document record appears in the XML Documents box.

How edit existing XML document records:

1. Select the XML document you wish to edit.
2. Click the **Edit** button in the XML Documents box. The **Edit XML Document** dialog opens.
3. Edit the **XML Document Name** and a **Document Description** as appropriate.
4. Click the **Save** button. The updated document record appears in the XML Documents box.

How to add new document versions:

1. Select the XML document that you wish to add a version to. A list of existing versions appears in the Versions box.
2. Click the **Add** button in the Versions box. The **Add XML Document Version** dialog opens.
3. Enter the **Version Description** and the **Schema URL** for the document version. To view the schema, click the **View** button. See **How to view XML document version schemas:** on page 4-13 for more information.
4. Enter the **Major Version** number and the **Minor Version** number, and select the **Begin Date** and **End Date** for the version. You can enter the dates either by typing the date manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.
5. Click the **Save** button. The new document version appears in the Versions box.

How to edit existing document versions:

1. Select the XML document that corresponds to the version you wish to edit. A list of existing versions appears in the Versions box.
2. Select the document version you wish to edit from the list in the Versions box.
3. Click the **Edit** button in the Versions box. The **Edit XML Document Version** dialog opens.
4. Edit the **Version Description** and the **Schema URL** for the document version as appropriate. To view the schema, click the **View** button. See **How to view XML document version schemas:** on page 4-13 for more information.
5. Edit the **Major Version** number and the **Minor Version** number and change the **Begin Date** and **End Date** for the version as appropriate. You can enter the dates either by typing the date manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.
6. Click the **Save** button. The updated document version appears in the Versions box.

How to view XML document version schemas:

1. Select the XML document that corresponds to the version schema you wish to view. A list of existing versions appears in the Versions box.
2. Select the document version that corresponds to the version schema you wish to view from the list in the Versions box.
3. Click the **View** button. The **Document Version Schema** window opens.

Validation Scripts

Validation scripts allow you to enforce rules and constraints on the XML data coming into Oracle Utilities Transaction Management. Validation scripts also allow you to can define your own error codes. See **Appendix B**, **Oracle Utilities Transaction Management Validation Scripts**, for more information.

The Validation Script Setup screen is used to search, add, and edit validation script records, create and edit validation script version records, and to build, view, export, and import scripts.

How to search validation script records:

1. Select **Documents->Validation Scripts**. The Validation Script Setup screen opens.
2. Click the **Search** button.

   You can filter the search based on existing validation script and/or associated xml documents by entering the script name (or the first few letters of a script name) in the Validation Script field and/or the document name (or the first few letters of the document name) in the XML Document Name field in the Search box before clicking the **Search** button. For instance, if you want to search for all scripts whose name begins with the letter ON, you could enter "ON" in the Validation Script field.

   You can also use the percent sign ("%") in either field to search for any script that contains a letter or string of letters. For instance, if you want to search for all scripts whose name contains the letters INV (as in Invoice), you could enter "%INV" in the Validation Script field.
3. The search results appear in the Validation Scripts box.

How to add new validation script records:

1. Click the **Add** button in the Validation Scripts box. The **Add Validation Script** dialog opens.
2. Enter the **Validation Script Name**.
3. Select the **XML Document** to be associated with the validation script from the **XML Document Name** drop-down list.
4. Enter a **Description** for the record.
5. Click the **Save** button. The new validation script record appears in the Validation Scripts box.

How to edit existing validation script records:

1. Select the validation script you wish to edit.
2. Click the **Edit** button in the Validation Scripts box. The **Update XML Validation Script Details** dialog opens.
3. Enter the **Validation Script Name** as appropriate.
4. Select the **XML Document** to be associated with the validation script from the **XML Document Name** drop-down list.
5. Edit the **Description** for the record as appropriate.
6. Click the **Save** button. The updated validation script record appears in the Validation Scripts box.

How to add new validation script versions:

1. Select the Validation Script that you wish to add a version to. A list of existing versions appears in the Versions box.
2. Click the **Add** button in the Versions box. The **Add New Validation Script Version** dialog opens.
3. Enter the **Major Version** number and the **Minor Version** number, and select the **Begin Date** and **End Date** for the version. You can enter the dates either by typing the date

manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.

4. Enter an optional note for the version.
5. In the XML Document box, select the XML Document version from the **Document Version** drop-down list.
6. Click the **Save** button. The new version appears in the Versions box.

How to edit existing validation script versions:

1. Select the Validation Script that corresponds to the version you wish to edit. A list of existing versions appears in the Versions box.
2. Select the validation script version you wish to edit from the list in the Versions box.
3. Click the **Edit** button in the Versions box. The **Edit Script Version** dialog opens.
4. Change the **Major Version** number and the **Minor Version** number, and change the **Begin Date** and **End Date** for the version as appropriate. You can enter the dates either by typing the date manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.
5. Enter an optional note for the version, or edit an existing note if applicable.
6. In the XML Document box, select the XML Document version from the **Document Version** drop-down list.
7. Click the **Save** button. The updated version appears in the Versions box.

How to view and/or edit validation scripts:

1. Select the Validation Script that corresponds to the script you wish to view. A list of existing versions appears in the Versions box.
2. Select the validation script version that corresponds to the script you wish to view from the list in the Versions box.
3. Click the **View** button in the Versions box. The **Script Viewer** window opens.
4. Edit the script as appropriate.
5. Click the **Save** button. The script is saved and can be viewed.

How to export a validation script:

1. Select the Validation Script that corresponds to the script you wish to export. A list of existing versions appears in the Versions box.
2. Select the validation script version that corresponds to the script you wish to export from the list in the Versions box.
3. Click the **Export** button in the Versions box. The **Export Validation Script** window opens.
4. Click the browse (...) button. The **Save As** dialog opens.
5. Navigation to the destination directory and enter a name (exported scripts should be *.XSL files) in the **File Name** field and click **Open**. The specified path and file name will appear in the **Validation Script File Name** field on the **Export Validation Script** window.
6. Click the **Save** button to export the script to the specified destination directory.

How to import a validation script:

1. Select the Validation Script that you wish to import the script to. A list of existing versions appears in the Versions box.
2. Select the validation script version that you wish to import the script to from the list in the Versions box.

3. Click the **Import** button in the Versions box. The **Import Validation Script** window opens.
4. Click the browse (...) button. The **Open** dialog opens.
5. Navigation to the location of the script file you wish to import and click **Open**. The specified path and file name will appear in the **Validation Script File Name** field on the **Import Validation Script** window.
6. Click the **Save** button to import the script.

Maps

Maps are used to transform XML documents from one format to another using XSLT scripts.

The Map Setup screen is used to search, add, and edit map records, create and edit map version records, and to build XSL map scripts.

How to search map records:

1. Select **Documents->Maps**. The Map Setup screen opens.
2. Click the **Search** button.

   You can filter the search based on existing maps by entering the map name (or the first few letters of a map name) in the Map Name field in the Search box before clicking the **Search** button. For instance, if you want to search for all maps whose name begins with the letter ON, you could enter "ON" in the Map Name field.

   You can also use the percent sign ("%") in either field to search for any map that contains a letter or string of letters. For instance, if you want to search for all maps whose name contains the letters INV (as in Invoice), you could enter "%INV" in the Map Name field.
3. The search results appear in the Available Maps box.

How to add new map records:

1. Click the **Add** button in the Available Maps box. The **Map Detail** dialog opens.
2. Enter the **Map Name** and **Description**.
3. Click the **Save** button. The new map record appears in the Available Maps box.

How edit existing map records:

1. Click the **Edit** button in the Available Maps box. The **Map Detail** dialog opens.
2. Edit the **Map Name** and **Description** as appropriate.
3. Click the **Save** button. The updated map record appears in the Available Maps box.

How to add new map versions:

1. Select the Map that you wish to add a version to. A list of existing versions appears in the Map Versions box.
2. Click the **Add** button in the Map Versions box. The **Map Version Setup** dialog opens.
3. Enter the **Major Version** number and the **Minor Version** number, and select the **Begin Date** and **End Date** for the version. You can enter the dates either by typing the date manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.
4. Enter an optional note for the version.
5. Click the **Save** button. The new version appears in the Map Versions box.

How to edit existing map versions:

1. Select the Map that corresponds to the version you wish to edit. A list of existing versions appears in the Map Versions box.
2. Select the map version you wish to edit from the list in the Map Versions box.
3. Click the **Edit** button in the Map Versions box. The **Map Version Setup** dialog opens.
4. Change the **Major Version** number and the **Minor Version** number, and change the **Begin Date** and **End Date** for the version as appropriate. You can enter the dates either by typing the date manually, or by clicking the drop-down arrow to the right of the date field. This opens a calendar dialog that you can use to select the dates.
5. Enter an optional note for the version or edit an existing note as appropriate.
6. Click the **Save** button. The updated version appears in the Map Versions box.

How to view and edit existing map scripts:

1. Select the map record that corresponds to the script you wish to view. A list of existing versions appears in the Map Versions box.
2. Select the map version corresponds to the script you wish to view from the list in the Map Versions box.
3. Click the **View** button in the Versions box. The **Mapping Script** window opens.
4. Edit the script as appropriate.
5. Click the **Save** button. The script is saved and can be viewed/edited as needed.

How to export a map script:

1. Select the map record that corresponds to the map script you wish to export. A list of existing versions appears in the Versions box.
2. Select the map version that corresponds to the map script you wish to export from the list in the Versions box.
3. Click the **Export** button in the Versions box. The **Export Map** window opens.
4. Click the browse (...) button. The **Save As** dialog opens.
5. Navigation to the destination directory and enter a name (exported maps should be *.XSL files) in the **File Name** field and click **Open**. The specified path and file name will appear in the **Map File Name** field on the **Export Map** window.
6. Click the **Save** button to export the map script to the specified destination directory.

How to import a map script:

1. Select the Map record that you wish to import the map script to. A list of existing versions appears in the Versions box.
2. Select the map version that you wish to import the map script to from the list in the Versions box.
3. Click the **Import** button in the Versions box. The **Import a map** window opens.
4. Click the browse (...) button. The **Open** dialog opens.
5. Navigation to the location of the map script you wish to import and click **Open**. The specified path and file name will appear in the **XML Map File Name** field on the **Import a map** window.
6. Click the **Save** button to import the map.

Transactions Menu

The Transactions menu allows users to set up transactions and define transaction types and includes the following options:

- **Transaction Types**
- **Transaction Setup**

Transaction Types

The Transaction Type screen allows you to define the types of transactions and their priorities.

How to add new transaction type records:

1. Select **Transactions->Transaction Types**. The Transaction Type screen opens.
2. Click the **Add** button.
3. In the Transaction Type Setup box (the bottom portion of the screen), enter the **Transaction Type** and set the **Priority** for the transaction type.
4. Click the **Save** button. The new transaction type appears in the Transaction Type box.

How to edit existing transaction type records:

1. Select the transaction type you wish to edit. The Transaction Type Setup box displays the details of the selected transaction type.
2. In the Transaction Type Setup box (the bottom portion of the screen), edit the **Transaction Type** and change the **Priority** for the transaction type as appropriate.
3. Click the **Save** button. The updated transaction type appears in the Transaction Type box.

Transaction Setup

In many cases a business process may consists of multiple transactions. For example, an enrollment may consists of an enrollment request, a response, and usage. The Transaction Search and Setup screens are used to define the transactions that make up business processes.

How to search transaction records:

1. Select **Transactions->Transaction Setup**. The Transaction Search screen opens.
2. Click the **Search** button.

   You can filter the search based on existing transaction names by entering the transaction name (or the first few letters of a transaction name) in the Transaction Name field in the Search box before clicking the **Search** button. For instance, if you want to search for all transactions whose name begins with the letter ON, you could enter “ON” in the Transaction Name field.

   You can also use the percent sign (“%”) to search for any transaction that contains a letter or string of letters. For instance, if you want to search for all transactions whose name contains the letters INV (as in Invoice), you could enter “%INV” in the Transaction Name field.

   You can also filter the search based on Transaction Types by selected the appropriate transaction type from the Transaction Type drop-down list.
3. The search results appear in the Transactions box.

How to add new transaction records:

1. Click the **Add** button in the Transactions box. The **Transaction Detail** dialog opens.
2. Enter the **Transaction Name**.
3. Select the appropriate transaction type from the **Transaction Type** drop-down list.
4. Select the appropriate rejection transaction from the **Rejection Transaction** drop-down list.
5. Click the **Save** button. The new transaction record appears in the Transactions box.

How to edit existing transaction records:

1. Select the transaction you wish to edit in the Transactions box.
2. Click the **Edit** button in the Transactions box. The **Transaction Detail** dialog opens.
3. Edit the **Transaction Name** as appropriate.
4. Select the appropriate transaction type from the **Transaction Type** drop-down list.
5. Select the appropriate rejection transaction from the **Rejection Transaction** drop-down list.
6. Click the **Save** button. The updated transaction record appears in the Transactions box.

How to view and set transaction identification:

1. Select the transaction that you wish to view/setup transaction identification for in the Transactions box.
2. Click the **Transaction Identification** button in the Transactions box. The **Transaction Identification** dialog opens.

   There are four attributes used in transaction identification:

   - **Jurisdiction** (JURISCODE): The Jurisdiction associated with the transaction.
   - **External Transaction Type** (EXTTRANSTYPE): The external transaction type associated with the transaction.
   - **Original Transaction Type** (ORIGTRANSTYPE): The original transaction type associated with the transaction.
   - **Status Advice Type** (STATUSADVICETYPE): The status advice type associated with the transaction.

   Each of these attributes can have a value that is used by the Transaction Identification runtime service to determine the type and context of the transaction.

3. Select the JURISCODE attribute and type the Jurisdiction code in the **Value** field.
4. Repeat for each of the remaining attributes as appropriate.
5. Click the **Save** button. The transaction identification is set for the transaction.

How to set up new transactions:

1. Select the transaction that you wish to setup in the Transactions box.
2. Click the **Setup** button in the Transactions box. The **<transaction name> Documents** dialog opens.
3. Select the XML Document for the root transaction from the **XML Document** drop-down list.
4. Select an optional XML Map for the root transaction from the **XML Map** drop-down list.
5. Click the **Add Root** button. The transaction appears in the transaction tree box on the right of the window.

6. *Optional*. To add child transactions to a transaction, select the transaction, select the XML Document for the child transaction from the **XML Document** drop-down list, and the XML Map for the child transaction from the **XML Map** drop-down list.

   Click the **Add Child** button. The child transaction appears beneath the selected transaction in the transaction tree box on the right of the window.

7. *Optional*. To add sibling transactions to a child transaction, select the child transaction, select the XML Document for the sibling transaction from the **XML Document** drop-down list, and the XML Map for the sibling transaction from the **XML Map** drop-down list.

   Click the **Add Sibling** button. The sibling transaction appears beneath the selected child in the transaction tree box on the right of the window.

8. *Optional*. To add tasks to a transaction, select the transaction and click the **Task Details** button. The **Document Validations and Estimations** window opens. This window allows you to add validations, business rules, and destinations to selected transactions.

   Validations: To add (or edit or delete) validation scripts to the transaction:

   - Select the validation script from the **Validation** drop-down list in the Validations box.
   - Select the auto-response map from the **Auto Response Map** drop-down list in the Validations box.
   - Select the auto-response validation from the **Auto Response Validation** drop-down list in the Validations box.
   - Select the auto-response destination from the **Auto Response Destination** drop-down list in the Validations box.
   - Click the **Add** (or **Edit** or **Delete**)button in the Validations box. The validation script appears in the Validations box.

   Business Rules: To add (or edit or delete) business rules to the transaction:

   - Select the business rule from the **Business Rule** drop-down list in the Business Rules box.
   - Select the auto-response validation from the **Auto Response Validation** drop-down list in the Business Rules box.
   - Select the auto-response map from the **Auto Response Map** drop-down list in the Business Rules box.
   - Select the auto-response destination from the **Auto Response Destination** drop-down list in the Business Rules box.
   - Click the **Add** (or **Edit** or **Delete**) button in the Business Rules box. The validation script appears in the Business Rules box.

   Destinations: To add destinations to the transaction:

   - Select the destination from the left pane in the Destinations box.
   - Click the right arrow button (**>**). The selected destination appears in the right pane in the Destinations box.

9. *Optional*. To view the task tree for a transaction, select the transaction and click the **View Task Tree** button. The window expands to display the Document Tasks box, which contains a task tree for the selected transaction.

   To close the Document Tasks window, click the **Close Tree View** button.

How to view and edit transactions:

1. Select the transaction that you wish to edit in the Transactions box.
2. Click the **Setup** button in the Transactions box. The **<transaction name> Documents** dialog opens.
3. *Optional.* To add child transactions to a transaction, select the transaction, select the XML Document for the child transaction from the **XML Document** drop-down list, and the XML Map for the child transaction from the **XML Map** drop-down list.

   Click the **Add Child** button. The child transaction appears beneath the selected transaction in the transaction tree box on the right of the window.
4. *Optional.* To add sibling transactions to a child transaction, select the child transaction, select the XML Document for the sibling transaction from the **XML Document** drop-down list, and the XML Map for the sibling transaction from the **XML Map** drop-down list.

   Click the **Add Sibling** button. The sibling transaction appears beneath the selected child in the transaction tree box on the right of the window.
5. *Optional.* To add tasks to a transaction, select the transaction and click the **Task Details** button. The **Document Validations and Estimations** window opens. This window allows you to add validations, business rules, and destinations to selected transactions.

   Validations: To add (or edit or delete) validation scripts to the transaction:

   - Select the validation script from the **Validation** drop-down list in the Validations box.
   - Select the auto-response map from the **Auto Response Map** drop-down list in the Validations box.
   - Select the auto-response validation from the **Auto Response Validation** drop-down list in the Validations box.
   - Select the auto-response destination from the **Auto Response Destination** drop-down list in the Validations box.
   - Click the **Add** (or **Edit** or **Delete**) button in the Validations box. The validation script appears in the Validations box.

   To delete validation scripts from the transaction:

   - Select the validation script to be deleted from the list in the Validations box.
   - Click the **Delete** button in the Validations box. The validation script is removed from the Validations box.

   Business Rules: To add (or edit or delete) business rules to the transaction:

   - Select the business rule from the **Business Rule** drop-down list in the Business Rules box.
   - Select the auto-response validation from the **Auto Response Validation** drop-down list in the Business Rules box.
   - Select the auto-response map from the **Auto Response Map** drop-down list in the Business Rules box.
   - Select the auto-response destination from the **Auto Response Destination** drop-down list in the Business Rules box.
   - Click the **Add** (or **Edit** or **Delete**) button in the Business Rules box. The validation script appears in the Business Rules box.

   To delete business rules from the transaction:

   - Select the business rule to be deleted from the list in the Business Rules box.

- Click the **Delete** button in the Business Rules box. The business rule is removed from the Business Rules box.

Destinations: To add destinations to the transaction:

- Select the destination from the left pane in the Destinations box.
- Click the right arrow button (>). The selected destination appears in the right pane in the Destinations box.

To remove destinations from the transaction:

- Select the destination to be removed from the right pane in the Destinations box.
- Click the left arrow button (<). The selected destination is removed from the right pane in the Destinations box and appears in the left pane.

6. *Optional.* To view the task tree for a transaction, select the transaction and click the **View Task Tree** button. The window expands to display the Document Tasks box, which contains a task tree for the selected transaction.

   To close the Document Tasks window, click the **Close Tree View** button.

Process Menu

The Process menu allows users to set up processes and define their types and includes the following options:

- **Process Types**
- **Process Setup**

Process Types

The Process Types screen allows you to define process types.

How to add new process type records:

1. Select **Process->Process Types**. The Process Types screen opens.
2. Click the **Add** button.
3. In the Process Type Edit box (the bottom portion of the screen), enter the **Process Type** and **Process Description** for the process type.
4. Click the **Save** button. The new process type appears in the Process Types box.

How to edit existing process type records:

1. Select the process type record you wish to edit. The Process Type Edit box displays the details of the selected process type.
2. In the Process Type Edit box (the bottom portion of the screen), edit the **Process Type** and **Process Description** for the process type as appropriate.
3. Click the **Save** button. The updated process type appears in the Process Types box.

Process Setup

The Process Search and Setup screens allow you to search and define processes.

How to search process records:

1. Select **Process->Process Setup**. The Process Setup screen opens.
2. Click the **Search** button.

   You can filter the search based on existing process names by entering the process name (or the first few letters of a process name) in the Process Name field in the Search box before clicking the **Search** button. For instance, if you want to search for all processes whose name begins with the letter ON, you could enter "ON" in the Process Name field.

You can also use the percent sign ("%") to search for any process that contains a letter or string of letters. For instance, if you want to search for all processes whose name contains the letters INV (as in Invoice), you could enter "%INV" in the Process Name field.

You can also filter the search based on Process Types by selected the appropriate transaction type from the Process Type drop-down list.

3. The search results appear in the Process box.

How to add new process records:

1. Click the **Add** button in the Process box. The **Process** dialog opens.
2. Enter the **Process Name**.
3. Select the appropriate process type from the **Process Type** drop-down list.
4. Enter a description for the process.
5. Click the **Save** button. The new process record appears in the Process box.

How to edit existing process records:

1. Select the process you wish to edit.
2. Click the **Edit** button in the Process box. The **Process** dialog opens.
3. Edit the **Process Name** as appropriate.
4. Select the appropriate process type from the **Process Type** drop-down list.
5. Enter a description for the process or edit the description as appropriate.
6. Click the **Save** button. The updated process record appears in the Process box.

How to setup process transactions:

1. Select the process that you wish to setup in the Process box. The Process Transaction box at the bottom of the screen will be blank.
2. Click the **Add Root** button to add a root transaction for the process. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

   Click the **Save** button to save the transaction. The transaction will appear in the Process Transactions box.
3. *Optional.* To add child transactions to a transaction, select the transaction and click the **Add Child** button. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Optional. If the transaction will be used as a notify transaction, check the **Notify** checkbox

   Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

   Click the **Save** button to save the transaction. The transaction will appear beneath the selected transaction in the Process Transactions box.

4. *Optional.* To add sibling transactions to a child transaction, select the child transaction, and click the **Add Sibling** button. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Optional. If the transaction will be used as a notify transaction, check the **Notify** checkbox

   Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

   Click the **Save** button to save the transaction. The transaction will appear beneath the selected transaction in the Process Transactions box.

5. *Optional.* To setup transaction coupling between a child transaction and its parent, click the **Transaction Coupling** button. The **Define Coupling Elements** dialog opens.

   To define coupling elements for the transactions:

   - Select the element to be coupled from the **Parent** list in the Transaction Coupling Elements box (top of the dialog).
   - Select the element from the **Current** list in the Transaction Coupling Elements box that you wish to couple to the element from the **Parent** list.
   - Click the **Add** button in the Transaction Coupling Elements box. The selected pair of elements appear in the Currently Defined Coupling Elements box (bottom of the dialog).
   - Click the “X” in the upper right corner of the dialog to close the window.

6. *Optional.* To view details concerning a transaction, select the transaction and click the **View Transaction Details** button. The **<transaction name> Documents** dialog opens. See **How to view and edit transactions:** on page 4-21 for information about viewing and editing transaction details.

How to view and edit process transactions:

1. Select the process that you wish to view/edit in the Process box. The select process transaction will appears in the Process Transaction box at the bottom of the screen.

2. *Optional.* To edit a transaction, select the transaction and click the **Edit** button. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Optional. If the transaction will be used as a notify transaction, check the **Notify** checkbox

   Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

   Click the **Save** button to save the transaction. The transaction will appear beneath the selected transaction in the Process Transactions box.

3. *Optional.* To add child transactions to a transaction, select the transaction and click the **Add Child** button. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Optional. If the transaction will be used as a notify transaction, check the **Notify** checkbox

Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

Click the **Save** button to save the transaction. The transaction will appear beneath the selected transaction in the Process Transactions box.

4. *Optional.* To add sibling transactions to a child transaction, select the child transaction, and click the **Add Sibling** button. The **Process Transaction** dialog opens.

   Select the appropriate transaction from the **Transaction** drop-down list.

   Optional. If the transaction is optional, check the **Optional** checkbox, and select a receive status code from the **Receive Status Code** drop-down list.

   Optional. If the transaction will be used as a notify transaction, check the **Notify** checkbox

   Enter soft/hard expiration durations in the **Soft Expiration Days** and **Hard Expiration Days** fields.

   Click the **Save** button to save the transaction. The transaction will appear beneath the selected transaction in the Process Transactions box.

5. *Optional.* To setup or change transaction coupling between a child transaction and its parent, click the **Transaction Coupling** button. The **Define Coupling Elements** dialog opens.

   To define coupling elements for the transactions:

   - Select the element to be coupled from the **Parent** list in the Transaction Coupling Elements box (top of the dialog).
   - Select the element from the **Current** list in the Transaction Coupling Elements box that you wish to couple to the element from the **Parent** list.
   - Click the **Add** button in the Transaction Coupling Elements box. The selected pair of elements appear in the Currently Defined Coupling Elements box (bottom of the dialog).
   - Click the “X” in the upper right corner of the dialog to close the window.

   To delete coupling elements for the transactions:

   - Select the pair of elements to be deleted in the Currently Defined Coupling Elements box.
   - Click the **Delete** button in the Transaction Coupling Elements box. The selected pair of elements is removed from the Currently Defined Coupling Elements box.

6. *Optional.* To delete a transaction, select the transaction and click the **Delete** button. You cannot delete transactions that have child transactions beneath them.

7. *Optional.* To view details concerning a transaction, select the transaction and click the **View Transaction Details** button. The **<transaction name> Documents** dialog opens. See **How to view and edit transactions:** on page 4-21 for information about viewing and editing transaction details.

Business Rules Menu

The Business Rules menu allows you to set up specific business rules in Oracle Utilities Transaction Management, and includes the following options:

- **Business Rule Service**

Business Rule Service

The Business Rule Service screen allows you to set up business rule services and configure the rules for those services.

How to add new business rule service records:

1. Select **Business Rules-›Business Rule Service**. The Business Rule Service screen opens.
2. Click the **Add** button.
3. In the Transaction Type Setup box (the bottom portion of the screen), enter the **Business Rule Name**.
4. Click the **View Service Rules** button. The **Rules for Business Rule Service** window opens. This window allows you to associate one or more business rules with the business rule service.
5. The left pane displays the available business rules (created using the **Business Rules** screens). The right pane is blank. Select the business rule(s) applicable for the business rule service and click the right arrow button. The selected business rule(s) moves to the right pane.

   To move a business rule(s) from the right pane to the left pane, select the business rule(s) to move and click the left arrow button. The selected business rule(s) moves to the left pane.
6. Click the “X” in the upper right corner of the Rules for Business Rule Service window to close the window.
7. Click the **Save** button. The new business service appears in the Available Rules box.

How to edit existing business rule service records:

1. Select the business rule service you wish to edit in the Transaction Type box. The Transaction Type Setup box displays the details of the business rule service.
2. In the Transaction Type Setup box (the bottom portion of the screen), edit the **Business Rule Name** as appropriate.
3. To change the service rules, click the **View Service Rules** button. The **Rules for Business Rule Service** window opens. This window allows you to associate one or more business rules with the business rule service.
4. The left pane displays the available business rules (created using the **Business Rules** screen). The right pane is blank. Select the business rule(s) applicable for the business rule service and click the right arrow button. The selected business rule(s) moves to the right pane.

   To move a business rule(s) from the right pane to the left pane, select the business rule(s) to move and click the left arrow button. The selected business rule(s) moves to the left pane.
5. Click the “X” in the upper right corner of the Rules for Business Rule Service window to close the window.
6. Click the **Save** button. The updated business service appears in the Available Rules box.

Other Process and Transaction Configuration Tasks

This section outline a number of specific configuration tasks related to configuring transaction management processes and transactions, including:

- **Configuring Process Notification**
- **Editing Transaction Instance Header Attributes**
- **Understanding Transformation and Validation**

Configuring Process Notification

Within Oracle Utilities Transaction Management, processes can be joined through the use of "notify" transactions. This is a situation in which one process notifies another process that a transaction has been received. For example, a final usage transaction may be used by a usage process, but also by a billing process to provide a final bill to the customer, as well as a drop process to drop the customer. This section outlines how to configure process notifications of this type.

How to configure a "notify" transaction for a process:

1. Select **Process->Process Setup**. The Process Setup screen opens (see **Process Setup** on page 4-22 for more information about working with the Process Setup screen).
2. Select the process to which you wish to add a notification (that is, the process which will be notified) in the Process Box.
3. Select the transaction to which you wish to add the "notify" transaction in the Transaction box.
4. Click **Add Child**.
5. Select the "notify" transaction (this is a process transaction on the notifying process).
6. Check the **Notify** checkbox on the "notify" transaction.
7. Click **OK**.
8. Click **Edit** to adjust settings of the "notify" transaction if needed.
9. Click **Transaction Coupling** and define the fields in each transaction that must be equal in order for the notification to occur.

Example

Currently you have defined 2 processes (Usage & Billing), and you wish to use the Usage process to notify the Billing process when transactions have been received. In this case, you would add a usage transaction from the Usage process as a "notify" transaction to the Billing process.

Editing Transaction Instance Header Attributes

The LTMH Transaction table can be extended to support custom header attributes as needed to meet customer-specific requirements. This section describes the steps involved in changing the Header attributes of transactions in the LTMH Transaction table.

How to add new Transaction Header Attributes:

1. Add the new columns to the LTMH Transaction (HUBTRANSINSTANCE) table and to the Column Metadata table.

   Note: Changes to the Oracle Utilities Data Repository schema should be performed by Oracle Professional Services staff.

2. Get the Token schema from the HUBTOKENSCHEMA table and update the “LTMH Token Schema HeaderFields” section to include the new fields.

   By default, this section of the schema is as follows:

```
<xsd:element name="HEADERFIELDS">
  <xsd:complexType>
    <xsd:attribute name="JURISCODE" type="xsd:string"/>
    <xsd:attribute name="EXTTRANSTYPE" type="xsd:string"/>
    <xsd:attribute name="TRANSID" type="xsd:string"/>
    <xsd:attribute name="DATESENT" type="xsd:string"/>
    <xsd:attribute name="EXTERNALACCTNUM" type="xsd:string"/>
    <xsd:attribute name="INTERNALACCTNUM" type="xsd:string"/>
    <xsd:attribute name="SENDERID" type="xsd:string"/>
    <xsd:attribute name="RECEIVERID" type="xsd:string"/>
    <xsd:attribute name="CLEARINGHOUSEID" type="xsd:string"/>
    <xsd:attribute name="FROMSYSTEM" type="xsd:string"/>
    <xsd:attribute name="ORIGINALTRANSID" type="xsd:string"/>
    <xsd:attribute name="DUNS" type="xsd:string"/>
    <xsd:attribute name="POSTALCODE" type="xsd:string"/>
    <xsd:attribute name="TRANSXREF" type="xsd:string"/>
    <xsd:attribute name="PREVEXTACCTNUM" type="xsd:string"/>
    <xsd:attribute name="ORIGTRANSTYPE" type="xsd:string"/>
    <xsd:attribute name="STATUSADVICETYPE" type="xsd:string"/>
    <xsd:attribute name="ACCTVALIDATOR" type="xsd:string"/>
    <xsd:attribute name="NAMEVALIDATOR" type="xsd:string"/>
  </xsd:complexType>
</xsd:element>
```

3. Insert a new record for the new attribute into the XML Attribute (XMLATTRIBUTE) table. This table contains the following columns:

Column Name	Description
UIDXMLATTRIBUTE	Key to table - sequence
UIDXMLELEMENT	1
ATTRIBUTENAME	Name of attribute. Should be the same as the new column name in the LTMH Transaction table
ATTRIBUTEDESC	Attribute Description
DISPLAYNAME	The name to be displayed for the attribute
LSCOLUMN	Y
LSUIDCOLUMN	Reference to the UIDCOLUMN column in the Column Metadata table
DATATYPECD	S
IDCOLUMN	Y if used for transaction identification, N otherwise

Understanding Transformation and Validation

Oracle Utilities Transaction Management maintains an internal representation of documents flowing inbound and outbound. XSLT is used to transform inbound documents into the internal representation. Likewise, validation via a XML schema is also performed to check the outputted structure. On the outbound side, documents are assembled and transformed back into customer-specific format, again using XSLT. For more specific information and syntax on schemas and XSLT, please see various internet sites, such as www.w3schools.com or www.devguru.com.

Transformation

Because Oracle Utilities Transaction Management must be able to function with a variety of input structures and across diverse markets, XSLT is used to convert customer information into a predictable format so that operations may be performed on it. When sending outbound documents back to customers, a separate XSLT converts the information back into a format the recipient will understand.

Our schemas (described below) define the structure Oracle Utilities Transaction Management is expecting. Inbound transformation takes place prior to any parsing of the document. Because the Inbound Gateway can not know whether it is receiving a Functional Acknowledgement or a document containing one or more transactions, it must utilize a transformation that can handle either case. Converting a Functional Acknowledgement to internal format is fairly straightforward. Transaction documents are slightly more complicated. Advanced processing, such as looking up non-included data in a database, and "extension libraries" may be used. Use of extensions is by no means required for normal operation, but can provide very powerful options for users. More information can be found at http://xml.apache.org/xalan-j/extensionslib.html. These documents contain one or more business transactions, each of which must be copied in its entirety into the PAYLOAD node.

Since the Outbound Gateway can predict the type of document it is sending, it is not necessary to include provisions for both Functional Acknowledgements and transaction documents in the same XSLT file. The purpose of these translations is to return the outgoing data to a format the customer can understand. The transformations of both Functional Acknowledgements and transaction documents are both direct translations. The "Reason Codes" that can be expected within a Functional Acknowledgement are:

Reason Code	Description
UNKNOWN	An unspecified error was detected
DUPDOCREFID	The incoming document had a duplicate DOCREFID
DUPTRANSREFID	The TRANSREFID attribute had already been used in the system
VALIDATIONERROR	An error occurred during schema validation
INVALIDRECEIVER	The receiver specified in the RECEIVERID does not match the operating trading partner
INVALIDSENDER	The sender specified in the SENDERID does not match the operating trading partner (in the case of FileLoader's TPID and Inbound Gateway's TPID being the same) or the trading partner specified by FileLoader.
MISSINGTRANSACTION	The expected number of transactions within a document could not be found. This results from a comparison of the TRANSACTIONCOUNT attribute in TRANSCOLLECTION and the actual number of TRANSACTIONs within the collection. Note that TRANSACTIONCOUNT is an optional attribute and calculating the count accurately could be tricky.

Validation

Oracle Utilities Transaction Management includes starting points for the schemas that will validate incoming documents and functional acknowledgements. The schema that defines the structure of incoming transactions can by default be found in the **\\lodestar\ltmh\LTMHRuntime\schemas** directory and is called TransCollection.xsd. This file will need to be modified to accommodate incoming data. There are several points within the schema that are marked with comments. These comments explain what to add at a given point.

- The first point is to change the "market" namespace to that of the incoming documents. The use of namespaces allows separate control over the Oracle Utilities Transaction Management XML structure and the customer's XML structure.
- The next point of modification is the import statement for the customer's schema file. The namespace attribute must again be set as well as the file location of the document. Individual customer transactions will be copied into the result. This import statement will verify the content of each.
- The final required change is to define the outer-most node of a transaction. This is done under the "Payload" complex type definition. The name of the node pointed to in the above import is all that is needed.

At customer discretion is the entire "HeaderFields" attribute group. These attributes can be collected from the incoming document and saved for later use. Oracle Utilities Transaction Management includes a default set of attributes that customers may choose to use.

Functional Acknowledgement validation is much more straightforward. Since no customer data is copied into the internal format, no changes to the schema are required. Configuration of the outbound schemas is similarly easy. Once the SCHEMA_LOCATION and the FA_SCHEMA_LOCATION properties have been set on the Outbound Gateway process to the locations of schema files defining these types of documents, validation can take place. Since the outbound documents are in "customer" format, no changes are required to the schema templates. In order to prevent validation, these properties may be omitted. To prevent validation of incoming documents, set the Inbound Gateway's SCHEMA_LOCATION service property to a blank string (" "). Note that without the validation setting, incoming data will not be validated, and invalid structures could result in unpredictable faults at further points within the system.

Chapter 5

Oracle Utilities Transaction Management Server and Monitor

This chapter describes administrative tools used to run the Oracle Utilities Transaction Management application, and to monitor Oracle Utilities Transaction Management services and message queues. These tools include:

- **Oracle Utilities Transaction Management Server**
- **Oracle Utilities Transaction Management Monitor**

Note that these tools should not be accessible to end-users; they should only be used by system administrators.

Oracle Utilities Transaction Management Server

The Oracle Utilities Transaction Management Server is the core runtime engine that runs Oracle Utilities Transaction Management.

Starting the Oracle Utilities Transaction Management Server

To start the Oracle Utilities Transaction Management Server:

1. Select **Start-›Programs-›Oracle Utilities-›LTMHServer**.

   The **LTMHServer** window opens on the desktop.

2. The **Oracle Utilities Transaction Management Server Login Screen** opens.
3. Verify the **Local Host Name** (the machine name), the **Database Connection String**, the **User Name**, and **Password** and click **Login**. To cancel the connection, click **Cancel**.

Oracle Utilities Transaction Management Monitor

The Oracle Utilities Transaction Management Monitor allows administrators to monitor Oracle Utilities Transaction Management runtime services and message queues. The Oracle Utilities Transaction Management Monitor consists of three tabs:

- **Runtime Services Tab**
- **Runtime Queue Tab**
- **Messaging Tool Tab**

Starting the Oracle Utilities Transaction Management Monitor

To start the Oracle Utilities Transaction Management Monitor:

1. Select **Start->Programs->Oracle Utilities->LTMHMon**.

   The **LTMHMon** window opens on the desktop.

2. The **Oracle Utilities Transaction Management Monitor Login Screen** opens.
3. Verify the **Database Connection String**, the **User Name**, and **Password** and click **Login**. To cancel the connection, click **Cancel**.
4. The **Oracle Utilities Transaction Management Monitor** application opens.
5. Select the Oracle Utilities Transaction Management Server you wish to monitor from the Oracle Utilities Transaction Management Server list and click **Connect**. To disconnect from an Oracle Utilities Transaction Management Server, click **Disconnect**.

Runtime Services Tab

The Runtime Services tab allows you to monitor, start, stop, and refresh Oracle Utilities Transaction Management runtime services.

For each runtime service active on the Oracle Utilities Transaction Management Server, the Runtime Services tab displays the following information:

- **Service Name**: The name of the service.
- **Queue Name**: The message queue associated with the service.
- **Status**: The current status of the service. Status is indicated by color:
  - **Green**: Running
  - **Red**: Stopped
- **Messages**: The number of messages in the associated message queue.

How to start a service:

1. Select the service in the Service Name column and click **Start**.

How to stop a currently active service:

1. Select the service in the Service Name column and click **Stop**.

How to refresh a currently active service:

1. Select the service in the Service Name column and click **Refresh**.

Runtime Queue Tab

The Runtime Queue tab allows you to monitor and refresh Oracle Utilities Transaction Management message queues.

For each runtime queue, the Runtime Queue tab displays the following information:

- **Queue Name**: The message queue.
- **Messages**: The number of messages in the message queue.

How to refresh a message queue:

1. Select the queue in the Queue Name column and click **Q Refresh**.

Messaging Tool Tab

The Messaging Tool tab allows you to work with messages in the Oracle Utilities Transaction Management message queues.

The Messaging Tool tab provides a number of functions for working with messages and message queues.

How to work with the Messaging Tool:

1. Select the message queue you wish to work with from the **JMS Queues** list. If sending a message, type the message in the message box below the JMS Queues list.
2. Click the appropriate function button.

Function	Description
Send	Sends a message to a specified message queue. The message is typed in the message box below the JMS Queues list.
Receive	Removes and destroys the first available message in the selected queue (used primarily for testing).
Browse	Displays all messages in the selected queue.
Flush	Destroys all messages in the selected queue.
Clear	Clears the message box below the JMS Queues list.
Load	Loads multiple messages in the selected queue.

Oracle Utilities Transaction Management Services

Oracle Utilities Transaction Management services run on the Oracle Utilities Transaction Management server, and manage several of the functions of Oracle Utilities Transaction Management, including managing inbound and outbound transactions.

Service Properties

Oracle Utilities Transaction Management services are defined in the Hub Service table in the Oracle Utilities Data Repository. The Properties column in this table defines specific properties expected and/or required by the service. This column should contain an XML string listing the properties used by the service.

The Properties XML string should be in the following form:

```
<PROPERTIES>
  <PROPERTY NAME="PropertyName1" VALUE="PropertyValue1"/>
  <PROPERTY NAME="PropertyName2" VALUE="PropertyValue2"/>
  .
  .
  .
</PROPERTIES>
```

Element Descriptions

Each of the data elements used by this structure are described below.

Elements:

- **PROPERTIES:** Root element that contains one or more PROPERTY elements.
- **PROPERTY:** Element that defines a specific Property used by the service.

  Attributes:

  - **NAME:** The name of the Property.
  - **VALUE:** The value for the Property.

  Names and appropriate values for each property are listed in the descriptions below.

Property Descriptions

Each of the properties used by the services are described in the following format:

NAME: The name of the Property.

- **Required**: Is the Property required?
- **Possible Values**: Possible values for the Property.
- **Description**: A description of the Property.
- **Default Value**: The default value for the Property.

Inbound Gateway Service Property Descriptions

The Inbound Gateway Service expects the following Properties:

SCHEMA_LOCATION

- **Required**: Yes
- **Possible Values**: A valid URI for a schema
- **Description**: A required parameter of one or more sets of namespace-schema pairs. These should be space- delimited. Provided schemas are used to validate the contents of documents after they have been received and decrypted, but before they are transformed to internal representation. Schema validation may be deactivated by setting this attribute to empty string (not recommended). By setting SCHEMA_HAS_NAMESPACE to false, the namespace may be omitted and the location only may be supplied.
- **Default Value**: N/A

SCHEMA_HAS_NAMESPACE

- **Required**: No
- **Possible Values**: Y (yes) or N (no)
- **Description**: An optional parameter controlling whether the XML parser should expect a namespace with the schema location.
- **Default Value**: “Y”

MAPNAME

- **Required**: No
- **Possible Values**: Any Valid Map Name
- **Description**: A parameter that specifies the name of a map stored in the HUBMAP table. This map transforms the market XML document into the XML document. The map should be created and updated through the Oracle Utilities Transaction Management Configuration Workbench
- **Default Value**: N/A

VALIDATE_IDS

- **Required**: No
- **Possible Values**: Y (yes) or N (no)
- **Description**: An optional parameter controlling whether or not to perform a check for duplicate DOCREFIDs and TRANSIDs.
- **Default Value**: “Y”

FA_QUEUE

- **Required**: No
- **Possible Values**: Name of a JMS Messaging Queue
- **Description**: An optional parameter listing the name of the messaging queue onto which to place outbound FA tokens. To disable the sending of FA's, omit this parameter.
- **Default Value**: N/A

XML_BUFFER_SIZE

- **Required**: No
- **Possible Values**: Size in kilobytes

- **Description**: An optional parameter to controlling the SAX parser read increments. Some tests indicate that a larger buffer size can improve the parsing performance for relatively large files. The default buffer size in Xerces is 2K. This would give a good performance for small documents (less than 10K). For documents larger than 10K, specifying the buffer size to 4K or 8K will significantly improve the performance. But it's not recommended to set it to a value larger than 16K. For really tiny documents (1K, for example), you can also set it to a value less than 2K, to get the best performance.
- **Default Value**: “2”=KBs

FileLoader Service Property Descriptions

The FileLoader Service expects the following Properties:

RECEIVERID

- **Required**: No
- **Possible Values**: Any Valid Trading Partner ID
- **Description**: An optional parameter that defines the Receiver. The Receiver is a valid Trading Partner ID from the HubTradingPartner table. The polling directory is defined by this Receiver. If POLL_DIRECTORY is not defined, this property must be defined.
- **Default Value**: N/A

POLL_DIRECTORY

- **Required**: No
- **Possible Values**: A valid File System directory
- **Description**: An optional parameter defining the polling directory. If RECEIVERID is not defined, this property must be defined.
- **Default Value**: N/A

POLL_EXTENSION

- **Required**: No
- **Possible Values**: Any string
- **Description**: An optional parameter defining the extension to place on the exported document. The default value is "LTMH". Only available when POLL_DIRECTORY is defined.
- **Default Value**: “LTMH”

POLL_INTERVAL

- **Required**: No
- **Possible Values**: Number of minutes
- **Description**: An optional parameter defining the polling interval of the service.
- **Default Value**: 15 minutes

SEND_FA

- **Required**: No
- **Possible Values**: Y (yes) or N (no)
- **Description**: An optional parameter defining whether to send a Functional Acknowledgement. This information is passed to the next service through the TOKEN.
- **Default Value**: “N”

ENCRYPTION_CLASS

- **Required**: No
- **Possible Values**: Any valid Cryptography Class name
- **Description**: An optional parameter defining the Cryptography Class used for encryption. This information is passed to the next service through the TOKEN.
- **Default Value**: N/A

Outbound Gateway Service Property Descriptions

The Outbound Gateway Service expects the following Properties:

SENDERID

- **Required**: Yes
- **Possible Values**: Any Valid Trading Partner ID
- **Description**: A required parameter that defines the Sender. The Sender is a valid Trading Partner ID from the HubTradingPartner table. All documents sent by this instance of the Outbound Gateway will come from this Sender.
- **Default Value**: N/A

TRANS_PARENT_NODE

- **Required**: Yes
- **Possible Values**: Any Valid Node Name
- **Description**: A required parameter that defines the name of the parent node for the transaction node. The node name is defined by the market.
- **Default Value**: N/A

MAPNAME

- **Required**: Yes
- **Possible Values**: Any Valid Map Name
- **Description**: A required parameter that specifies the name of a map stored in the HUBMAP table. This map transforms the XML document into the market XML document. The map should be created and updated through the Oracle Utilities Transaction Management Configuration Workbench.
- **Default Value**: N/A

FA_MAPNAME

- **Required**: Yes
- **Possible Values**: Any Valid Map Name
- **Description**: A required parameter that defines the name of the map stored in the HUBMAP table. This map transforms the XML FA into the market XML FA. The map should be created and updated through the Oracle Utilities Transaction Management Configuration Workbench.
- **Default Value**: N/A

SCHEMA_LOCATION

- **Required**: No
- **Possible Values**: A valid URI for a schema
- **Description**: An optional parameter of one or more sets of namespace-schema pairs. These should be space- delimited. This schema definition is used to validate the contents of a document, after it has been transformed into the market layout. If this parameter is not defined, the document will not be validated.
- **Default Value**: N/A

FA_SCHEMA_LOCATION

- **Required**: No
- **Possible Values**: Any valid URI for a schema
- **Description**: An optional parameter of one or more sets of namespace-schema pairs. These should be space- delimited. This schema definition is used to validate the contents of a functional acknowledgement, after it has been transformed into the market layout. If this parameter is not defined, the functional acknowledgement will not be validated.
- **Default Value**: N/A

EXPORT

- **Required**: No
- **Possible Values**: Y (yes) or N (no)
- **Description**: An optional parameter that determines whether or not to export the document. If encryption is activated, the exported document would be encrypted.
- **Default Value**: “N”

EXPORT_DIRECTORY

- **Required**: No
- **Possible Values**: Any valid directory
- **Description**: An optional parameter defining the directory to export a document. The directory is relative to the Java runtime folder.
- **Default Value**: N/A

EXPORT_EXTENSION

- **Required**: No
- **Possible Values**: Any string
- **Description**: An optional parameter defining the extension to place on the exported document.
- **Default Value**: “LTMH”

REQUEST_QUEUE

- **Required**: No
- **Possible Values**: Any valid JMS Messaging Queue
- **Description**: An optional parameter defining the JMS queue to send a "request" for HTTPS. If this parameter is not defined, the request is not sent.
- **Default Value**: N/A

POLL_INTERVAL

- **Required**: No
- **Possible Values**: Number of minutes
- **Description**: An optional parameter defining the polling interval of the Outbound Gateway.
- **Default Value**: “15” minutes

MAXTRANS

- **Required**: No
- **Possible Values**: Number
- **Description**: An optional parameter defining the max number of transactions in a document.
- **Default Value**: “100” transactions per document

ENCRYPTION_CLASS

- **Required**: No
- **Possible Values**: Any valid Cryptography Class Name
- **Description**: An optional parameter defining the Cryptography Class used for encryption. If this parameter is not defined, the document is not encrypted.
- **Default Value**: N/A

PAYLOAD_OUTPUT

- **Required**: No
- **Possible Values**: D (decrypt) or E (encrypt)
- **Description**: An optional parameter defining the type of data exported. If export is activated, the data exported can either be encrypted or decrypted. The default value is encrypted.
- **Default Value**: “E” Encrypted

Chapter 6

Configuring Oracle Utilities Transaction Management Security

This chapter describes how to configure security to work with the Oracle Utilities Transaction Management application, including:

- **Oracle Utilities Transaction Management Permissions**

Oracle Utilities Transaction Management Permissions

The functionality of the Oracle Utilities Transaction Management application can be restricted through use of Roles, configured using the Security Administration functions of the Oracle Utilities Energy Information Platform.

The permissions that can be assigned to Roles for the Oracle Utilities Transaction Management application include the following:

- **Trans Mgt Hub**: Enables the Transaction Management menu option.
    - **Details**: Enables viewing document, process, transaction, and request details.
        - **Inbound Document**: Enables viewing of inbound documents
            - **View**: Enables the View option on the inbound document details screen.
            - **View FA Payload**: Enables the View FA Payload option on the inbound document details screen.
            - **Resubmit**: Enables the Resubmit option on the inbound document details screen
        - **Outbound Document**: Enables viewing of outbound documents.
            - **View**: Enables the View option on the outbound document details screen.
            - **View FA Payload**: Enables the View FA Payload option on the outbound document details screen.
        - **Process**: Enables viewing of processes.
            - **Terminate**: Enables the Terminate option on the process details screen.
        - **Transaction**: Enables viewing of transactions.
            - **Reject**: Enables the Reject option on the transaction details screen.
            - **Resubmit**: Enables the Resubmit option on the transaction details screen.
            - **Override**: Enables the Override option on the transaction details screen.
            - **Terminate**: Enables the Terminate option on the transaction details screen.
            - **View**: Enables the View option on the transaction details screen.
        - **Request**: Enables viewing of requests.
            - **View Response**: Enables the View Response option on the request details screen.
    - **Search**: Enables searching of documents, processes, transactions, requests/responses, and exceptions.
        - **Inbound/Outbound**: Enables the Search Inbound/Outbound menu item and function.
        - **Process**: Enables the Search Process menu item and function.
            - **Resubmit**: Enables the Resubmit option on the Details tab of the Process screen.
            - **Override**: Enables the Override option on the Details tab of the Process screen.
        - **Transaction:** Enables the Search Transaction menu item and function
            - **Resubmit**: Enables the Resubmit option on the Details tab of the Transaction screen.

        - **Override**: Enables the Override option on the Details tab of the Transaction screen.
    - **Request/Response**: Enables the Search Request/Response menu item and function.
    - **Exceptions**: Enables the Search Exceptions menu item and function.
    - **System/Token**: Enables the Search System/Token menu item and function.
        - **View Token**: Enables the View Token option on the System Exceptions screen.
- **Run Reports**: Enables the Run Reports option on the Transaction Management menu.
- **View Reports**: Enables the View Reports option on the Transaction Management menu.

Important Notes about Assigning Oracle Utilities Transaction Management Permissions

By default, the Oracle Utilities Transaction Management permissions restrict access to all Oracle Utilities Transaction Management functions and screens. To allow users access to Oracle Utilities Transaction Management functionality, create roles and enable appropriate permissions for each.

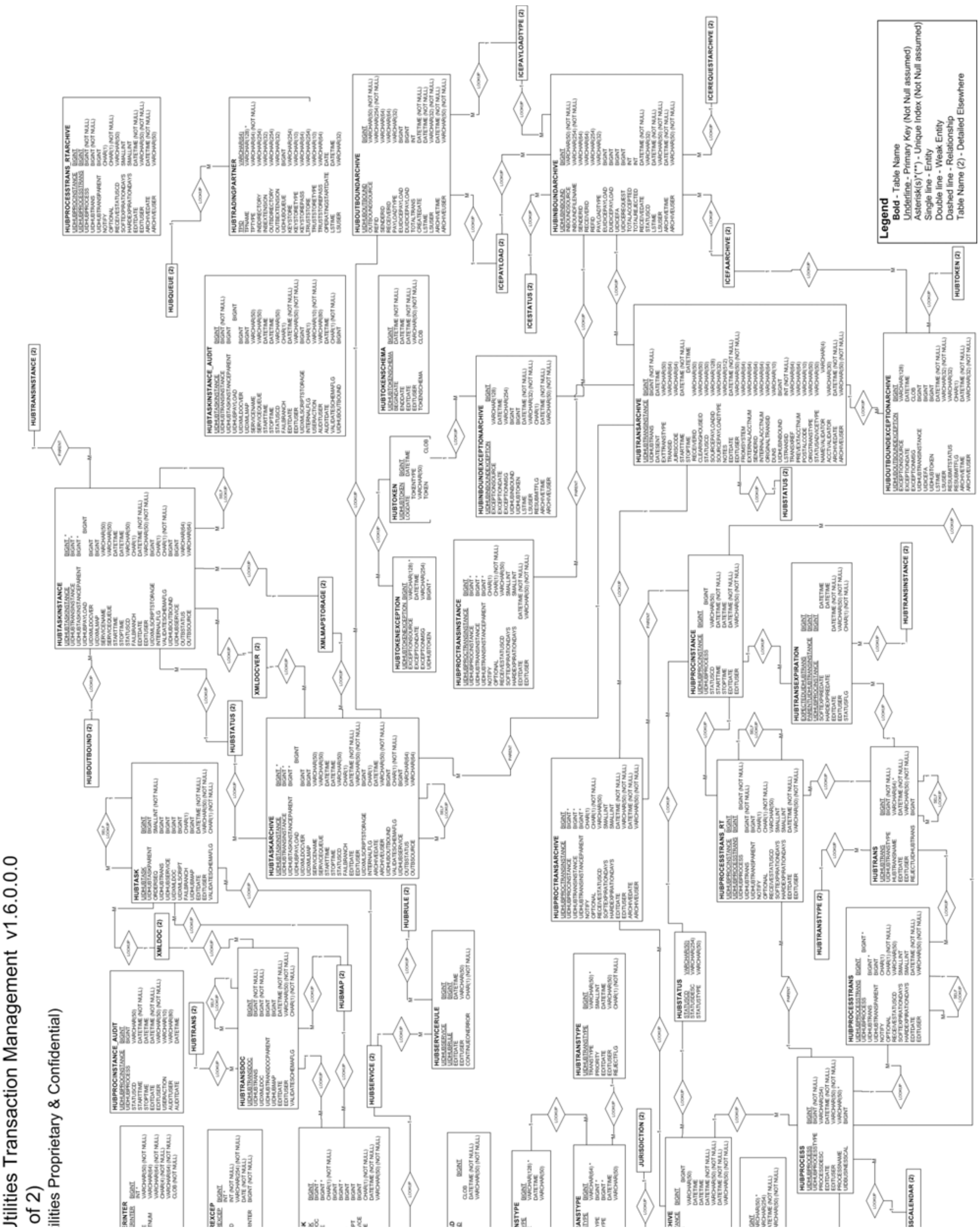
Appendix A

Oracle Utilities Data Repository Transaction Management Database Schema

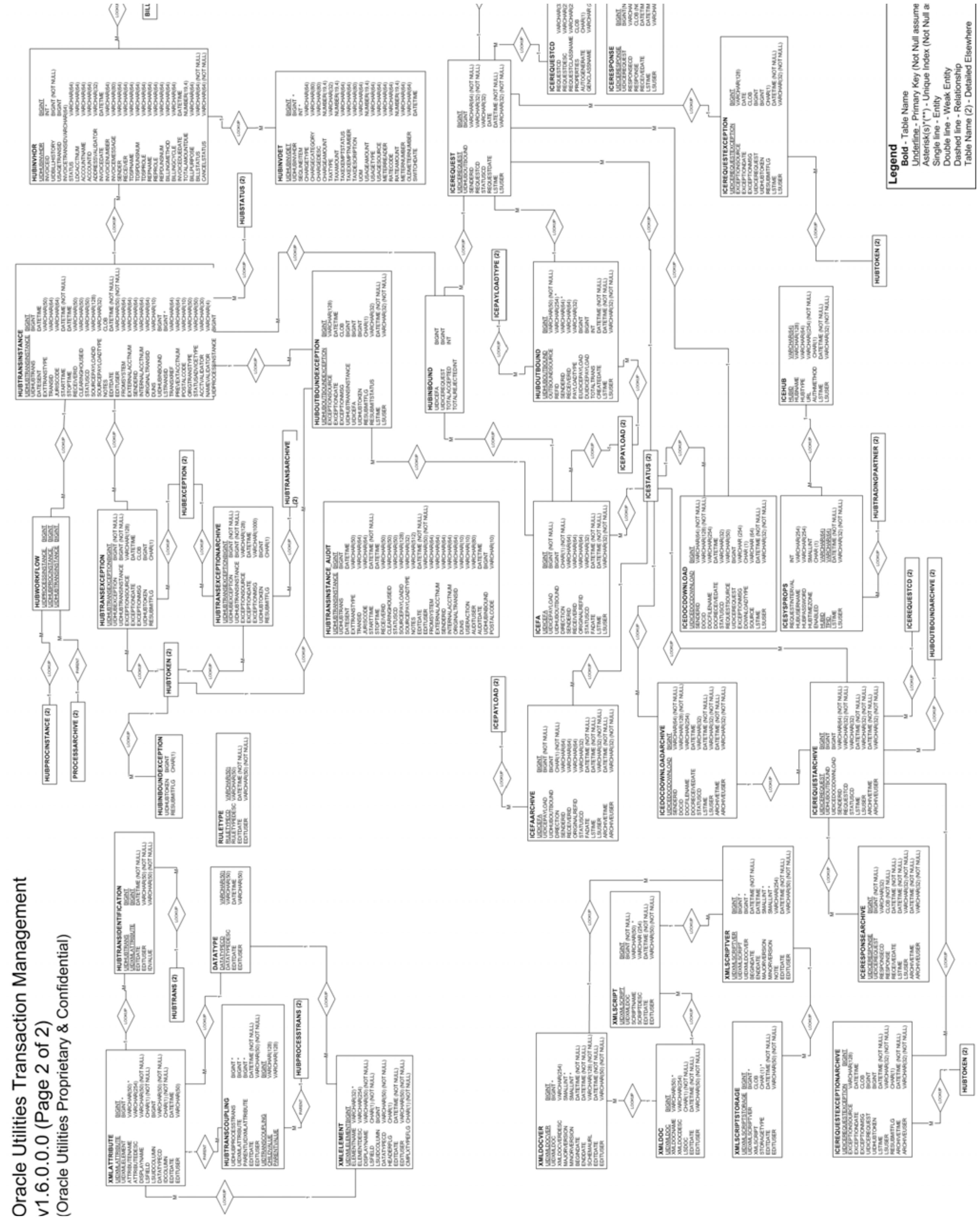
This appendix includes a diagram of the Oracle Utilities Data Repository Transaction Management database schema (v1.6.1.0.0) that provides details regarding the table and columns in the Oracle Utilities Transaction Management schema, as well as the relationships between these tables in the Oracle Utilities Data Repository. This information is very useful when writing Rules Language statements or constructing database queries. This includes:

- **Oracle Utilities Transaction Management Database Schema, page 1**
- **Oracle Utilities Transaction Management Database Schema, page 2**

Oracle Utilities Transaction Management Database Schema, page 1



Oracle Utilities Transaction Management Database Schema, page 2



Appendix B

Oracle Utilities Transaction Management Validation Scripts

Validation scripts allow you to enforce rules and constraints on the XML data coming into Oracle Utilities Transaction Management, as well as define your own error codes. This appendix provides an overview of how to create validation scripts, including:

- **Validating Unconditionally Required Elements and Attributes**
- **Validating Conditionally Required Elements**
- **Using Looping Structures and Relative Paths**

Validating Unconditionally Required Elements and Attributes

This section describes how to validate elements and attributes that are unconditionally required.

Rule Required Tags

Validation rules for unconditionally required elements must be placed inside "Rule Required" tags (`<RULE_REQUIRED> </RULE_REQUIRED>`).

Basic Syntax

The basic syntax for validation is as follows:

```
<RQ XPATH="/BusinessProcess/PayloadData/MessageType"/>
```

The above line would be interpreted as "MessageType is a required element in the XML data and must contain a value other than NULL."

Attributes and Values

The @ symbol denotes an attribute.

Use the VALUE keyword to indicate that an element or attribute must be equal to a certain value. For example:

```
<RQ XPATH="/BusinessProcess/PayloadData/Clearinghouse/@ClearinghouseRole"
VALUE="40,41"/>
```

The line above would be interpreted as "attribute ClearinghouseRole is required and must contain a value of 40 or 41."

Example

The following example shows how the rules may be written for unconditionally required elements and attributes.

```
<RULE_REQUIRED>
  <RQ XPATH="/BusinessProcess/PayloadData/MessageType"/>
  <RQ XPATH="/BusinessProcess/PayloadData/TransactionType"/>
  <RQ XPATH="/BusinessProcess/PayloadData/TransactionNumber"/>
  <RQ XPATH="/BusinessProcess/PayloadData/TransactionCreateDate"/>
  <RQ XPATH="/BusinessProcess/PayloadData/Utility/@UtilityRole" VALUE="41"/>
  <RQ XPATH="/BusinessProcess/PayloadData/Utility/UtilityDUNSNumber"/>
  <RQ XPATH="/BusinessProcess/PayloadData/Clearinghouse/@ClearinghouseRole"
VALUE="40,41"/>
</RULE_REQUIRED>
```

XPATH Values

The XPATH value indicates the path of the element or attribute within the XML structure. The following is a portion of a DTD; follow the highlighted items to see how the XPATH is derived.

```
<!ELEMENT BusinessProcess (BusinessProcessHeader, PayloadData)>

<!ELEMENT BusinessProcessHeader (BusinessProcessType, BusinessProcessState,
           DateSent, BusinessProcessESIID?, BusinessProcessActionCode?,
           Delivery, AdditionalRequest?, MappingStatus, RoutingInformation)>

<!ELEMENT BusinessProcessType (#PCDATA)>

<!ELEMENT BusinessProcessState (#PCDATA)>

<!ELEMENT DateSent (#PCDATA)>

<!ELEMENT BusinessProcessActionCode (#PCDATA)>

<!ELEMENT BusinessProcessESIID (#PCDATA)>

<!ELEMENT Delivery (BusinessProcessId, OriginalBusinessProcessId,
           GlobalProcessId, Sender, Receiver, ThirdParties?, DeliveryMethod,
           TransactionFormat)>

<!ELEMENT BusinessProcessId (#PCDATA)>

<!ELEMENT OriginalBusinessProcessId (#PCDATA)>

<!ELEMENT GlobalProcessId (#PCDATA)>

<!ELEMENT Sender (#PCDATA)>

<!ELEMENT Receiver (#PCDATA)>

<!ELEMENT DeliveryMethod (#PCDATA)>

<!ELEMENT TransactionFormat (#PCDATA)>

<!ELEMENT ThirdParties (#PCDATA)>

<!ELEMENT AdditionalRequest (#PCDATA)>

<!ELEMENT MappingStatus (#PCDATA)>

<!ELEMENT RoutingInformation (AckTime, Route, FastPath)>

<!ELEMENT AckTime (#PCDATA)>

<!ATTLIST AckTime  e-dtype NMTOKEN  #FIXED 'string' >
<!ELEMENT Route (#PCDATA)>

<!ATTLIST Route  e-dtype NMTOKEN  #FIXED 'string' >
<!--END Business Process -->
<!--START Business Data -->
<!ELEMENT PayloadData (MessageType, TransactionType, TransactionNumber,
           TransactionCreateDate, OriginalTransactionNumber?, Utility?,
           Clearinghouse, RetailElectricProvider, ProviderOfLastResort?,
           Customer*, TransactionData+)>

<!ATTLIST PayloadData  UniqueTransactionId CDATA  #REQUIRED >
<!ELEMENT MessageType (#PCDATA)>
```

Validating Conditionally Required Elements

Validation scripts often must validate conditionally required elements and attributes. That is, elements that may or may not be present. This section addresses how to handle conditionally required elements.

IF/THEN Statements

IF/THEN statements are used to define the rules for conditionally required elements and attributes. For example, consider the following rule:

```
<IF XPATH="/BusinessProcess/PayloadData/TransactionData/Premise/
MeteringLocation/@MeteringLocationQualifier" VALUE="MA,MX">
     <THEN XPATH="/BusinessProcess/PayloadData/TransactionData/Premise/
PremiseData/LoadProfile"/>
</IF>
```

This rule states that if attribute MeterLocationQualifier is equal to MA or MX then LoadProfile is required.

Optional Elements

The OPTIONAL keyword can be used to designate an element or attribute as optional, Notice the OPTIONAL keyword in the following example:

```
<IF XPATH="/BusinessProcess/PayloadData/RetailElectricProvider/
RetailElectricProviderName" OPTIONAL="YES">
  <THEN XPATH="../RetailElectricProviderName"/>
  <THEN XPATH="../RetailElectricProviderDUNSNumb"/>
  <THEN XPATH="../@RetailElectricProviderRole"/>
</IF>
```

The “OPTIONAL=YES” keyword indicates that the RetailElectricProviderName element is optional. The rules in the THEN portion only apply if that element is present. The default value is NO. Also, notice that a relative path is used in the THEN portion of the rule. In the preceding example, “../” indicates that all the items in the THEN portion of the statement are children of *"/BusinessProcess/PayloadData/RetailElectricProvider”*.

Conditional Values

In the following code example, notice that the VALUE keyword can also be used in both the IF and THEN portions of the rules.

```
<IF XPATH="/BusinessProcess/PayloadData/TransactionData/ServiceOrder/
@RequestType" VALUE="021">
  <THEN XPATH="/BusinessProcess/PayloadData/Customer/ServiceAddress/
ServiceName"/>
  <THEN XPATH="/BusinessProcess/PayloadData/Customer/ServiceAddress/
StreetAddress"/>
  <THEN XPATH="/BusinessProcess/PayloadData/Customer/ServiceAddress/
StreetAddressCity"/>
  <THEN XPATH="/BusinessProcess/PayloadData/Customer/ServiceAddress/
StreetAddressState"/>
  <THEN XPATH="/BusinessProcess/PayloadData/Customer/ServiceAddress/
StreetAddressZip"/>
  <THEN XPATH="/BusinessProcess/PayloadData/TransactionData/ServiceOrder/
PremiseType" VALUE="01,02,03"/>
</IF>
```

In this example, if the value of the RequestType attribute is 021, then ServiceName, ServiceStreetAddress, ServiceAddressCity, ServiceAddressState, and ServiceAddressZip are required, and the value of the TransactionData/ServiceOrder/PremiseType element is 01,02,03.

Nesting

You can also nest IF/THEN statements, as shown in the following example:

```
<IF XPATH="/BusinessProcess/PayloadData/TransactionData/ServiceOrder/
@RequestType" VALUE="001">
    <THEN XPATH="/BusinessProcess/PayloadData/TransactionData/ServiceOrder/
EffectiveDateChange"/>
    <THEN XPATH="/BusinessProcess/PayloadData/TransactionData/Premise/
PremiseData/LoadProfile"/>

    <IF XPATH="/BusinessProcess/PayloadData/TransactionData/Customer/
ServiceAddress/ServiceName" OPTIONAL="YES">
    <THEN XPATH="../ServiceName"/>
    <THEN XPATH="../ServiceStreetAddress"/>
    <THEN XPATH="../ServiceAddressCity"/>
    <THEN XPATH="../ServiceAddressState"/>
    <THEN XPATH="../ServiceAddressZip"/>
    </IF>

</IF>
```

The rule above indicates that if attribute RequestType equals 001 then EffectiveDateChange and LoadProfile are required, and if RequestType = 001 and ServiceName are present then ServiceName, ServiceStreetAddress, ServiceAddressCity, ServiceAddressState, and ServiceAddressZip are required.

Using Looping Structures and Relative Paths

This section describes how and when to use relative paths and looping with validation scripts.

When to Use Relative Paths

Relative paths can be used in IF/THEN clauses to represent looping structures. Consider the following XML example:

```
<LineItem>
  <Customer>
    <OrderInfo>
      <CustomerID>12</CustomerID>
      <ItemNo>1</ItemNo>
      <ItemQty>1</ItemQty>
      <ItemDescription>1</ItemDescription>
    </OrderInfo>
    <OrderInfo>
      <CustomerID>13</CustomerID>
      <ItemNo>2</ItemNo>
      <ItemQty>2</ItemQty>
      <ItemDescription>2</ItemDescription>
    </OrderInfo>
  </Customer>
</LineItem>
```

If you wanted to verify that if CustomerID is “12”, then ItemNo must equal “1” and ItemQty and ItemDescription must be present, you could code it as follows:

```
<IF XPATH="/LineItem/Customer/OrderInfo/CustomerID" VALUE=”12”>
  <THEN XPATH="../ItemNo" VALUE=”1”/>
  <THEN XPATH="../ItemQty"/>
  <THEN XPATH="../ItemDescription"/>
</IF>
```

The XPATHS in the THEN statements are relative to the XPATH in the IF statement. ‘../’ represents one element back from the last element of “/LineItem/Customer/OrderInfo”, so ‘../ ItemNo/’ represents “/LineItem/Customer/OrderInfo/ItemNo”.

If you apply the rules script above to the XML example, each iteration of OrderInfo will be processes separately. IF XPATH=“/LineItem/Customer/OrderInfo/CustomerID”, VALUE=“12” will evaluate the first CustomerID and only validate ItemNo, ItemQty, and ItemDescription in the first iteration. Once the first iteration is validated, processing will continue, re-evaluating CustomerID in the second iteration (CustomerID=13) and validating the corresponding ItemNo, ItemQty, and ItemDescription. Because CustomerID=“13” in the second iteration, ItemNo, ItemQty, and ItemDescription are not required and will return a successful validation.

Consider the following XML example:

```
<LineItem>
  <Customer>
    <OrderInfo CustomerID="12">
      <ItemNo>1</ItemNo>
      <ItemQty>1</ItemQty>
      <ItemDescription>1</ItemDescription>
    </OrderInfo>
    <OrderInfo CustomerID="13">
      <ItemNo>2</ItemNo>
      <ItemQty>2</ItemQty>
      <ItemDescription>2</ItemDescription>
    </OrderInfo>
  </Customer>
</LineItem>
```

In this scenario CustomerID is an attribute of OrderInfo, so the validation process of the OrderInfo loop is similar to the preceding example. However, the coding of the rules script is slightly different, as in the following example:

```
<IF XPATH="/LineItem/Customer/OrderInfo/@CustomerID" VALUE=”12”>
      <THEN XPATH="./ItemNo" VALUE=”1”/>
      <THEN XPATH="./ItemQty"/>
      <THEN XPATH="./ItemDescription"/>
</IF>
```

Notice that in THEN statements, ‘./’ is used to denote the relative path instead of ‘../’. In this scenario ‘./’ represents the entire path to the attribute, but does not including the attribute because CustomerID only describes the element OrderInfo and isn’t actually an element itself. So, ‘./’ represents “/LineItem/Customer/OrderInfo”.

When Not to Use Relative Paths

In the following example, the CustomerIDs are not directly associated with any items in the OrderInfo loop, so you would not be able to use relative paths.

```
<LineItem>
  <Customer>
    <CustomerID>12</CustomerID>
    <OrderInfo>
      <ItemNo>1</ItemNo>
      <ItemQty>1</ItemQty>
      <ItemDescription>1</ItemDescription>
    </OrderInfo>
    <OrderInfo>
      <ItemNo>1</ItemNo>
      <ItemQty>1</ItemQty>
      <ItemDescription>1</ItemDescription>
    </OrderInfo>
    <CustomerID>13</CustomerID>
    <OrderInfo>
      <ItemNo>2</ItemNo>
      <ItemQty>2</ItemQty>
      <ItemDescription>2</ItemDescription>
    </OrderInfo>
  </Customer>
</LineItem>
```

In cases like the preceding example, it would be illogical to use a relative path, as shown in the following example.

```
<IF XPATH="/LineItem/Customer/CustomerID" VALUE=”12”>
      <THEN XPATH="../OrderInfo/ItemNo" VALUE=”1”/>
</IF>
```

The processor would not to be able to determine which OrderInfo loop is being referenced because CustomerId is the same (not unique) for each iteration of OrderInfo.

Error Handling

You can define your own error codes by adding an ERRORCODE attribute, as shown in the following example:

```
<RQ XPATH="/BusinessProcess/PayloadData/MessageType" OPTIONAL="NO"
ERRORCODE=”NoMessageType” />
```

If MessageType is not present, you will get a “NoMessageType” as the error code. If you do not include the ERRORCODE attribute, a default value of ERRORCODE=“ERROR” is used.

Sample Rule Script for 814_20

The following is a sample validation script for an 814_20 transaction.

```
  <?xml version="1.0" ?>
  <!DOCTYPE EDI814_20 (View Source for full doctype...)>
- <EDI814_20>
  - <RULE_REQUIRED>
      <RQ XPATH="/BusinessProcess/PayloadData/MessageType"
         OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionType"
         OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionNumber"
         OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionCreate-
         Date" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Utility/@Utility-
         Role" VALUE="41" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Utility/Utili-
         tyName" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Utility/UtilityDUN-
         SNumber" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Clearinghouse/
         @ClearinghouseRole" VALUE="40,41" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Clearinghouse/Clear-
         inghouseName" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/Clearinghouse/Clear-
         inghouseDUNSNumber" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/@ServiceOrderId" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/@CommodityType" VALUE="EL" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/@RequestStatus" VALUE="7" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/@RequestType" VALUE="001,002,021" OPTIONAL="NO"
         />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/@ServiceType" VALUE="IN,MP" OPTIONAL="NO" />
      <RQ XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
         viceOrder/ESIID" OPTIONAL="NO" />
    </RULE_REQUIRED>
  - <RULE_CONDITION>
    - <!--
    Conditional Required for All

    - <IF XPATH="/BusinessProcess/PayloadData/RetailElectricPro-
         vider/RetailElectricProviderName" OPTIONAL="YES"
         LOGIC="YES">
        <THEN XPATH="../RetailElectricProviderName" LOGIC="YES"
           OPTIONAL="NO" />
        <THEN XPATH="../RetailElectricProviderDUNSNumb"
           LOGIC="YES" OPTIONAL="NO" />
        <THEN XPATH="../@RetailElectricProviderRole"
           LOGIC="YES" OPTIONAL="NO" />
      </IF>
    - <IF XPATH="/BusinessProcess/PayloadData/TransactionData/
         Premise/MeteringLocation/MeteringNumber" OPTIONAL="YES"
         LOGIC="YES">
        <THEN XPATH="../@MeteringLocationQualifier"
           VALUE="MA,MQ,MR,MX" LOGIC="YES" OPTIONAL="NO" />
      </IF>
```

```
- <!--
Required if Create

- <IF XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
     viceOrder/@RequestType" VALUE="021" LOGIC="YES"
     OPTIONAL="NO">
    <THEN XPATH="/BusinessProcess/PayloadData/Customer/Ser-
       viceAddress/ServiceName" LOGIC="YES" OPTIONAL="NO" /
       >
    <THEN XPATH="/BusinessProcess/PayloadData/Customer/Ser-
       viceAddress/StreetAddress" LOGIC="YES" OPTIONAL="NO"
       />
    <THEN XPATH="/BusinessProcess/PayloadData/Customer/Ser-
       viceAddress/StreetAddressCity" LOGIC="YES"
       OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Customer/Ser-
       viceAddress/StreetAddressState" LOGIC="YES"
       OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Customer/Ser-
       viceAddress/StreetAddressZip" LOGIC="YES"
       OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/ServiceOrder/PremiseType" VALUE="01,02,03"
       LOGIC="YES" OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/ServiceOrder/ESIIDStartDate" LOGIC="YES"
       OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/ServiceOrder/ESIIDEligibilityDate" LOGIC="YES"
       OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/Premise/PremiseData/StandardInterconnection-
       Point" LOGIC="YES" OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/Premise/PremiseData/DistributionLossFactor"
       VALUE="A,B,C,D,E,T" LOGIC="YES" OPTIONAL="NO" />
    <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/ServiceOrder/ERCOTTerritoryRelationship"
       VALUE="ERCOT,SERC,SPP,WSCC" LOGIC="YES"
       OPTIONAL="NO" />
  - <!--
         THEN XPATH="/BusinessProcess/Pay-
     loadData/TransactionData/Premise/Premise-
     Data/WholesaleDeliveryPoint" OPTIONAL="YES"/

  - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/Premise/MeteringLocation/MeteringNumber"
       LOGIC="YES" OPTIONAL="NO">
     - <!--
     Required if Create and Meter Addition or
       Meter Exchange

     - <IF XPATH="/BusinessProcess/PayloadData/Transac-
          tionData/Premise/MeteringLocation/@MeteringLo-
          cationQualifier" VALUE="MA,MX" LOGIC="YES"
          OPTIONAL="NO">
         <THEN XPATH="/BusinessProcess/PayloadData/
            TransactionData/Premise/PremiseData/Load-
            Profile" LOGIC="YES" OPTIONAL="NO" />
       </IF>
```

```
          - <!--
          Required if Create and Meter Addition

          - <!--
          Required if Create and Meter Remove

          - <!--
          Required if Create and Meter Exchange

            <IF XPATH="/BusinessProcess/PayloadData/Transac-
               tionData/Premise/MeteringLocation/@MeteringLo-
               cationQualifier" VALUE="MX" LOGIC="YES"
               OPTIONAL="NO" />
          </IF>
        </IF>
     - <!--
     Required if Change

     - <IF XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
          viceOrder/@RequestType" VALUE="001" LOGIC="YES"
          OPTIONAL="NO">
          <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/ServiceOrder/EffectiveDateChange" LOGIC="YES"
             OPTIONAL="NO" />
          <THEN XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/Premise/PremiseData/LoadProfile" LOGIC="YES"
             OPTIONAL="NO" />
       - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/Customer/ServiceAddress/ServiceName"
             OPTIONAL="YES" LOGIC="YES">
            <THEN XPATH="../ServiceName" LOGIC="YES"
               OPTIONAL="NO" />
            <THEN XPATH="../ServiceStreetAddress" LOGIC="YES"
               OPTIONAL="NO" />
            <THEN XPATH="../ServiceAddressCity" LOGIC="YES"
               OPTIONAL="NO" />
            <THEN XPATH="../ServiceAddressState" LOGIC="YES"
               OPTIONAL="NO" />
            <THEN XPATH="../ServiceAddressZip" LOGIC="YES"
               OPTIONAL="NO" />
          </IF>
       - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/Premise/PremiseData/DistributionLossFactor"
             OPTIONAL="YES" LOGIC="YES">
            <THEN XPATH="/BusinessProcess/PayloadData/Transac-
               tionData/Premise/PremiseData/DistributionLoss-
               Factor" VALUE="A,B,C,D,E,T" LOGIC="YES"
               OPTIONAL="NO" />
          </IF>
       - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/ServiceOrder/PremiseType" OPTIONAL="YES"
             LOGIC="YES">
            <THEN XPATH="/BusinessProcess/PayloadData/Transac-
               tionData/ServiceOrder/PremiseType"
               VALUE="01,02,03" LOGIC="YES" OPTIONAL="NO" />
          </IF>
       - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
             Data/ServiceOrder/ERCOTTerritoryRelationship"
             OPTIONAL="YES" LOGIC="YES">
            <THEN XPATH="/BusinessProcess/PayloadData/Transac-
               tionData/ServiceOrder/ERCOTTerritoryRelation-
```

```
         ship" VALUE="ERCOT,SERC,SPP,WSCC" LOGIC="YES"
         OPTIONAL="NO" />
    </IF>
 - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/ServiceOrder/ChangeReason" OPTIONAL="YES"
       LOGIC="YES">
      <THEN XPATH="/BusinessProcess/PayloadData/Transac-
         tionData/ServiceOrder/ChangeReason"
         VALUE="DTM307,N18R,REFAQ,REFKK,REFPTC,REF-
         SPL,REFSR" LOGIC="YES" OPTIONAL="NO" />
    </IF>
 - <IF XPATH="/BusinessProcess/PayloadData/Transaction-
       Data/Premise/MeteringLocation/MeteringNumber"
       LOGIC="YES" OPTIONAL="NO">
    - <!--
    Required if Change and Meter Addition

    - <!--
    Required if Change and Meter Remove

    - <IF XPATH="/BusinessProcess/PayloadData/Transac-
         tionData/Premise/MeteringLocation/@MeteringLo-
         cationQualifier" VALUE="MR" LOGIC="YES"
         OPTIONAL="NO">
       - <!--
        Can have more than one meter

         <THEN XPATH="/BusinessProcess/PayloadData/
            TransactionData/Premise/MeteringLocation/
            MeterDetails/EndMeterRead/@EndMeterRead-
            Type" LOGIC="YES" OPTIONAL="NO" />
         <THEN XPATH="/BusinessProcess/PayloadData/
            TransactionData/Premise/MeteringLocation/
            MeterDetails/@TimeOfUse"
            VALUE="41,42,43,51,71" LOGIC="YES"
            OPTIONAL="NO" />
       - <IF XPATH="/BusinessProcess/PayloadData/
            TransactionData/Premise/PremiseData/Unmet-
            eredServices/UnmeteredServiceType"
            OPTIONAL="YES" LOGIC="YES">
            <THEN XPATH="/BusinessProcess/PayloadD-
               ata/TransactionData/Premise/Premise-
               Data/UnmeteredServices/
               UnmeteredServiceType"
               VALUE="AR,BB,FL,HA,IN,LV,MH,MV,PS,RR,
               SD,TL,WM,WS" LOGIC="YES"
               OPTIONAL="NO" />
            <THEN XPATH="/BusinessProcess/PayloadD-
               ata/TransactionData/Premise/Premise-
               Data/UnmeteredServices/
               UnmeteredServiceDescription"
               LOGIC="YES" OPTIONAL="NO" />
            <THEN XPATH="/BusinessProcess/PayloadD-
               ata/TransactionData/Premise/Premise-
               Data/UnmeteredServices/
               NumberOfDevices" LOGIC="YES"
               OPTIONAL="NO" />
         </IF>
      </IF>
    - <!--
    Required if Change and Meter Exchange
```

```
- <IF XPATH="/BusinessProcess/PayloadData/Transac-
     tionData/Premise/MeteringLocation/@MeteringLo-
     cationQualifier" VALUE="MX" LOGIC="YES"
     OPTIONAL="NO">
  - <!--
    Can have more than one meter

    <THEN XPATH="/BusinessProcess/PayloadData/
       TransactionData/Premise/MeteringLocation/
       OldMeterNumber" LOGIC="YES" OPTIONAL="NO"
       />
  - <IF XPATH="/BusinessProcess/PayloadData/
       TransactionData/Premise/MeteringLocation/
       MeterDetails/EndMeterRead" OPTIONAL="YES"
       LOGIC="YES">
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/EndMeterRead/
          @EndMeterReadType" LOGIC="YES"
          OPTIONAL="NO" />
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/EndMeterRead"
          LOGIC="YES" OPTIONAL="NO" />
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/@TimeOfUse"
          VALUE="41,42,43,51,71" LOGIC="YES"
          OPTIONAL="NO" />
    </IF>
  </IF>
- <!--
Required if Change and Meter Exchange or
  Meter Addition

- <IF XPATH="/BusinessProcess/PayloadData/Transac-
     tionData/Premise/MeteringLocation/@MeteringLo-
     cationQualifier" VALUE="MA,MX" LOGIC="YES"
     OPTIONAL="NO">
  - <IF XPATH="/BusinessProcess/PayloadData/
       TransactionData/Premise/MeteringLocation/
       MeterDetails/MeterMultiplier"
       OPTIONAL="YES" LOGIC="YES">
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/
          MeterMultiplier/@MeterMultiplier-
          Type" LOGIC="YES" OPTIONAL="NO" />
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/
          MeterMultiplier" LOGIC="YES"
          OPTIONAL="NO" />
       <THEN XPATH="/BusinessProcess/PayloadD-
          ata/TransactionData/Premise/Meterin-
          gLocation/MeterDetails/@TimeOfUse"
          VALUE="41,42,43,51,71" LOGIC="YES"
          OPTIONAL="NO" />
    </IF>
  - <IF XPATH="/BusinessProcess/PayloadData/
```

```
               TransactionData/Premise/MeteringLocation/
               MeterDetails/NumberOfDials" OPTIONAL="YES"
               LOGIC="YES">
               <THEN XPATH="/BusinessProcess/PayloadD-
                  ata/TransactionData/Premise/Meterin-
                  gLocation/MeterDetails/NumberOfDials/
                  @NumberOfDialsMeterType" LOGIC="YES"
                  OPTIONAL="NO" />
               <THEN XPATH="/BusinessProcess/PayloadD-
                  ata/TransactionData/Premise/Meterin-
                  gLocation/MeterDetails/@TimeOfUse"
                  VALUE="41,42,43,51,71" LOGIC="YES"
                  OPTIONAL="NO" />
            </IF>
          - <IF XPATH="/BusinessProcess/PayloadData/
               TransactionData/Premise/MeteringLocation/
               MeterDetails/StartMeterRead"
               OPTIONAL="YES" LOGIC="YES">
               <THEN XPATH="/BusinessProcess/PayloadD-
                  ata/TransactionData/Premise/Meterin-
                  gLocation/MeterDetails/
                  StartMeterRead/@StartMeterReadType"
                  LOGIC="YES" OPTIONAL="NO" />
               <THEN XPATH="/BusinessProcess/PayloadD-
                  ata/TransactionData/Premise/Meterin-
                  gLocation/MeterDetails/@TimeOfUse"
                  VALUE="41,42,43,51,71" LOGIC="YES"
                  OPTIONAL="NO" />
            </IF>
          - <IF XPATH="/BusinessProcess/PayloadData/
               TransactionData/Premise/MeteringLocation/
               MeterChangeReason" OPTIONAL="YES"
               LOGIC="YES">
               <THEN XPATH="/BusinessProcess/PayloadD-
                  ata/TransactionData/Premise/Meterin-
                  gLocation/MeterChangeReason"
                  VALUE="DTM313,REF4L,REF4P,REFIX,REFLO
                  ,REFMT,REFNH,REFPR,REFPRT,REFTZ"
                  LOGIC="YES" OPTIONAL="NO" />
            </IF>
          </IF>
       </IF>
     </IF>
   - <!--
   Required if Retire

   - <IF XPATH="/BusinessProcess/PayloadData/TransactionData/Ser-
        viceOrder/@RequestType" VALUE="002" LOGIC="YES"
        OPTIONAL="NO">
        <THEN XPATH="./ESIIDEndDate" LOGIC="YES" OPTIONAL="NO" /
          >
     </IF>
  </RULE_CONDITION>
</EDI814_20>
```

Index