Oracle Utilities Network Management System

Configuration Guide

Release 1.9.0.2

October 2009

ORACLE®

Oracle Utilities Network Management System Configuration Guide, Release 1.9.0.2

Contents

Chapter 8

Chapter 9

Preface

Please read through this document thoroughly before beginning your product implementation The purpose of this guide is to provide implementation guidelines for a standard Oracle Utilities Network Management System implementation. This document discusses installation, interfaces, modeling, and software configuration that are considered typical and acceptable for a standard product implementation.

Audience

This document is intended for anyone responsible for the implementation of Oracle Utilities Network Management System.

Related Documents

- Oracle Utilities Network Management System Product Fix Documents
- Oracle Utilities Network Management System Installation Guide
- Oracle Utilities Network Management System Adapters Guide
- Oracle Utilities Network Management System User's Guide

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.

Convention	Meaning
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Chapter 1

System Overview

The Oracle Utilities Network Management System is an operations resource management system that runs on a UNIX platform. The system administrator is responsible for maintaining the UNIX operating system, the Oracle Utilities Network Management System, and the PC connections to remote workstations. This guide provides details about installing, optimizing, and troubleshooting the Oracle Utilities Network Management System and assumes that the reader is an experienced UNIX user.

- **System Overview**
- **Hardware and Third Party Software**
- **Network Architecture**
- **Architecture Guidelines**

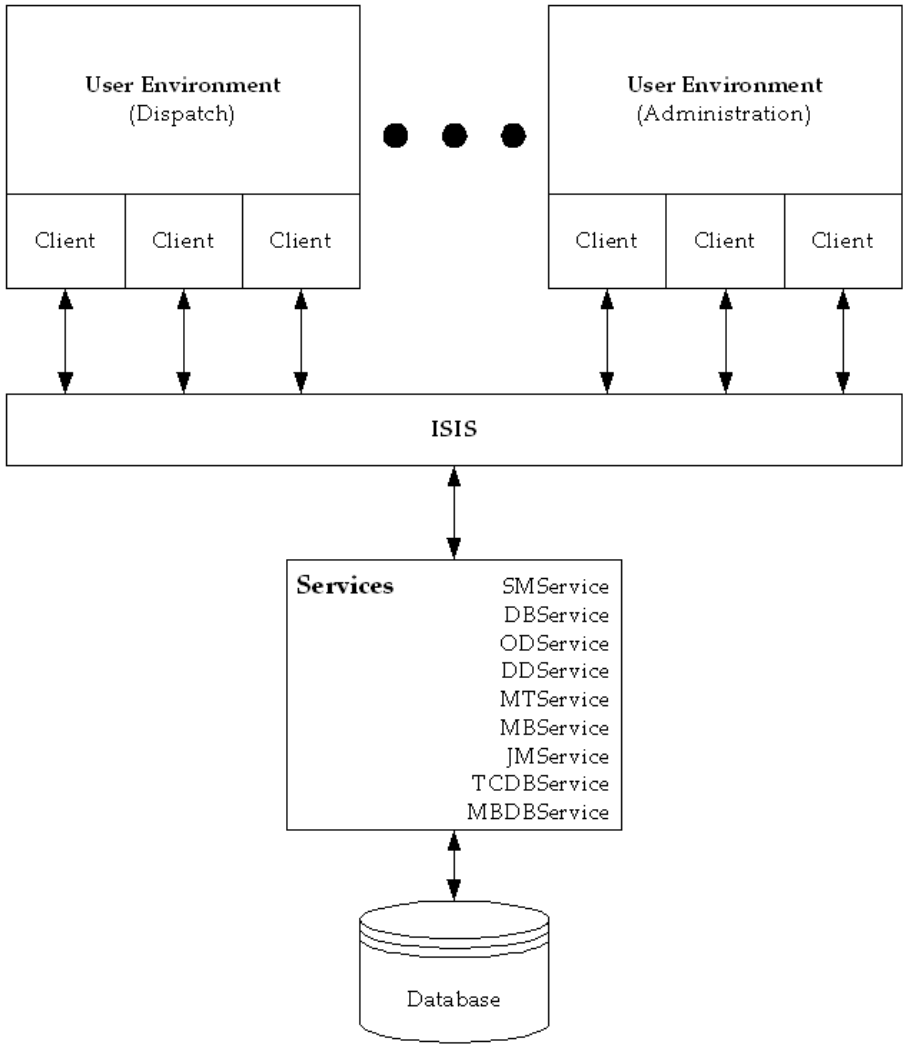
System Overview

An Oracle Utilities Network Management System includes:

- Isis
- Oracle Utilities Network Management System services
- Oracle Utilities Network Management applications
- User environments
- A tablespace in an Oracle database

The Oracle Utilities Network Management System can be broken down into individual components. Each component is installed and configured separately. Oracle Utilities Network Management System uses a client/server architecture. The server supports services and application tools required to run Oracle Utilities Network Management System software, while the clients display a graphical user interface to allow the user to interact with the system. Inter-application/Service communication is managed with a concurrency management and messaging system called Isis. Isis is the backbone of the communication architecture for an Oracle Utilities Network Management System. The network model, system configuration, and operational data is all stored persistently in an Oracle database, accessed and maintained via the various Oracle Utilities Network Management System *DBService Services.

The following diagram illustrates the standard configuration of the Oracle Utilities Network Management System.



The table below describes the Oracle Utilities Network Management System components.

Component	Description
User Environments	A set of user environments must be configured for the use of the client tools. The user environments support clients running Oracle Utilities Network Management System software in various modes, such as dispatch or administration. The display of the Oracle Utilities Network Management System applications can be forwarded to remote computers with the use of third party software (e.g., Hummingbird Exceed).
Isis	Clients access services and tools through a central concurrency management and messaging system called Isis. Isis is a real-time implementation of message oriented middleware and comprises the backbone of the system, providing access to the server for each client and the communication required between tools and services. Isis delivers the organized information to the client applications.

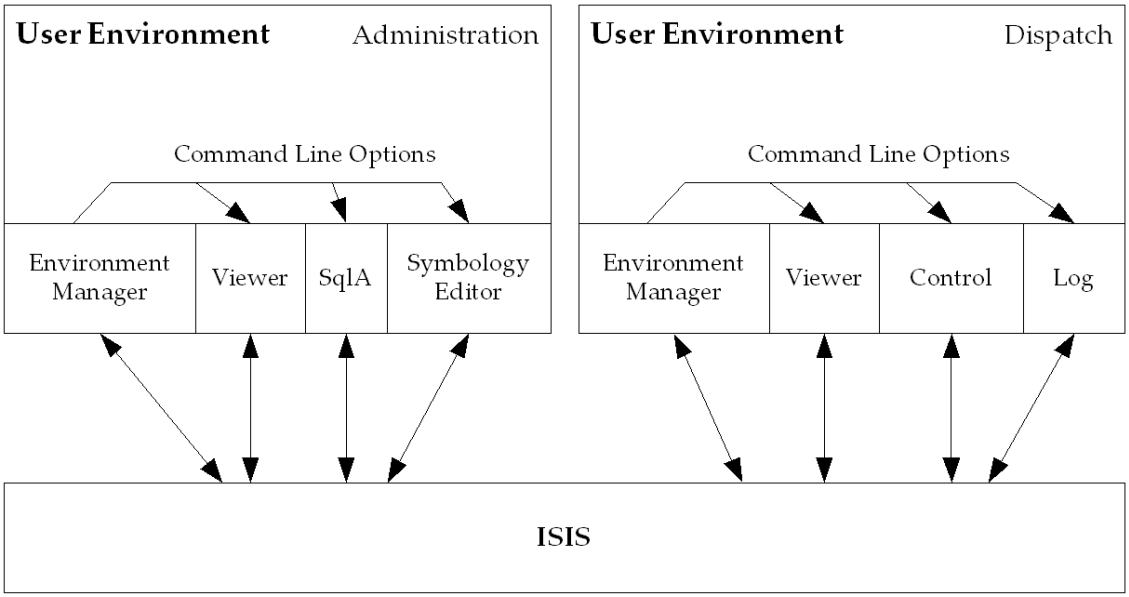
Component	Description
Services	Services maintain and manage the real-time model and data. Services also cache information from the database tables to optimize information requests from the applications.
Applications	Applications consist primarily of the front-end tools used by the Operational User. These tools access data from the services for presentation to the user and perform specific actions corresponding to appropriate business practices.
Oracle Database	The Oracle Database contains the complete network model, configuration, and operational data history of an Oracle Utilities Network Management System.

Note: Services, applications, and the Oracle tablespaces can be spread over multiple servers or run on a single server. The simplest configuration is for everything (Oracle RDBMS, Oracle Utilities Network Management System services and Oracle Utilities Network Management System applications) to run on a single (generally SMP) server. Common variations would include the use of a cluster based hardware server to support high-availability (for Oracle and Oracle Utilities Network Management System Services) and running two or more separate app-servers to support Oracle Utilities Network Management System Applications. This provides flexibility for system configuration, depending on your needs and hardware.

User Environments

Oracle Utilities Network Management System forms a managed visual workspace that organizes tools into related groups allowing users to perform specific tasks. Each group of tools makes up a separate user environment, or user type. The user type, entered when logging in to the application, establishes the environment by specifying the scripts that are run to launch tools. These scripts determine the set of tools available for each user type and define the sequence in which the tools are started.

Additionally, each tool supports a number of command line options that are chosen for each user type and stored in the database. These options are used when starting the tools and tailor each tool to the particular environment. The tools within each environment interact via information exchanged through Oracle Utilities Network Management System Model Management.



The user environment is configured using environment and services configuration files:

- The Oracle Utilities Network Management System resource file ($NMS_HOME/.nmsrc) is the UNIX environment configuration file that sets environment variables specific to the Oracle Utilities Network Management System. The .nmsrc file contains a list of environment variables that must be set in order for an Oracle Utilities Network Management System to operate correctly. The environment variables configure the connections to the database, Isis, viewer symbology, etc. New variables added to this file replace the old variables when the file is sourced within a shell.
- The .nmsrc file should be sourced each time the file is changed. The change in values will only affect the shell in which the file was sourced. Other previously existing shells will either need to be stopped and new ones started or have the .nmsrc file sourced in them as well. Also, programs that have previously been executed and are still running will need to be restarted in the new shell for the environment variables to take affect for them as well. Not recognizing these factors and failing to make the necessary adjustments is a significant source of problems encountered with installation and on-going maintenance of an Oracle Utilities Network Management System.
- The configuration file .profile should source the .nmsrc file on login. Refer to .profile configuration for information on how to do this.
- System Data file ($NMS_HOME/etc/system.dat) determines how and in what order Oracle Utilities Network Management System services and server hosted interfaces are started.
- Model Build System Data file ($NMS_HOME/etc/system.dat.model_build) determines how services are started for model build and configuration operations.
- Initialization System Data file ($NMS_HOME/etc/system.dat.init) determines how DBService is started for initial database setup.
- The ces_datefmt file (pointed to by the $DATEMSK environment variable) defines the optional date formats that may be encountered from the input of trouble call information.
- isis.prm is a parameter file (pointed to by the $ISIS_PARAMETERS environment variable) specifying the environment specific Isis parameters. The default values for Isis are generally sufficient for small- to medium-sized systems, but may need to be tuned and adjusted for large models and scalability.

Isis

Isis is the common messaging bus through which client processes and services interact on a TCP/IP network. Isis is primarily concerned with passing messages between nodes configured on the network. These nodes may be configured with system services, client tools, or a combination of both.

There are multiple hardware and message bus configurations that can be applied within the scope of a single Oracle Utilities Network Management System.

Database

Oracle Utilities Network Management System currently supports Oracle relational database management systems (RDBMS). The database persistently manages the tables that define the information constructs of the operations model. System services cache information from the relational tables. These tables include the management of system constructs such as handles and aliases, class hierarchy, topology model, device status, events, incidents (trouble calls), outages and conditions.

Database installation and configuration follow these basic steps:

- The selected database management system is initially configured using the product's standard installation and configuration procedures.
- Using Oracle Utilities Network Management System utilities, the initial schema is installed and populated.
- If you are performing an upgrade, you may need to perform a migration of the schema and population of the database.

All Oracle Utilities Network Management System schema definitions follow the SQL standard. Schema installation and population use SQL scripts that are generally executed via the SQL interface (ISQL.ces) to Oracle. If this is a single Oracle instance installation (not a redundant RDBMS configuration installation) the interactive SQLPlus interface of Oracle can generally be used for most configurations.

The necessary data elements required for an Oracle Utilities Network Management System consist of the following components.

Component	Description
Oracle Tablespaces	Used for persistent storage of production data (e.g., network components, operations data, etc), customer information and indexes. Connectivity to the tablespace is defined by the $RDBMS_USER, $RDBMS_PASSWD, $RDBMS_HOST environment variables which are equivalent to a DBA-defined Oracle user and password, and the connection global name as defined in the tnsnames.ora configuration file. The Oracle Utilities Network Management System model is typically loaded into three or more separate tablespaces, Operations data, Operations index and Customer (name, address, phone, account, etc) data.

Component	Description
Maps	Maps are stored in the $OPERATIONS_MODEL directory (usually $NMS_HOME/data). Two versions of maps are stored here, binary and text. In addition there are two types of maps, electrical and background. Electrical maps can always be regenerated from the RDBMS. Depending on how background maps are built for your model, background maps may not be. Some models build background maps as translations of background data from the customer-specific master GIS. Maps are tied to a specific database and cannot be associated with any other. In addition, the binary maps are O/S specific and used for faster loading during runtime. They can be quickly re-created from the text maps.

Hardware and Third Party Software

Since specific system requirements can change with new releases, they are not available as part of this document. For the most current requirements, refer to the *Oracle Utilities Network Management System Release Content* document.

Network Architecture

Running Oracle Utilities Network Management System software over a shared local area network and wide area network requires a network analysis. A network analysis allows you to validate the ability of the network to support the software. Network latency can cause significant problems with an Oracle Utilities Network Management System. Since all inter-process communication is managed by Isis via TCP/IP, significant latency on or a constrained network bandwidth can cause slowdowns, reduced throughput and possibly process shutdown. If you intend to operate Oracle Utilities Network Management System – Operator's Workspace over a WAN (any kind of WAN) or a LAN with more than around 5 millisecond latency, it is may be necessary (likely required) that you use some form of X11 proxy (for example HummingBird Exceed OnDemand).

Architecture Guidelines

This chapter provides an overview of the product module dependencies and locations, the logical hardware relationships, and sample physical hardware implementations:

- **Product dependencies and locations**
- **Logical hardware design**
- **Sample server implementations**
- **Hardware sizing**
- **Printing**

Overview

The guidelines in this section complement the information contained in the Oracle Utilities Network Management System Release Content. The Product Summary and Dependencies document has been replaced by a combination of the Release Content and this Architecture Guidelines document.

This section contains information about product module dependencies and locations, the logical hardware relationships, and sample physical hardware implementations. It should provide the

information needed to understand the relationships between the software modules and the hardware that is required to implement.

For an overall product summary, please refer to the Oracle Utilities Network Management System User Guide.

Product Dependencies and Locations

The following table describes Oracle Utilities Network Management System product module dependencies and their locations.

Module/ Component	Product	Dependency	Server	Client	Location
Model Management	OMS SE / DMS SE				
NMS Core Services			Unix		System Server
Web Gateway			Unix		System Server
Configuration Assistant				Windows	Web Client
US Electric Ops Model	OMS SE / DMS SE	Model Management			
Model Builder			Unix		System Server
US Standard Configuration	OMS SE / DMS SE	Model Management			
Application Configuration				Unix / Windows	Application Server/ Web Client
Operator's Workspace	OMS SE / DMS SE	Model Management			
Core Applications				Unix / Windows	Application Server/ Web Client
Trouble Management	OMS SE	Operator's Workspace			
Trouble Management Service			Unix		System Server
Trouble Management Applications				Unix / Windows	Application Server/ Web Client
High Availability	OMS SE / DMS SE	Model Management			

Module/ Component	Product	Dependency	Server	Client	Location
Cluster Capability			Unix		RDBMS Server/ System Server
Redliner	OMS SE / DMS SE				
Redliner Application				Windows	Clients
GIS Adapters	OMS SE / DMS SE	Model Management			
ESRI Adapter					GIS Server
Intergraph Adapter					GIS Server
Smallworld Adapter					GIS Server
OU Adapters	OMS SE				
CCB Adapter		Trouble Management	Unix		System Server
MWM Adapter		Trouble Management	Unix		System Server
Generic Adapters	OMS SE				
IVR Adapter		Trouble Management	Unix		System Server
CIS Adapter		Trouble Management	Unix		System Server
Switching Management	OMS EE / DMS SE	Model Management			
Switching Service			Unix		System Server
Switching Application - Motif				Unix	Application Server
Switching Application - Web				Windows	Web Client
Power Flow Extensions	DMS EE				
Power Flow Service		Model Management	Unix		System Server
Power Flow Applications		Operator's Workspace		Unix	Application Server
Suggested Switching	DMS EE	Power Flow Extensions	Unix		System Server

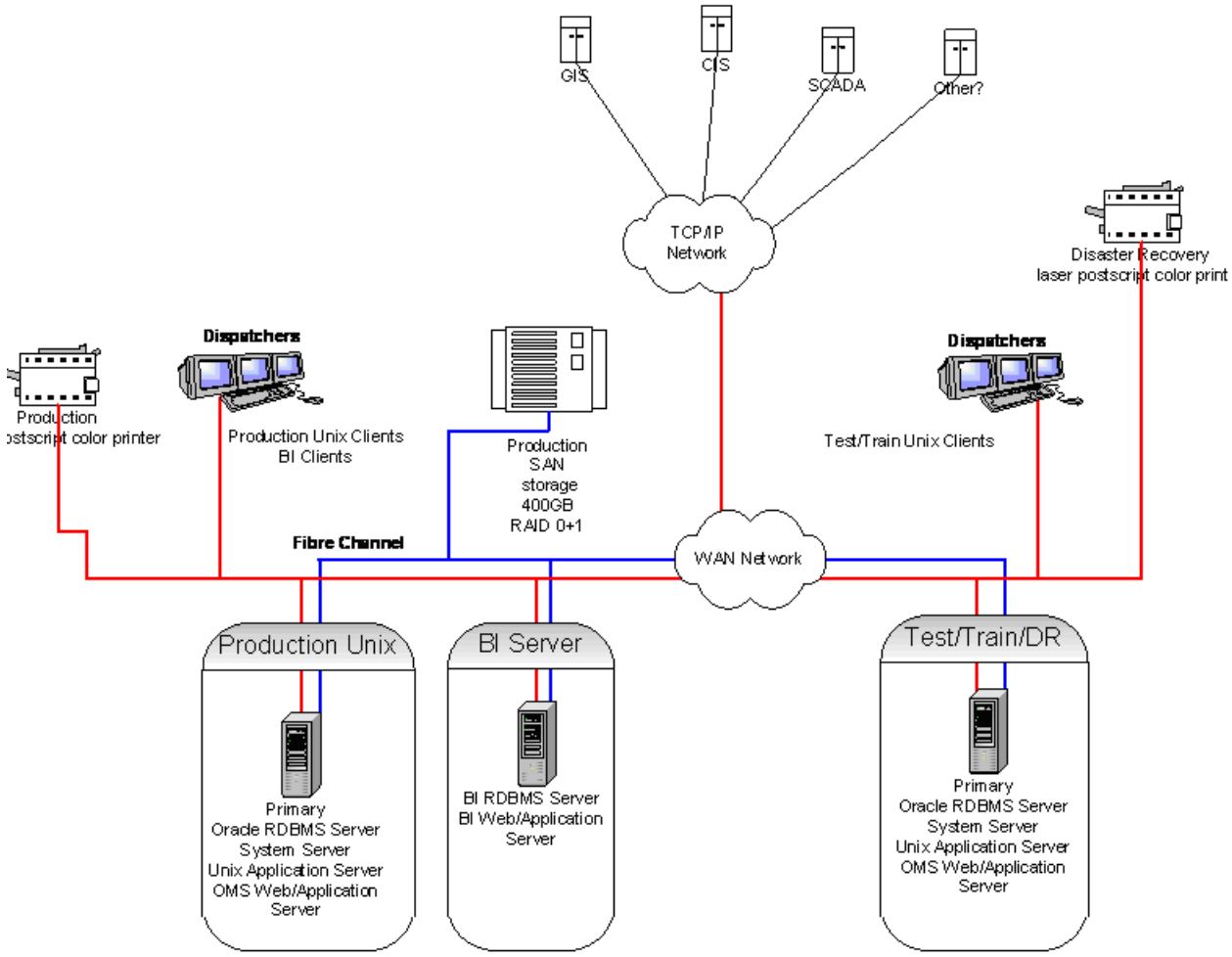
Module/ Component	Product	Dependency	Server	Client	Location
Feeder Load Management	DMS EE	Power Flow Extensions	Unix		System Server
Fault Location, Isolation & Service Restoration	NMS FLISR	Switching Management and SCADA Adapters	Unix		System Server
Volt/VAR Optimization	NMS VVO	Power Flow Extensions	Unix		System Server
Fault Location Analysis	NMS FLA	Power Flow Extensions	Unix		System Server
Schematics	OMS EE / DMS EE	Model Management			
Schematics Generator			Unix		System Server
Generic MQ Adapters	OMS EE				
CIS MQ Adapter		Trouble Management	Unix		System Server
CIS MQ Callback Adapter		Trouble Management	Unix		System Server
IVR MQ Adapter		Trouble Management	Unix		System Server
Mobile MQ Adapter		Trouble Management	Unix		System Server
WMS MQ Adapter		Trouble Management	Unix		System Server
Generic Adapters	OMS SE				
AMR Adapter		Trouble Management	Unix		System Server
Storm Management	NMS Storm	Trouble Management or Web Trouble		Windows	Web Client
Web Workspace	NMS Web Client	Model Management		Windows	Web Client
Web Trouble	NMS Web Client	Web Workspace		Windows	Web Client
Web Call Entry	NMS Call Center	Trouble Management or Web Trouble		Windows	Web Client

Module/ Component	Product	Dependency	Server	Client	Location
Web Callbacks	NMS Call Center	Trouble Management or Web Trouble		Windows	Web Client
Call Overflow Adapter	NMS Call Center				
21st Century Adapter		Trouble Management or Web Trouble	Unix		System Server
SCADA Extensions	NMS SCADA	Operator's Workspace or Web Workspace		Unix	Application Server
SCADA Adapters	NMS SCADA				
ICCP Blocks 1 & 2			Windows		ICCP Server
ICCP Block 5			Windows		ICCP Server
Generic SCADA			Unix		System Server
Service Alert	NMS Paging				
Service AlertService		Trouble Management or Web Trouble	Unix		System Server
Service Alert Client		Trouble Management or Web Trouble		Windows	Web Client
BI Framework	OUBI SE / OUBI EE		Unix		BI Web/ App Server
NMS Schema	OUBI SE / OUBI EE	Model Management	Unix		BI RDBMS Server
NRT & Historical Extractors	NMS Extractors and Reports	NMS Schema, and Trouble Management or Web Trouble	Unix		RDBMS Servers
NMS Portal	OUBI SE / OUBI EE	NRT & Historical Extractors	Unix		BI RDBMS Servers

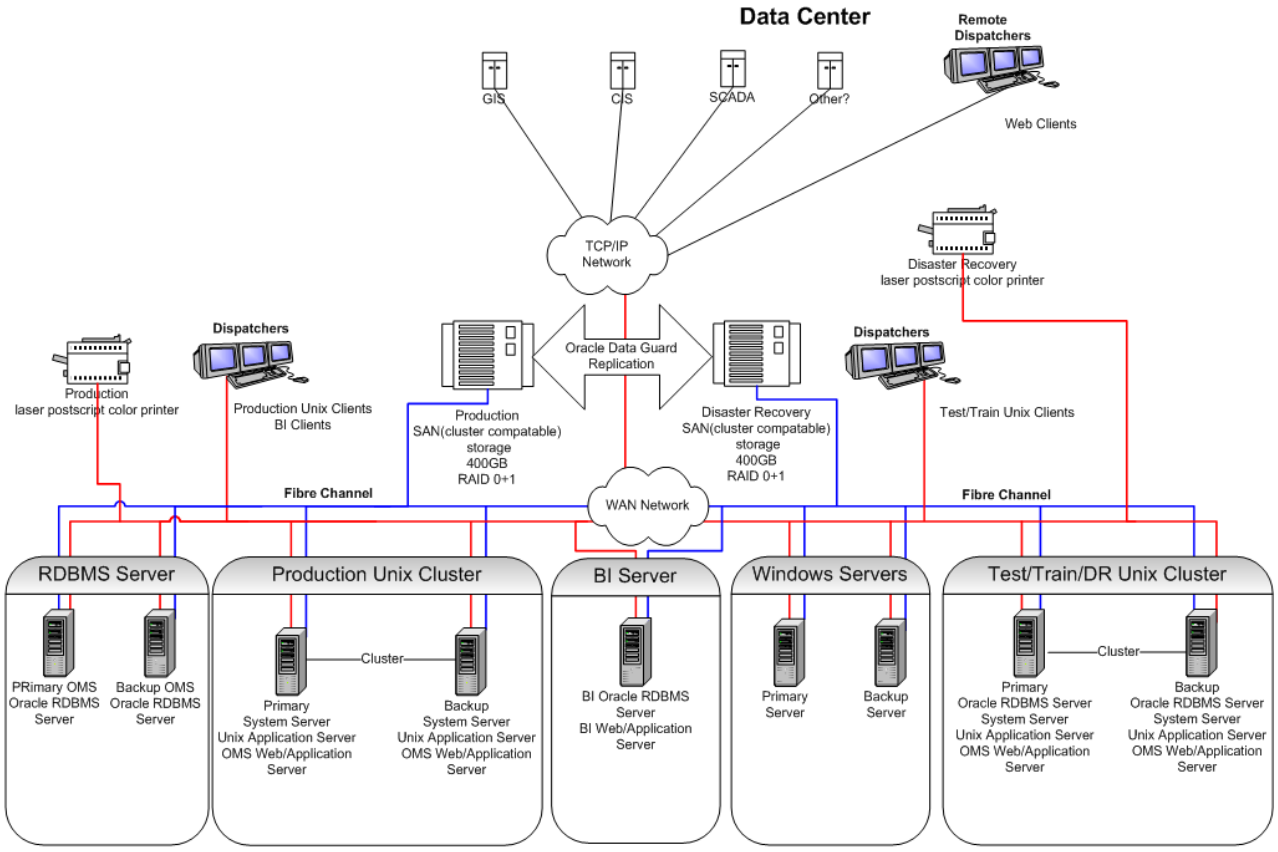
Module/ Component	Product	Dependency	Server	Client	Location
Analytics Portal	NMS Extractors and Reports	NMS Portal, and Trouble Management or Web Trouble		Windows	BI Web/ App Server
Feeder Load Analysis Portal	NMS Extractors and Reports	NMS Portal and Power Flow Extensions		Windows	BI Web/ App Server
Historical Storms Portal	NMS Extractors and Reports	NMS Portal and Storm Management		Windows	BI Web/ App Server
Trouble Reporting	NMS Extractors and Reports	Trouble Management or Web Trouble		Windows	BI RDBMS Server BI Report Server
Storm Reporting	NMS Extractors and Reports	Storm Management		Windows	BI RDBMS Server
Switching Reporting	NMS Extractors and Reports	Switching Management		Windows	BI RDBMS Server

Oracle Utilities Network Management System High-Level Hardware Diagrams

Example Simple High-Level Hardware/Software Diagram



Example Complex High-Level Hardware/Software Diagram



Hardware Sizing

Hardware sizing guidelines are not discussed in this document. There are many variables that affect hardware sizing and the calculations would be more complex than what is suitable for this document. Hardware sizing is best handled by the Professional Services team that is working on the project.

Printing

Printing from the Unix-based NMS applications requires that the printer be PostScript-compatible.

Chapter 2

Standard Product Implementation

This chapter provides an overview of a standard implementation of Oracle Utilities Network Management System, including:

- **Overview**
- **Software Release Level**
- **Installation**
- **Interfaces**
- **Modeling and GIS Integration**
- **Operations Modules Software Configuration**
- **Web Workspace and Web Trouble Software Configuration**
- **Management Reporting Modules Software Configuration**
- **Service Alert**

Overview

The changes to the Oracle Utilities Network Management System standard product software, installation, interfaces, modeling, and software configuration are considered typical and acceptable for a standard product implementation. Staying within these guidelines discussed in this guide allows a customer to follow the standard configuration from release to release and significantly reduces Oracle Utilities Network Management System migration and upgrade issues.

The intent is to allow a customer to make changes that follow the 80/20 rule; that is, a customer should be able to stick to 80% of the standard product configuration and only make the 20% configuration changes which are absolutely necessary to make the implementation successful.

There are many additional configuration changes possible and technically supported by Oracle; however, changes outside of these guidelines are considered project scope changes and redefine the project as a non-standard configuration project. This in turn creates testability and maintainability issues, as non-standard configuration may not be encompassed by our test process and can result in issues with which our customer support department may not be familiar. In addition, deviations from the product configuration mean your system will not conform as closely to standard product documentation and training material.

Software Release Level

A standard product implementation should utilize a release of Oracle Utilities Network Management System with no software code changes, additions or modifications. The software should be on an officially supported release code line and not a special project code line. Only

patches that are produced by the Oracle support organization and/or the project team should be installed when necessary to fix critical problems.

Installation

The installation should be done according to the guidelines taught in the Oracle Utilities Network Management System 'System Administration' class and follow all recommended procedures for system configuration. The software should be installed on servers and clients in a configuration that meets the requirements stated in the Oracle Utilities Network Management System Architecture Guidelines document for the installed release and modules. This includes the minimal disk, memory and CPU guidelines as well as the required operating system level and patches. The utilized Oracle Utilities Network Management System software modules should have all dependent Oracle Utilities Network Management System software modules installed and configured. The required third party products should be installed and at the supported release level as stated in the Oracle Utilities Network Management System Release Content document for the installed release.

Interfaces

A standard product implementation should use the Oracle Utilities Network Management System standard CIS, IVR, WMS and mobile data interfaces with an officially supported middleware gateway such as the MQSeries gateway or using the Oracle Table Interface. SCADA system integration should be done utilizing the Oracle Utilities Network Management System LiveData ICCP Adapter. AMR/AMI and AVL integrations should be done using the Oracle Utilities Network Management System MultiSpeak Adapter. Paging and email notification integration should be done using the Service Alert supported services.

When interfaces are done to non-standard systems that cannot be supported utilizing the standard interfaces described above, they should be done utilizing the published APIs and should not directly read or write to the Oracle Utilities Network Management System operations database.

Database level and/or reporting integration may be done using the Oracle Utilities Business Intelligence database and must utilize tables and attributes described in the Oracle published schema.

Modeling and GIS Integration

The following sections describe some recommended guidelines to follow when you integrate Oracle Utilities Network Management System with a GIS.

GIS Model Extractor

The GIS extractor utilized should either be supported by Oracle or by one of our modeling partners. The extractor should produce Oracle standard model preprocessor (MP) files and utilize the Oracle conventions for model building and an approved incremental update process.

Standard Preprocessor

The Oracle Utilities Network Management System standard preprocessor supports eighteen different rules that allow for data translation (for instance, expand elbows, or add recloser bypass switch). It is acceptable to use as many of these rules as necessary to build an acceptable operations model. The standard preprocessor takes as input model preprocessor (MP) files and produces Oracle standard model build (MB) files.

Device Types and Attributes

Select which device types (classes) are used from the standard model definition, mapping the customer's GIS data to these existing classes.

- Define unique class alias names based on the GIS attribute(s).
- Select which attributes are used from the standard model definition (providing at a minimum those necessary for the required modules), again mapping the customer's GIS data to the existing attributes.
- Utilize Oracle-provided modeling workbooks to define the model used for the project, which is used to generate the project classes and inheritance.
- The name of any device may be constructed from one or more GIS attributes.
- The display name for any device type can be changed (for instance, allows the device type to have a different name on the control tool).

Software Configuration Dependent Upon Device Types

There are a number of aspects to OMS software configuration that depend upon the device types that are chosen to be built within the OMS data model. In so far as the data model can change for different facilities, the software configuration must be adapted. The following configuration settings are dependent upon the resulting OMS model definition and require adaptation for every project. These configurations are generated automatically by Oracle to match the defined OMS model.

- Control Tool panels
- Hide/Display configuration
- Trouble Management Stop Classes
- Symbology mapping and symbol set

Operations Modules Software Configuration

This section describes how you configure the operations modules in Oracle Utilities Network Management System, including:

- **Operator's Workspace**
- **Trouble Management**
- **Web Call Entry**
- **Web Callbacks**
- **Switching Management**
- **Storm Management**
- **Power Flow Extensions**
- **Suggested Switching**
- **Redliner**
- **SCADA Extensions**
- **Web Workspace and Web Trouble Software Configuration**
- **Management Reporting Modules Software Configuration**
- **Service Alert**
- **Fault Location, Isolation, and Service Restoration (FLISR)**
- **Volt/VAr Optimization**
- **Fault Location Analysis**
- **Feeder Load Management**

Overview

Unless there is sound reason to change them, Oracle recommends that labels, buttons, table columns and dialogs be left as-is for consistency. This avoids confusion and further improves our ability to support our customers. However, there are cases where such changes are allowed, and the following sections identify those cases. There are also cases where it is allowable to delete a field, button or label. This may mean that the deleted item is actually just "hidden". Depending upon where on the form the deleted or hidden item was originally placed, there may be some "white space" remaining where the deleted item was present.

Operator's Workspace

The following sections describe the cases where it is allowable to change values in the Operator's Workspace.

Login

Allowable values you may change include:

- Add project logo bitmap
- Add and remove usernames and passwords (Customer can do this using Configuration Assistant)
- Delete or rename user types

Authority

Allowable values you may change include:

- Define specific control zone hierarchy (up to 5 levels)

Viewer

Allowable values you may change include:

- Change project symbology file used by Oracle Utilities Network Management System (Customer responsibility using Oracle provided tools) - includes AVL crew symbology if configured
- Viewer background color may be gray or black
- Annotation and/or landbase color may be changed to match the Viewer background. All annotation is assumed to be one color, and all landbase graphics are assumed to be a single color.
- Zoom levels
- Declutter / reclutter
- Big Symbols
- Model-dependent application defaults
    - Current feeder trace-to class and current feeder column
    - Selectable and unselectable objects

Control Tool

Allowable values you may change include:

- Change labels for actions
- Delete actions

Environment Manager

Allowable values you may change include:

- Delete or rename buttons

Trouble Management

The following sections describe the cases where it is allowable to change values for Trouble Management.

Prediction Rules

Allowable values you may change include:

- Delete any standard rule set
- Change parameter values of any rule in any standard rule set (Customer can do this using Configuration Assistant)
- Delete any rule in any standard rule set (except in cases where there are rule dependencies)

Call Entry

Allowable values you may change include:

- Add/Remove Trouble Codes but must map to Oracle standard trouble codes

Work Agenda

Allowable values you may change include:

- Change labels of any column
- Change labels of any button
- Add three permanent filters (Customer can do this using Configuration Assistant)
- Add three permanent sorts
- Change set of Work Queues (or Dispatch Groups)

Event Details

Allowable values you may change include:

- Delete outage reporting drop down menus
- Rename outage reporting drop down menus
- Add and delete items on outage reporting drop down menus (Customer can do this using Configuration Assistant)
- Add additional option menu field verification prior to completion (e.g., not only must the Failure and Remedy be changed from "Unselected", but it may also check for values in other option menu fields prior to completion)
- Remove current completion validation check or any other configured validation check

Crew Administration

Allowable values you may change include:

- Add and Remove Crew Types from standard list of crew types
- Add and Remove Personnel Job Titles from standard list of job titles
- Add and Remove Vehicle/Equipment types from standard list of vehicle/equipment type

Trouble Summaries

Allowable values you may change include:

- Match summaries to control zone hierarchy
- Change labels of any column
- Change labels of any button

Damage Assessment

Allowable values you may change include:

- Add, remove, or rename damage types
- Modify the minutes to repair, and minutes to repair if inaccessible, for each damage type
- Add, remove, or rename damage parts

Web Call Entry

The following describes the values that are allowable to change for Web Call Entry:

- Add login front-end if desired
- Add and remove usernames (if login configured)
- Add/Remove Trouble Codes but must map to Oracle standard trouble codes (Customer can do this using Configuration Assistant)
- Change labels and order of columns in Outages Summary
- Modify Event History Cause dropdowns to reflect outage reporting drop down menus

Web Callbacks

The following describes the values that are allowable to change for Web callbacks:

- Add and remove usernames
- Add/Remove Callback Status options but must map to Oracle standard callback statuses
- Change labels of any column in main window and View My Callback Lists window
- Change ordering of columns in main window and View My Callback Lists window

Switching Management

The following sections describe the cases where it is allowable to change values for Switching Management.

Switching List/Safety List

Allowable values you may change include:

- Change labels of columns
- Change labels of buttons

Switching Documents

Allowable values you may change include:

- Change labels on any header field
- Delete any header field
- Change header labels on any switching step field

- Add additional required fields verification prior to state change
- Remove any validation check or any other configured validation check

Safety Documents

Allowable values you may change include:

- Rename any safety document
- Change labels on field of standard documents
- Add additional required fields verification prior to state change
- Remove any validation check or any other configured validation check
- Delete any fields of standard documents
- Delete any standard documents
- Define up to three new safety documents (starting from a copy of any standard safety documents and making any of the allowable changes listed above)

Storm Management

The following describes the values that are allowable to change for Storm Management:

- Change labels of columns
- Change labels of buttons
- Change the historical average lookup values, but not how they are used in the algorithm
- Define a sort order for the events that is used by the analysis engine prior to stepping through its periodic analysis, within the constraints of the configuration options available for this purpose
- Change storm outage type names, definitions and restoration order, within the constraints of the configuration options available for this purpose, including adding or removing some (but not all) outage types (directly tied to the historical average lookup value definition process)
- Specify which of the top three control zone levels is the "simulation level" (the level at which the lookup values are specified)
- Define which crew types are eligible to assess/repair which storm outage types
- Define performance factors for each crew type
- Define nominal crew resources
- Change storm shift definitions, within the constraints that there must be at least one but no more than four shifts, and the sum of all shift lengths must be exactly 24 hours (directly tied to the historical average lookup value definition process)
- Change storm season definitions, within the constraints that there must be at least one but no more than four seasons, and each month must be part of a season (directly tied to the historical average lookup value definition process)
- Change storm holiday definitions, including the removal of all holiday definitions (directly tied to the historical average lookup value definition process)
- Change storm special conditions types (directly tied to the historical average lookup value definition process)
- Change storm level names, including adding or removing some (but not all) levels
- Change list of company names for the crew resources, including adding or removing some (but not all) names

- Add and remove usernames and passwords
- Delete or rename user types

Power Flow Extensions

The following sections describe the cases where it is allowable to change values for Power Flow Extensions.

Power Flow User Tools

Allowable values you may change include:

- Change labels of columns on Power Flow user tools
- Change labels of buttons on Power Flow user tools

Power Flow Algorithm Rules

Allowable value you may change includes:

- Change parameter values of any power flow algorithm rule in PFS_RULES table.

Load Profile

Allowable values you may change include:

- Number of seasons
- Number of day types
- Number of load periods per day

Seasonal Conductor and Transformer Flow Ratings

Allowable values you may change include:

- Normal limit
- Emergency limit

Power Flow Switching Extensions

Allowable value you may change includes:

- Change labels of Power Flow specific columns on switching steps

Suggested Switching

The following describes the values that are allowable to change for Suggested Switching:

- Change labels of columns on Suggested Switching user tools
- Change labels of buttons on Suggested Switching user tools

Redliner

There are no configuration options available

SCADA Extensions

The following describes the values that are allowable to change for SCADA Extensions:

- Change labels of columns on SCADA work agenda page
- Change labels of buttons on SCADA work agenda page

- Change alarm limit values

Web Workspace and Web Trouble Software Configuration

This section describes how you configure the Web Workspace and Web Trouble modules in Oracle Utilities Network Management System.

Overview

Some of the items in Web Workspace and Web Trouble are configured using the same mechanism as Operator's Workspace and Trouble Management, so it is not possible to configure Web Workspace and Web Trouble differently from the Operator's Workspace and Trouble Management configuration. In other cases, Web Workspace and Web Trouble configuration is performed differently so it is possible to have differences. However, Oracle recommends being as consistent as possible between the product lines. Oracle does recognize that in some cases the differences are desirable due to the desire to make Web Workspace and Web Trouble usable by less experienced users or to accommodate differences due to the PC/web environment of Web Workspace and Web Trouble.

Web Workspace

The following sections describe the cases where it is allowable to change values in Web Workspace.

Login

Allowable values you may change include:

- Add and remove usernames and passwords (Customer can do this using Configuration Assistant)
- Delete or rename user types

Authority

Allowable value you may change includes:

- Define specific control zone hierarchy (up to 5 levels) - uses same as is configured for Operator's Workspace

Viewer

Allowable values you may change include:

- Change project symbology file used by Oracle Utilities Network Management System (Customer responsibility using Oracle provided tools) - uses same as is configured for Operator's Workspace
- Viewer background color may be gray or black.
- Annotation and/or landbase color may be changed to match the Viewer background. All annotation is assumed to be one color.
- Initial zoom level
- Declutter / reclutter
- Big Symbols
- Selectable and unselectable objects

Control Tool

Allowable values you may change include:

- Change labels for actions
- Delete actions

Environment Manager

Allowable values you may change include:

- Delete or rename buttons

Web Trouble

The following sections describe the cases where it is allowable to change values in Web Trouble.

Prediction Rules

Web Trouble uses the same rules as configured for Trouble Management. Allowable values you may change include:

- Delete any standard rule set
- Change parameter values of any rule in any standard rule set (Customer can do this using Configuration Assistant)
- Delete any rule in any standard rule set (except in cases where there are rule dependencies)

Work Agenda

Allowable values you may change include:

- Change labels of any column
- Change labels of any button
- Change ordering of columns
- Add three permanent filters (Customer can do this using Configuration Assistant. This uses the same filters as configured for Trouble Management)
- Add three permanent sorts
- Change set of Work Queues (or Dispatch Groups) - Customer can do this using Configuration Assistant. This uses same as those configured for Trouble Management

Event Details

Allowable values you may change include:

- Delete outage reporting drop down menus
- Rename outage reporting drop down menus
- Add and delete items on outage reporting drop down menus - uses same as is configured for Trouble Management
- Add additional option menu field verification prior to completion (e.g., not only must the Failure and Remedy be changed from "Unselected", but it may also check for values in other option menu fields prior to completion)
- Remove current completion validation check or any other configured validation check

Crew Administration

Allowable values you may change include:

- Add and remove Crew Types from standard list of crew types - uses same as is configured for Trouble Management
- Add and remove Personnel Job Titles from standard list of job titles - uses same as is configured for Trouble Management
- Add and remove Vehicle/Equipment types from standard list of vehicle/equipment type - uses same as is configured for Trouble Management

Management Reporting Modules Software Configuration

The following sections describe the cases where it is allowable to change values in the Management Reporting modules.

Business Intelligence

Allowable values you may change include:

- Modify extractor to match configuration within guidelines of accepted product configuration changes
- Oracle provides standard Oracle Utilities Network Management System zones and portals; it is the customer's responsibility to modify these to meet business needs

Trouble Reports

Allowable values you may change include:

- Oracle provides standard reports in Business Objects or Oracle Discover (customer choice); it is the customer's responsibility to modify these to meet business needs

Switching Reports

Allowable values you may change include:

- Oracle provides standard reports in Business Objects or Oracle Discover (customer choice); it is the customer's responsibility to modify these to meet business needs

Storm Reports

Allowable values you may change include:

- Oracle provides standard reports in Business Objects or Oracle Discover (customer choice); it is the customer's responsibility to modify these to meet business needs

Service Alert

Service Alert provides a user interface for update and maintenance of the contact list, notification parameters, customer contact information, and critical/sensitive customer information; it is the customer's responsibility to do this administration via the provided tool. You may modify XSL messages for use by the supported notification mechanisms/devices.

Fault Location, Isolation, and Service Restoration (FLISR)

Allowable values you may change for FLISR include:

- Change labels of any column
- Change labels of any button

Volt/VAr Optimization

Allowable values you may change for Volt/VAr Optimization include:

- Change labels of any button

Fault Location Analysis

Allowable values you may change for Fault Location Analysis include:

- Change labels of any column
- Change ordering of columns
- Change formatted string value for "Distance from Upstream Switch" column, for example from ft to yds or meters, depending on the GIS units used.

Feeder Load Management

The following describes the values that are allowable to change for Feeder Load Management:

- Change labels of any column
- Change ordering of columns

Chapter 3

UNIX Configuration

Oracle Utilities Network Management System is installed and configured on a UNIX workstation or server. The workstation or server must be properly configured before running the software. This chapter describes the UNIX configuration required for optimal use of Oracle Utilities Network Management System. It includes the following topics:

- **UNIX User Names**
- **Korn Shell**
- **Executables/Run-Times**
- **Operating System Configuration**

UNIX User Names

Oracle recommends you create a minimum of two users: one administrative user and one or more application users.

Creating an Administrative User

The administrative user, as the name implies, has central control over many critical aspects of the Oracle Utilities Network Management System. This user is the central controller of:

- Isis – configuration and starting and stopping of the Isis processes
- Oracle Utilities Network Management System services – Stopping and starting and repository of service logs
- Oracle Utilities Network Management System binaries –compiled code, configuration files.
- Database connection that has write privileges as well as read privileges
- Model-building data.

It should be noted that for data security, Oracle Utilities Network Management System tools that can be used to directly modify data are installed with permissions set so that only the administrative user is allowed to execute them.

The administrative user (e.g., nms) has access to critical components of the system. This user owns and maintains the services, the starting of the services, model building, binaries, the database, and the configuration standards. The administrative user maintains the Oracle Utilities Network Management System UNIX-based configuration and executables in one location. The administrative users Oracle environment variables ($RDBMS_USER, $RDBMS_PASSWD, $RDBMS_HOST) point to the ORACLE production tablespace owner. Thus, when services are started the user has the necessary read/write access to the production tablespace.

The administrative user:

- Owns the executable and runtime directories.
- Has read-write permissions to the production database.
- Owns the service processes (DBService, MTService, etc.)
- Performs all sms_start.ces commands.
- Performs all model builds.

> **Note:** A model build user could be created on a second machine in order to share processing load. This user should be configured in the same fashion as the administrative user with respect to database access and sms_start.ces/ ces_setup.ces access.

Creating an Application User

The application user is the standard end user of Oracle Utilities Network Management System, such as a dispatcher. This is a user who may want to run Oracle Utilities Network Management System UNIX-based tools and applications. The application user will have access to the application binaries installed in the Oracle Utilities Network Management System administrative user's directories through environment variables (such as $PATH). Application users generally have read and execute permissions for the executables and runtime directories mentioned above - with some exceptions for privileged applications. The Oracle Utilities Network Management System application user's Oracle environment variables provide a read-only Oracle user connection. Production data changes can only be made through normal Oracle Utilities Network Management System application (authorized) access operations.

The application user:

- Runs Oracle Utilities Network Management System UNIX-based applications.
- Is capable of viewing production data in read-only mode.
- Has read/execute permissions to the administrative user's runtime directories.
- Has indirect write permissions to the database through tools and applications.

Korn Shell

The Korn Shell sets environment variables and provides a command line interface to the operating system. The Korn Shell (ksh) standardizes command line execution and requests, such as running scripts, executing applications, and operating the services. The Korn Shell uses a file called .profile to configure itself. Both the administrative and application users need to have

- Their default shell set to ksh.
- The .profile configured to source the Oracle Utilities Network Management System configuration file (.nmsrc).

For your convenience, templates of a generic .profile and .nmsrc file are included in the Oracle Utilities Network Management System software distribution, under $CES_HOME/templates. These files can be copied to $NMS_HOME/.profile and $NMS_HOME/.nmsrc and then modified to suit your installation.

.profile Configuration

The Korn Shell is configured using .profile file. It is a hidden file that exists in the user's home directory. When a user logs in, this file executes, setting environment variables and defining terminal configuration. The following is required for setting up .profile.

The .profile file must source the user environment configuration file, .nmsrc. This is an easy addition to .profile. Add the following line to the bottom of .profile using any text editor.

```
. ~/.nmsrc
```

This runs .nmsrc in the current shell and initializes all of the environment variables within the .nmsrc file in the current working environment.

The .profile file must also execute correctly when called from another script, as well as when the user logs in at a terminal. Anything in .profile that is terminal-specific should be placed in an "if" clause to suppress execution if the .profile is not being run from a terminal.

```
# Set a variable to be true when .profile is
# being run from a terminal rather than a script.
#
if tty -s
then
TTY=true;
else
TTY=false;
fi
#
# Protect items that must only be run from a
# terminal and not from a script.
#
if $TTY
then
stty Compaq
tset -I -Q
PS1="`hostname`>"
fi
```

The search path environment variable, $PATH, tells the operating system where to locate the files necessary for software execution. It must include the directories that contain the Oracle Utilities Network Management System UNIX-based software. The entry in .profile will look something like the following example, where *<project>* is the application user name:

```
export PATH=/users/<project>/bin:/users/nms/bin:$PATH
```

This entry searches the user's home directory before the nms directory, letting customer specific tools and scripts take precedence over the Oracle Utilities Network Management System base executables provided.

When on a UNIX terminal, the DISPLAY variable is used to direct the windowing system to display itself on a specific screen. The syntax is:

```
hostname:display_number.screen_number
```

For example, to export the display to the machine ceshost on screen 2, the entry in .profile is: export DISPLAY=nmshost.yourdomain.com:0.2

Executables/Run-Times

The Oracle Utilities Network Management System UNIX-based software is installed in the product home directory ($CES_HOME/bin). When commands are entered at the prompt, the shell looks for the appropriate bin directory for a matching program. The PATH environment variable determines where the shell looks for the bin directory, so PATH must be modified to include the location of the Oracle Utilities Network Management System software. It is defined in the .nmsrc file located in the user's home directory and it may contain multiple path names, each separated with a colon (:). The shell parses each path name until the corresponding program is located or each path name is exhausted.

> **WARNING!** It is extremely important that the first two items in $PATH are the locations of (a) the bundled third-party software for Oracle Utilities Network Management System (/opt/oms-9.1), and (b) the location of the Oracle Utilities Network Management System software itself.

The syntax is as follows.

```
export PATH=<pathname>:<pathname>
```

For example,

```
export PATH=/opt/oms-9.1/:$CES_HOME/bin:/usr/local/bin/:$PATH
```

> **Note:** The PATH environment variable is also set in .profile for shell initialization purposes, but for this purpose it is better to modify the variable in .nmsrc. The .profile should only be edited by a competent system administrator, and a working version should always be backed up.

Operating System Configuration

A standard operating system installation will often not be optimally configured to work with an Oracle Utilities Network Management System. Sometimes the user will spawn more processes than allowed by the standard kernel configuration. Other times, a map file may require a larger data segment than the average user. Due to problems like these, you may find that you will have to tweak the operating system configuration, which may include reconfiguring the kernel or some other part of your UNIX system.

The values that are specified in this guide are examples only, as the correct values depend on how large your operating model is, how you use the system (e.g., as a server, app-server, or client) and what kind of a load is placed on the system. This section should give you an idea of how to change components of the operating system that frequently become a problem running Oracle Utilities Network Management System.

HP-UX

In HP-UX, all modifications must be made by adjusting kernel parameters and then recompiling the kernel. Use SAM (System Administration Manager) to configure the kernel in HP-UX.

1. Login as a super user. Super user login is required to run SAM.
2. Type sam at the command line.
3. Once SAM has started, select the "Kernel Configuration" icon.
4. From the new window, select the "Configurable Parameters" icon. This opens a list of configurable parameters, their current values, pending values, and a brief description.

5. Change the following parameters by double-clicking on the parameter and entering the new value in the box labeled "New Value."

Parameter	New Value	Comment
maxdsiz	0x0C000000	This results in a calculated value of 201326592.
maxuprc	150	
maxusers	45	Maxusers is a macro that will increase other parameters as it increases. It has nothing to do with how many users may login.

6. When finished changing parameters, select the "Actions" menu from the menu bar and then select "Process New Kernel" to compile the new kernel and move it into place. The changes take effect after the system reboots.

Solaris

In Solaris, limits to data segment size and the number of files available to the user are defined by the ulimit command. For the most part, these parameters do not need to be tweaked, but should you need to, you can run:

```
$ ulimit -d <datasegment size in kilobytes>
```

(Usually 256 Mb will be enough)

```
$ ulimit -n <number of file descriptors>
```

(Usually 1024 will be enough)

AIX

AIX sets its limits in a system configuration file called /etc/security/limits. You can type "man limits" from the command line for the documentation on how to modify this file. The following table describes the parameters you may have to modify.

Parameter Name	Description
Nofiles	The soft limit on the number of open file descriptors
nofiles	The hard (upper) limit on the number of open file descriptors
data	The soft limit on data segment size
data_hard	The hard (upper) limit on data segment size

Users can adjust these parameters using the ulimit command, as long as the parameters are below the hard limit configuration. If your parameter requirements are under the hard limit, you may want to consider adding the appropriate ulimit command to the .nmsrc file instead of modifying the limits file. For example, adding a line that states ulimit -d 262144 would set the data segment size limit to 256 MB; having it in the .nmsrc file would ensure that the limit is set correctly each time the user logs in.

Linux

In Linux, limits to data segment size and the number of files available to the user are defined by the ulimit command. For the most part, these parameters do not need to be tweaked, but should you need to, you can run:

```
$ ulimit -d <datasegment size in kilobytes>
```

(Usually 256 MB will be enough)

```
$ ulimit -n <number of file descriptors>
```

(Usually 1024 will be enough)

Core File Naming Configuration

Unix systems can generally be set up to save a core file if an executable experiences a non-recoverable error of some sort. Standard Unix configuration generally names this file “core” and places it in the directory where the executable was executed. The problem with this configuration is that if a core file does get generated it can happen that a second core file (from the same or different executable) can overwrite the original core file - thus hiding information that could possibly be used to better track down the source of the problem. This is not an entirely uncommon phenomenon for Unix system and there are generally Unix OS specific steps that can be taken to have the OS generate core files with process specific names - to help prevent information from being lost and make it easier to solve problems if they do occur. Below are some OS specific options for this purpose:

HP-UX

If we want the change to be 'permanent' (until the next kernel build) we can use adb - as the root user.

```
echo "core_addpid/W 1" | adb -k -w /stand/vmunix /dev/mem
echo "core_addpid?W 1" | adb -k -w /stand/vmunix /dev/mem
```

With

```
adb -k -w /stand/vmunix /dev/mem
"?" acts in /stand/vmunix as object file
"/" acts in /dev/mem
"W" writes, "X" displays the referenced symbol as hex value, "D" as
   decimal.
"$h" help page
"$q" quit
```

Solaris

As root edit /etc/coreadm.conf (COREADM_INIT_PATTERN=core.%p) or run coreadm -i "core.%p".

AIX

From your .nmsrc file:

```
export CORE_NAMING=true
```

or - as the root user - if you want to change it for the entire system. Do man on chcore for more info.

```
chcore -n on -d
```

Linux

Note the following may be the default on the Linux distributions supported by Oracle Utilities Network Management System.

```
echo "1" > /proc/sys/kernel/core_uses_pid
```

You should also check to make sure the ulimit for core files is set to unlimited - otherwise no core or a truncated core file may be created:

```
ulimit -c unlimited
```

Chapter 4

Isis Configuration

Isis is the backbone of the Oracle Utilities Network Management System. It is the messaging bus through which all components communicate. This chapter provides the details for configuring Isis. It includes the following topics:

- **Isis Terminology**
- **Isis Architecture**
- **Isis Directory Structure**
- **Isis Configuration Files**
- **Isis Environment Variables**
- **Isis Log Files**
- **Starting Isis**
- **The cmd Tool**
- **Troubleshooting**

Isis Terminology

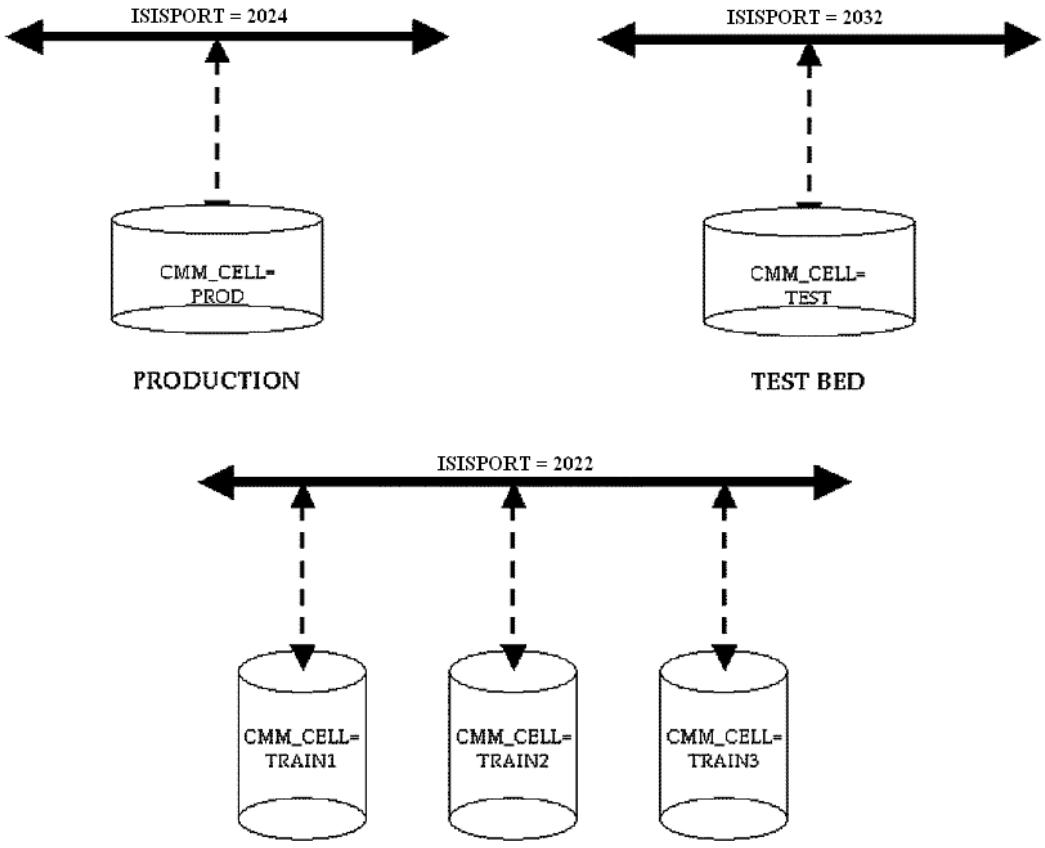
The following table describes Isis terms used in this chapter.

Term	Definition
Ports	Isis requires a set of three TCP/IP ports for communication. These are defined in the sites file. These ports are also defined in the /etc/services file. The port definitions in the /etc/services file may be overridden through the use of the ISISPORT and ISISREMOTE environment variables.
Site	Each node (client workstation or server) that runs the Isis protocol is a site. Each site has a defined number (1, 2, 3...) and a set of TCP/IP ports that are used to communicate with it, as defined in the sites file.
protos	Protos is the name of the Isis protocol process. Each Isis site has one copy of this process.

Isis Architecture

The following diagrams show an architecture in which a production system is running on a machine(s) (Services and Applications) with a specified ISISPORT, while additional Oracle Utilities Network Management System applications are run on separate ISISPORTs. The Isis process we are most concerned with is isis-protos - often referred to as simply protos. To have an operational Isis messaging backbone you must have access to at least one protos on your network (normally the same machine you are running on). Separate Oracle Utilities Network Management System applications can be run using the same ISISPORTs as long as the CMM_CELL names are unique. However, it is highly recommended that a production system retain its own individual (private) port. This is because all processes using a single protos share the same set of ports combining environments will limit the scalability of the participating Oracle Utilities Network Management System environments. Further, sharing a protos process between production and non-production Oracle Utilities Network Management System environments is not recommended.

The diagrams below show a production system on its own port and CELL, a test bed with its own port and CELL, and a training system with a single port supporting multiple CELLs for individual training environments.



Stopping an Isis protos process associated to a port (e.g., 2042) stops all Oracle Utilities Network Management System services and applications in every CMM_CELL (e.g., PROD) associated with that Isis port (2042). This does not affect any Isis processes running on other ports (2032, 2020). Stopping services and tools within a CELL does not affect the Oracle Utilities Network Management System services and tools in any other CELL.

Isis Directory Structure

The Isis directory structure is provided for verification purposes only. The structure never changes.

Isis is installed in a directory that reflects its version number, such as $CES_HOME/isis-3.4.13. However, you should make a symbolic link to that directory called "isis" to provide a common access point to the Isis directory. This link lets new versions of Isis be installed without changing any scripts. It provides a common point of reference that can be easily changed to point to a new version or reversed as necessary. To define the link, from the $CES_HOME directory, type at the prompt:

```
ln -s <directory in which Isis resides> isis
```

For example, for version number isis-3.4.13:

```
ln -s isis-3.4.13 isis
```

> **Note**: The nms_install script will automatically create the symbolic link when doing an initial system installation.

Once the symbolic link is established, the Isis runtime directory is always referred to as isis when creating scripts or setting environment variables. When a new version of Isis is installed, update the directory name and the link. The following table describes the contents of the isis directory.

Directory	Contents
bin	Isis executables, including the isisboot script, which is used to start up Isis. The cmd command resides here as well. cmd provides a command line interface to Isis that is useful for verifying connections and debugging problems.
lib	Isis runtime libraries
include	Contains Isis include files used in compiling the software

run_isis

The run_isis directory contains Isis configuration files:

- sites, which defines all of the clients of Isis, and
- isis.rc, which provides startup information for Isis.

It also contains the nohup.out log file, which contains the output from the **isisboot** command before Isis becomes functional, as well as the <site>.logdir.<port>/<site>_protos.log, where output is placed after Isis becomes functional.

Isis Configuration Files

This section addresses the files that affect the configuration of Isis software. Some of these files are Isis specific files, while others are operating system files.

Isis sites File

The Isis sites file is located in $CES_HOME/etc/run_isis. It identifies all nodes on the network that will be running Isis, assigns them a unique Isis identification/site number, and defines the TCP/IP port numbers under which they will run.

This file must be updated and consistent across all nodes running Isis in the computer network. For each entry in the sites file, a corresponding entry should exist in the /etc/hosts file in case the DNS services fail. The format of this file specifies the Isis node number, network service ports, hostname, user name, and a comment:

```
+ 001:2042,2042,2043 server1.oracle.com ces,hp9000s800
+ 002:2042,2042,2043 server2.oracle.com ces,hp9000s800
+ 003:2042,2042,2043 client1.oracle.com ces,Alphaserver
+ 004:2042,2042,2043 client2.oracle.com ces,Alphaserver
+ 005:2042,2042,2043 client3.oracle.com ces,hp9000s700
+ 006:2042,2042,2043 client4.oracle.com ces,E450
+ 007:2042,2042,2043 client5.oracle.com ces,E250
```

The leading plus sign is very important and this file cannot have any comments (except in the comment section of the end of each line). This file should parallel the entries in the /etc/hosts file. The /etc/services file should be configured with the default port numbers of the ports to be used for communications.

isis.rc Startup File

The isis.rc file is located in $CES_HOME/etc/run_isis. It contains the following information:

- Which machines may run Isis
- The number of machines that run Isis
- The Isis processes to start
- The location of Isis logs

A generic isis.rc license file is now included in the Oracle Utilities Network Management Systems software distribution. Previously each customer was provided with a customized isis.rc file, but this is no longer a requirement and the isis.rc provided with the distribution should be used instead.

/etc/hosts

The /etc/hosts file is a UNIX operating system file. It defines all of the nodes in the computer network configuration. This file must be updated and consistent across all nodes in the computer network. The format of this file specifies the Internet Protocol address, hostname and any aliases, all separated by tabs. Comments begin with a #. Also, there should be an alias provided for all machines, which is less than 15 characters, for the Isis processes; display hosts managed by the applications cannot have more than 15 characters.

```
# See the hosts (4) manual page for more information.
# Note: The entries cannot be preceded by a space.
#   The format described in this file is the correct format.
#   The original Berkeley manual page contains an error in
#   the format description.
127.00.0.1localhostloopbackloghost
200.100.100.1 server1.oracle.comserver1
200.100.100.2 server2.oracle.comserver2
200.100.100.3 client1.oracle.comclient1
200.100.100.4 client2.oracle.comclient2
200.100.100.5 client3.oracle.comclient3
200.100.100.6 client4.oracle.comclient4
200.100.100.7 client5.oracle.comclient5
```

Isis Environment Variables

Isis environment variables let client user names connect to different user environments. For example, a user might switch from a configuration environment to a model build environment. For information on the settings for Isis environment variables, see *Isis Environment Variables* on page 4-5.

ISISPORT and ISISREMOTE

ISISPORT is set to the second (tcp) service port in the sites file, and ISISREMOTE is set to the third (bcast) service port in the sites file. These environment variables override the default settings in /etc/services. These variables tell the tools and services where to listen for Isis messages.

ISISHOST

This variable specifies a possible list of nodes on which Isis will be running. It is used to configure a remote Isis system. A remote Isis system is one in which Isis and some applications run on different nodes. The applications on startup will cycle through the comma delimited list specified by this variable and will seek to make a UDP connection with the Isis process running on the remote node until it finds a valid Isis process with the same ISISPORT and ISISREMOTE values. This value can be for fail over; the applications will check each node in the list on startup, and if the first is down, they will connect with the next in the list. This is not needed or used with most implementations of Oracle Utilities Network Management System.

CMM_CELL

This variable lets different sets of services and tools exist on the same service ports. Messages received on the ports from tools and services started with a different CMM_CELL are disregarded.

CMM_CELL can be set to any value, as long as it is unique from other CMM_CELL variables running on the same service ports.

ISIS_PARAMETERS

Specifies the Isis parameter file to be referenced by applications and services on startup. An example of possible parameter file content for an Oracle Utilities Network Management System appears below:

```
#isis.prm
isis_NativeThreadStackSize 131072
# specify that all applications should provide their
# parameters when a dump occurs
isis_prmDumpAllParameters 1
# allow messages which can have 10MB of information, model
# builds may require messages of this size
isis_msgMessageSizeLimit 10000000
#
isis_UDPSndbuf             131072
isis_UDPRcvbuf             131072
#
isis_iclPacketHighWaterMark 49152
isis_iclPacketLowWaterMark 32768
# don't go below 2048
isis_iclMaxSlots            4096
#PROTOS
protos_maxLocalClients 1024
protos_maxRemoteClients 1024
```

```
protos_taskHigh 100
protos_taskLow 95
```

Isis Multi-Environment Considerations

When configuring multiple Oracle Utilities Network Management System environments, each site should be assigned unique port numbers. This logically partitions each network and prevents unwanted cross effects. As an example, the on-line system environment may be assigned to the 204x ports while the model build environment may be set to 214x, the off-line engineering environment set to 224x, and so on. While it is possible to configure all systems on the same port with CMM_CELL values differentiating the systems, it is not recommended.

Isis Log Files

nohup.out.<port>

The nohup.out.<port> log file keeps track of events while Isis is initializing. It is located in $CES_HOME/etc/run_isis. The nohup.out file contains clues if there is any difficulty in starting Isis.

<Site No.>.logdir.<port>

This is the directory where the Protos and Incarnation logs reside. The location of this directory is defined in the isis.rc file, and is typically found in $CES_HOME/etc/run_isis.

The Protos Log

protos is the Isis protocol process. This process logs its messages to $CES_HOME/etc/run_isis/<Site No.>.logdir.<port>/<Site No.>_protos.log. Check here for runtime problems with Isis.

The Incarn Log

This is a short file called $CES_HOME/oms/run_isis/<Site No.>.logdir.<port>/<Site No.>.incarn and it usually includes a single line containing the incarnation number for the particular site.

Starting Isis

isisboot

isisboot is the script that initializes Isis. It is located in $CES_HOME/isis/bin. On startup isisboot reads the isis.rc license file to determine if it can proceed. If so, it reads the sites file to determine the default network ports and site (node) identification numbers to use.

Initializing Isis

To initialize Isis, complete these steps:

1. From the ces user name type:

   ```
   isisboot
   ```

2. When complete (which will take approximately 1 minute), type:

   ```
   cmd status
   ```

This determines if Isis has successfully started and will provide information similar to the following:

```
cmd:  my_site_no = 1
my_host = tstaix01
Isis version = Isis release V3.4.12 $Date: 2007/08/2217:42:39 $
verbose mode = off
```

3. If it has started successfully, type:

```
cmd sites
```

Result: Isis lists all connected machines. For example:

```
tstaix01:cesadmin$ cmd sites
 ***  viewid = 1/1
   tstaix01.oracle.com              [site_no 1 site_incarn 3]
```

Starting Isis on Non-Default Ports

Isis may need to run on ports other than the default ports listed in the sites file. It is common to separate different sets of services by running Isis for those services on separate network ports. For example, a configuration system may run on 1601, 1602 and 1603, while the model build services run on ports 1701, 1702 and 1702. Therefore it may be necessary to switch a client from one set of Isis ports to another.

To start Isis on a non-default port, complete these steps:

1. To check which ports to use, at the prompt type:

   ```
   echo $ISISPORT
   ```

   This returns the port configured for this environment.

2. As the ces user, start isisboot with the "-p" flag. The syntax is:

   ```
   $CES_HOME/isis/bin/isisboot -p <port number>
   ```

   For example, to start Isis on ports 2052 and 2053, type:

   ```
   $CES_HOME/isis/bin/isisboot -p 2052
   ```

   Results:

   - isisboot will then create a new isis.rc file called isis.rc.<port number> from the existing isis.rc.
   - A new sites file called sites.<port number> will be created from the existing sites file.
   - New log files will also be created with the same naming convention.

The cmd Tool

Verify the connection to Isis using the cmd tool. If cmd is working, Isis is functioning as well. The syntax for cmd is:

```
$cmd <options>
```

Type cmd from the UNIX command line to bring up the command line interface, identified by the cmd> prompt. The following table presents a subset of cmd commands.

Command	Description
sites	Shows all nodes connected by Isis on the current ports: cmd>sites *** viewid = 34/5 test1.ces.com [site_no 34 site_incarn 1] test2.ces.com [site_no 33 site_incarn 1] test3.ces.com [site_no 6 site_incarn 1]
status	Provides the current status of the Isis protos. Part of the information returned is the current Isis version corresponding to the executed cmd binary.
list	Provides a list of all the Isis groups and applications connected to the protos. This identifies the CMM_CELL and process group. This can be used to identify remaining processes still running.
snapshot	Sends a message to all applications currently connected to Isis to generate an Isis dump. All the Isis related information for this process is written to disk in a log file with the process ID as the prefix (<pid>.log). This log file can be found in the run.*Service directories for services or the directory from which applications have been launched. Isis dumps are extremely useful when debugging problems, as they can tell the developer exactly what messages are being processed at the time the dump was generated.
rescan	Tells protos to update the site view.
shutdown	Causes the protocols process to shutdown. Wait 4 minutes before restarting Isis after a shutdown or an unsuccessful start attempt, and verify that all processes are completely down by checking the process list on each node (ps -aef).
Help	Print all cmd command options.
Help <command>	Print information about a specific command.

Exiting cmd

Type “quit” to exit cmd.

Troubleshooting

When an Oracle Utilities Network Management System application or Service is experiencing problems, some helpful information would include an Isis dump of the applications process, the log file associated with that application, and the output of the processes list.

Generating an Isis Dump File

To generate an Isis dump file, complete these steps:

1. On each node of concern:

```
ps -aef > $(hostname)_ps.out
```

2. Identify the process ID of the problem application(s).

```
grep -i <application> $(hostname)_ps.out
```

3. Use the following command to generate an Isis dump file for a specific process:

```
kill -USR2 <pid>
```

The process will not be affected and will continue to operate, but upon receiving the USR2 signal, it will generate an Isis dump (<pid>.log) in the directory from which that tool was launched.

> **Note:** Any subsequent USR2 messages will result in the process appending a new Isis dump to the <pid>.log file. Only the user running the applications can perform this action.

Generating an Isis Dump File for All Applications

An alternative is to issue the cmd snapshot command, which will create an Isis dump for all applications. The applications will continue to run, but every single application running will create an Isis dump file. This will clutter the file system, but it is sometimes the best way to gather all the information you need to investigate a problem.

To issue the cmd snapshot and obtain a list of all the current Isis dump files, enter these commands:

```
cd $NMS_HOME
ps -aef > $(hostname)_ps.out
touch DUMP_START
cmd snapshot
find . -name [1-9]*.log -newer $NMS_HOME/DUMP_START > ~
       /logs.txt 2>/dev/null
zip isis-dumps.zip 'cat ~/logs.txt'
```

This set of commands will find all the .log files that start with a number and were generated after the time the DUMP_START time-stamped file was created. It will create a file called `isis-dumps.zip` that can be sent back to Customer Support for investigation. With the full set of logs, Customer Support can track all the interactive messaging for problem investigation and resolution.

Reporting a Problem to Customer Support

In general, when reporting a problem to Customer Support, the following information can speed the problem identification and resolution process:

- An explanation of what the observed symptoms were and where they occurred.
- An explanation of how to repeat the problem, if possible.
- An explanation of expected behavior.
- A specific time frame when the problem occurred.
- Example data demonstrating the problem (e.g., event numbers, crew names, etc.).
- Service logs and environment log files of the affected Services/Applications.
- Isis dumps of the affected application and services at the time the problem was observed. A complete Isis dump of all processes may be requested if the problem is repeatable, along with a process list output file.
- The core file, if one exists for the process.
- Any other activity that occurred prior to, or concurrent with, the issue that may stand out as a possible contributor.

Chapter 5

Database Configuration

Oracle Utilities Network Management System currently supports the Oracle Relational Database Management System (RDBMS). The RDBMS must be properly installed and configured prior to using the Oracle Utilities Network Management System software. This chapter provides the configuration requirements for Oracle. It includes the following topics:

- **Oracle Installation Guidelines**
- **Oracle Tablespaces**
- **Oracle Users**
- **Starting Oracle**

Oracle Installation Guidelines

It is recommended that Oracle Enterprise Edition be installed. Please see the Oracle RDBMS installation documentation for specific Oracle installation requirements.

Oracle Tablespaces

Every Oracle Utilities Network Management System must have its own Oracle tablespace set. In general, the tablespaces consist of the following:

Tablespace	Description
Production	The production tablespace (ces_db) contains all of the production data for Oracle Utilities Network Management System. This includes model data, outages, and data that is produced by operations performed in Oracle Utilities Network Management System.
Production Temporary (Optional)	The production temporary tablespace (ces_tmp) temporarily stores operating data prior to insertion into the production tablespace. The default Oracle TEMP tablespace should be the designated temporary tablespace for the system. Oracle is more efficient when managing temporary data in this way. Make sure that a sufficient amount of space is allotted to TEMP.
Production Index	The production index tablespace (ces_idx) contains all of the indexes for the production tablespace. The ces user's .nmsrc file must contain the CES_INDEX_TABLESPACE environment variable referencing this tablespace.

Tablespace	Description
Customer Data	The customer data tablespace (<*project*>_customers_db) belongs to the customer. It is populated with the entire customer database by the CIS extraction process. Public synonyms are assigned to the customer tables and selectability is granted to production Oracle users so that the necessary table joins can be created.

Each tablespace should be located on a separate disk to enhance performance and decrease bottlenecks due to high volumes of input/output.

It is key that the tablespaces are provided with sufficient disk space and are monitored regularly for growth. When a tablespace runs out of disk space, operational data will be lost and Oracle Utilities Network Management System services will discontinue to function properly.

Oracle Instances

For performance, scalability and simplicity there is normally only one Oracle instance on a production machine. It is not generally recommended that a production machine have multiple Oracle instances on the same machine. An exception would be where a cluster is used; you may want an Oracle instance installed on both sides of the cluster (production on the primary side, Model Build, Test, or Oracle Utilities Business Intelligence on the secondary side). Under normal circumstances there would only be one instance of Oracle on each side – if one side of the cluster fails you could end up with two instances on the surviving node. In general, try to keep it simple.

You should consult with your Oracle Utilities Network Management System Professional Services technical team to develop a creative solution to meet your specific needs.

Oracle Utilities Network Management System uses three environment variables that are set in the $HOME/.nmsrc file to create a connection to the Oracle database. These are:

- RDBMS_USER – Oracle user that owns the tablespace where the data will be stored:
- RDBMS_PASSWD – Password for the RDBMS_USER as defined in Oracle.
- RDBMS_HOST – Instance name for the Oracle connection

If Oracle tablespaces for different Oracle Utilities Network Management System implementations occupy the same Oracle instance on a machine resource, then the RDBMS_USER and RDBMS_PASSWD environment variables must be different. The user-password pair RDBMS_USER/RDBMS_PASSWD generally owns a complete set of Network Management System tables that are used for a single Oracle Utilities Network Management System environment – and are often created in a "Network Management System instance specific" set of tablespaces – though this is not required. If two separate Oracle Utilities Network Management System environments attempted to use the same RDBMS_USER/RDBMS_PASSWD combination, the databases would likely become corrupted. This is a common mistake. Be aware of the shell you are using, the environment variables, and their values.

> **Note:** Two or more Oracle Utilities Network Management System instances on a single machine can be acceptable (depending on machine resources) for testing, training and model build environments.

It may be necessary to tune Oracle for the specific environment it will be operating on. Typically a qualified DBA can perform the necessary tuning and modifications. Often this is an iterative process that requires running the full Oracle Utilities Network Management System on the production machines and capturing statistics for analysis.

Other Environment Variables

Other Oracle-specific environment variables may need to be different between systems, but these are due to how the DBA has constructed the environments. Other than the NLS specific environment variables noted below, these are listed in one of the example tables in chapter five.

When Oracle is loaded onto a given platform, the Oracle instance itself will generally have a default National Language Support (NLS) setting. Oracle Utilities Network Management System client applications (like DBService) which utilize the Oracle Call Interface (OCI) need to know what NLS settings to use for inserting and interpreting result sets from Oracle. Presently, the easiest way to do this is as follows:

1. Add the following environment variable to your .nmsrc file: NLS_LANG

   Note: For a US configuration, Oracle believes the NLS_LANG environment variable (as far as OCI is concerned) typically defaults to AMERICAN_AMERICA.WE8ISO8859P1. Thus if a customer sets their Oracle NLS to something other than this value (inside of Oracle during instance setup) - and does not specify the NLS_LANG environment variable to appropriately match, DBService will not start. You will see a note in the DBService log file indicating a mismatch that must be rectified.

2. The following process should work for setting NLS_LANG:

   ```
   Set NLS_LANG to
   <NLS_DATE_LANGUAGE>_<NLS_TERRITORY>.<NLS_CHARACTERSET>
   ```

   where each "NLS component" needs to match the values returned by this query:

   ```
   SELECT * FROM v$nls_parameters WHERE parameter IN
     ( 'NLS_DATE_LANGUAGE',

   'NLS_CHARACTERSET', 'NLS_TERRITORY' )
   ```

   For example, we have NLS_LANG=AMERICAN_AMERICA.WE8ISO8859P1, and our query returns:

   ```
   PARAMETERVALUE
   ----------------------------
   NLS_DATE_LANGUAGEAMERICAN
   NLS_TERRITORYAMERICA
   NLS_CHARACTERSETWE8ISO8859P1
   ```

3. Set the ORA_NLS10 environment variable. For example:

   ```
   export ORA_NLS10=/users/oracle/product/10/nls/data
   ```

   (or wherever your valid Oracle nls/data directory is located)

Oracle Users

Once the tablespace is established, you must create users and grant their permissions. Oracle users are those users that have access to the Oracle tablespaces. Before defining the users, it is important to discuss the security role that a user can possess.

Security Roles

Security roles determine the level of database operations that a user can perform. There are two types of security roles:

Role	Description
ces_rw	Read-write role. This role has read and write privileges to the production data. It can create, drop, update to, and insert to, all of the production tablespace objects.
ces_ro	Read-only role. This role can only connect and select data from the production tablespace objects. **Note:** Certain security tables, such as ces_users, are excluded from the view of the ces_ro role.

Users

There are three Oracle users. Each user directly relates to the tablespaces. Substitute specific customer name for <project> where noted below.

User	Description
<project>_CES	The <project>_ces Oracle user is the owner of the production tablespace. This user has a ces_rw role and maintains full control of the data elements in the production tablespace.
<project>	The <project> Oracle user is the application user. This user has a ces_ro role to the production tablespace.
<project>_customers	The <project>_customers user has full privileges to the customer data tablespace only and no privileges on the production tablespace.

Starting Oracle

Complete the following steps to start Oracle:

1. Login as oracle. If logged in as the root user, the system will not request a password. At the prompt, type:

   ```
   su - oracle
   ```

2. Login to SQL*Plus:

   - As the oracle user, type:

     ```
     sqlplus /nolog
     ```

   - At the SQL> prompt, type:

     ```
     connect / as sysdba
     startup
     quit
     ```

3. Start the listener. As the oracle user, type:

   ```
   lsnrctl start
   ```

Note: The tnsnames.ora and listener.ora files must be properly configured to start the oracle listener. The location of these files may vary by system, but they must be consistent on all machines requiring connections via SQLNET.

4. Login as the distribution user and test the connection to Oracle. At the prompt, type:

```
ISQL.ces
```

This references the RDBMS_USER, RDBMS_PASSWD and RDBMS_HOST to establish the connection to the database. If this connection is successful, a SQL> prompt will appear.

Chapter 6

Environment Configuration

Many problems that occur during an initial installation and setup of an Oracle Utilities Network Management System result from improperly defined environment variables and a misunderstanding of their usage and impact. This chapter describes the environment variables, their standard settings, and where they are located. This information should help you avoid a number of problems.

Because of the innate flexibility allowed by environment variables, there are an infinite number of permutations you can apply for a system setup. Not everything you can do with these variables should be done. This chapter describes the suggested settings that you should adhere to in order to avoid confusion.

This chapter includes the following topics:

- **System Resource File**
- **Modifying Environment Variables**

System Resource File

The System Resource file ($HOME/.nmsrc) houses the environment variables that enable the Oracle Utilities Network Management System to operate correctly and consistently. They define the connections information for the database and ISIS, as well as environment specific configuration settings such as viewer symbology, application geometry, and more.

You will need to modify the System Resource file in part for application to specific systems. One suggestion is to use environment variable dependencies. By doing this you can simplify the process of changing values; by changing one variable that is a root dependency, the change will cascade through a number of others, limiting your required changes and maintaining consistency throughout the file.

Modifying Environment Variables

To modify the environment variables, complete these steps:

- Modify the variable you want to change with the new settings in the .nmsrc file.
    - Type .**nmsrc** at the prompt to source the file in the current working environment. The new variables replace the old variables.

    Note: New variables replace the old variables when the file is sourced. You should source .nmsrc each time you change the file. The file .profile automatically sources .nmsrc at startup.

Environment Variables

The table below lists the required environment variables and their standard settings that must be modified depending on the type and number of environments you are constructing. See templates/nmsrc.template for more variables. Other variables may be added as well, depending on the functionality of your system. This is not an exhaustive list, but it does address the variables typically required to start an Oracle Utilities Network Management System.

Environment Variable	Example Setting	Description
NMS_ROOT	/users/nmsadmin	Provides a common location to place the base Oracle Utilities Network Management System directories and files owned by the administrator. It is recommended that you set this to the home directory of the Oracle Utilities Network Management System administrator. By specifying this directory correctly, you can use it to simplify other installations. When this value changes, the change will be cascaded throughout the other dependent environment variables. This environment variable is used by a number of scripts and processes.
CES_HOME	$NMS_ROOT/nms/ product/1.9.0.2	This environment variable is set to the product installation directory for the active installation.
NMS_HOME	${HOME}	The nmsadmin username home directory. This is the directory where the implementation directory and runtime directories exist. For Release 1.9.0.2 this must be set to the nmsadmin username home directory.
NMS_PROJECT	config	This is the project name, can default to "config" or will match the customers project name (e.g., OracleLite). This is the name that is immediately to the left of the "product" name in the CES_SITE environment variable.
NMS_CONFIG	$NMS_HOME/ $NMS_PROJECT	This is the location of the project configuration and implementation files. The name (i.e., config) must also match the CES_SITE variable on the left side ("config product ces") and exactly match the NMS_PROJECT environment variable.

Environment Variable	Example Setting	Description
CES_DATA_FILES	$CES_HOME/sql	Defines the location of data files used in various scripts and routines that define aspects of system configuration. This variable must be defined and can be accessed from a number of scripts. The standard location for these files is the $CES_HOME/sql directory. Examples are ces_classes.dat, ces_inheritance.dat, and ces_devices.cel.
CES_DAYS_TO_LOG	5	Identifies how long to store the old log files. When services are restarted, log files older than the set number of days (5 in this case) will be removed.
CES_INDEX_TABLESPACE	ces_idx	Used by the ISQL.ces process. It will parse SQL scripts that create indexes and make sure that the index is actually created in the specified tablespace name. This tablespace must be owned by the RDBMS_USER. The practice of separating indexes from operational data improves Oracle performance.
CES_LOG_DIR	$HOME/log	Defines for services and login environments where to place the resulting log files. Since log file generation requires write access for a process, the user who started the process must own this directory. To this end, the environment variable must be $HOME/log. You may need to manually create this directory for each new user (application or administration). It is highly recommended that this directory be located on a different filesystem from $CES_HOME
CES_MASTER_VIEWER	"VIEW;0;1"	Defines the process name for the Viewer that is to be designated as the "Master Viewer". This is the Viewer that will receive all the load messages from **View** buttons on tools like the Work Agenda. Typically, this is the first Viewer started from the Environment Manager. This will let the **View** button//action from other windows designate a specific Viewer for loading new maps, rather than changing the current view in all the running Viewers in an environment.
CES_SITE	Project specific. Example: <project> product ces	Defines the configuration inheritance path for a system. When the setup process executes, it searches this site variable from left to right looking for configuration files with prefixes that match the value in the site variable. This feature lets you inherit or override the ces or product configuration. This variable is used by most of the configuration scripts and will be specified by Oracle.
CES_SYSDATE	Environment specific. Example: %D%R|%D|%R	Defines the display format for which all applications will display date and time elements. The format for this requires specifying 3 formats: date and time | date | time The three formats specified in this environment variable must also be added to the $DATEMSK file.

Environment Variable	Example Setting	Description
CMM_CELL	Environment specific. Will be specified to some unique value for each system. Example: production	Allows for encapsulation of Isis messages within a specific group of the same CMM_CELL specification. All applications that join up and connect to a specific set of services must have the same CMM_CELL, as well as ISISPORT and ISISREMOTE variables.
DATEMSK	$CES_HOME/etc/ces_datefmt	This file will be generated and updated by Oracle. It defines all the expected date formats that can be encountered as input by widgets and Services. Services will use the values in this file, for example, as a format dictionary when given call time as part of a trouble call. Expected time formats should be placed near the top of the file so that the search and compare algorithm encounters the most likely values as quickly as possible.
ISIS_PARAMETERS	$CES_HOME/etc/run_isis	Identifies which file to reference for Isis parameters. This must be established before initiating an application.
ISISHOST	System specific. Usually the machine name of the database server.	Used for systems that are expected to be applying remote Isis connections. Basically an Isis set of protos processes are started on a designated service; when applications are started on a system without protos running, they will use this variable to identify the machine where protos is running and connect with the correct Isis processes. While Isis can run on multiple machines, a rule of thumb is to limit the number of sets of protos running to 5 or less. If you have more than 5 sets running on the same ports, CMM_CELL can encounter communication issues.
ISISPORT	System specific, the default should be 2042.	A TCP/IP connection port on which Oracle Utilities Network Management System processes communicate (via Isis).
ISISREMOTE	System specific, the default should be 2043.	A TCP/IP port used when you are making a connection to a "remote" protos. This can either be when the process is running on a machine without protos or if a local connection is attempted and all the local connections are filled.
MB_META_HOSTS	System specific. Example: AIX.0057F8F4C00	The value presented in <architecture> defines the O/S system on which the binary version of the data maps is built. These binary maps are O/S specific and are built from the system independent text version that resided directly in the OPERATIONS_MODEL directory. The value used in place of <architecture> is determined by the following UNIX command: $(uname).$(uname -m | sed -e "s/\//-/g")
OPERATIONS_RDBMS	System specific. Example: ces_db	Identifies the primary tablespace for the operations data.
ORACLE_HOME	System specific: Example:/usr/users/oracle/product/11	Identifies the home directory for the Oracle user. This is necessary to simplify other variables dependant on this path.

Environment Variable	Example Setting	Description
ORACLE_SID	System specific. Example: PRODSERV01	Identifies the Oracle session ID value. Either this variable or the SERVICE_NAME variable must be defined.
PREFERRED_ALIAS	Model specific. Example: OPS:PSU	Defines what alias of a device is to be displayed by default. In the example, the system will display the alias that has a DB_TYPE of OPS as found in the alias_mapping table. If an alias with a DB_TYPE of OPS does not exist, then the PSU (pseudo) alias will be displayed. This, by convention, is a unique name of <class_name.device_idx>. Depending on the model build definition, you can use and define other alias options, such as a SCADA alias.
RDBMS_HOSTS_DIRECT	System specific. Example: "PRODSERV01.world PRODSERV02.world"	A list of direct hosts for which ISQL.ces will make direct connections. This variable is not necessary for the majority of Oracle Utilities Network Management System applications, since most H/A systems will employ cluster solutions.
RDBMS_HOST	System specific. Example: PRODSERV01.world	Identifies the host machine for establishing an sqlnet connection via Oracle. This value must exist in the tnsnames.ora file on the system attempting a connection. This is required for the use of many setup scripts and ISQL.ces.
RDBMS_PASSWD	System specific.	The password used to establish a connection to the operations database. This is related to the $RDBMS_USER variable.
RDBMS_USER	System specific	The user name used to make a connection with the Oracle tablespace. For the production server, this name would correspond to the user with read/write access to the database, but for application servers this would be the name for the user with read-only access to the database.
SERVICE_NAME	System specific	The service name of the Oracle database that the system should connect to. Either this variable or the ORACLE_SID variable must be defined.
SYMBOLOGY_SET	System specific. Example: $OPERATIONS_MODELS /SYMBOLS/ PRODUCT_SYMBOLS.sym	Identifies the primary symbology file loaded by the Viewer. This file identifies the Viewer symbols for all objects.
VIEW_GEOMETRIES	System specific. Example: "1024x744+0+0,\ 512x384+0+384, \512x384+512+384,\512x384+0+0,\ 512x384+512+0"	Defines the start up geometries of the Viewers from the Environment Manager on the screen where an Environment Manager is mapped. The first value corresponds to the default size of the initial large Viewer. The other four settings define the sizing for the four smaller viewers started in succession from the Environment Manager. The format for the setting is: WIDTHxHEIGHTxXPOSxYPOS where the values are in pixels and the XPOS and YPOS refer to the top left position placement of the Viewer window.

Environment Variable	Example Setting	Description
VIEW_GEOMETRIES_NO_EMAN	System specific. Example: "1024x768+0+0,\ 512x384+0+384,\ 512x384+512+384,\ 512x384+0+0,\ 512x384+512+0"	Identifies the Viewer geometries for the screen where no Environment Manager GUI is mapped. For two (or more) screen systems, these are the settings that define Viewer appearance on the other screens. The format is the same as VIEW_GEOMETRIES.

Chapter 7

Services Configuration

The configuration of Oracle Utilities Network Management System services involves establishing the location of system services on server nodes in the computer network and defining their configuration and command line options.

This chapter includes the following topics:

- **Services Overview**
- **Service Alert Email Administration**
- **Service Alert Printing Administration**
- **Services Configuration File**
- **Model Build System Data File**
- **Starting and Stopping Services**

Services Overview

Oracle Utilities Network Management System services provide memory-based model management for RDBMS persistent electrical network model information - generally to support real-time access and performance objectives. The services maintain the memory resident data model for the real-time status of the electrical network. The memory model caches the necessary data to build the model from relational database tables. The services then solve this model (fills in the blanks, determine what is energized, grounded, looped, etc.) and optimize the result for client access. Each service generates and passes appropriate incremental model updates to ISIS (the Network Management System real-time publish/subscribe message bus) for publication. Interested applications subscribe to the published messages keep the Network Management System end users up to date with current state of the model.

Startup scripts that run when the operating system boots can be used to automatically start the Oracle RDBMS, ISIS, and Oracle Utilities Network Management System services. How you configure and where you place these scripts is based upon startup (default) Unix "runlevel" and your platform. For Linux platforms you can generally determine your current runlevel via:

```
/sbin/runlevel
```

For Linux startup/shutdown scripts are generally located in the /etc/init.d directory. A Unix softlink to each startup script to run for a given runlevel is generally made in the /etc/rc<run_level>.d directory. Other Unix operating systems have similar but often slightly different conventions. It is presently an exercise left to the system administrator to properly create and configure startup scripts that will properly run on startup for a given Operating System. Example scripts for some common startup scripts may be found in the $CES_HOME/templates file. These scripts are examples only and will need to modified/reviewed/tested locally to ensure they work properly for a given installation.

Oracle Utilities Network Management System Services are generally flexible and attempt to cater to the functional needs of various utility clients through the use of command line options and run-time parameters stored in the relational database. Below is a brief summary of the primary Oracle Utilities Network Management System Services. Details about available command line options and relational database parameters specific to each service can be found in the $CES_HOME/documentation/services directory.

SMService - System Monitor Service

SMService monitors the core Oracle Utilities Network Management System service and interface processes. It reads and caches the appropriate system.dat configuration file to determine which processes to initiate and monitor. In the event that a managed process fails, SMService restarts it based on the cached configuration data from the system.dat file.

The following variations of system.dat files should be located under $CES_HOME/etc. There should be *.template versions of these files in the $CES_HOME/template directory. These configuration files generally define the specific run-time executables and command line options necessary for a given Network Management System installation:

- system.dat.init - defines configuration required for initial setup.
- system.dat.model_build - defines minimum configuration required for initial model builds.
- system.dat file - defines configuration for fully operational Network Management System.

sms_start.ces will launch SMService, which in turn will cache the $CES_HOME/etc/system.dat file by default and then launch the remaining services, interfaces and adapters as defined by the $CES_HOME/etc/system.dat file. The following command sequence can be used to specify an alternate system.dat type file:

```
sms_start.ces –f  ~/etc/my_system.dat
```

The smsReport tool can be used to request and monitor the SMService view of the processes it is currently managing. smsReport is a non-GUI tool used to report the state of the system by querying SMService. It is executed in either one-shot or monitor mode. One-shot mode is the default mode that queries SMService for the current state and displays it to the user on exit. However, if the system state is INITIALIZING, then smsReport automatically switches to monitor mode so as to not exit prior to initialization completing before exiting. Monitor mode is set by starting smsReport with the -monitor command line option. It serves the same function as the default one-shot mode but does not close after the system state has been reported.

To shutdown the Oracle Utilities Network Management System (gracefully) use the following script:

```
sms_stop.ces
```

The sms_stop.ces script will shutdown all of the user environments (one at a time) and then the services in reverse order to how they where defined to startup in the ~/etc/system.dat file. Using this script generally prevents certain deadlock conditions which can occur if an attempt is made to stop all user environments and system services at the same time.

DBService - Database Service

DBService provides database access for any processes attached directly to the ISIS message bus within the Oracle Utilities Network Manage System environment. The messaging backbone, ISIS, directs database queries and commands to the appropriate Oracle RDBMS server and returns results to the requesting process.

> **Note**: A given instance of DBService allows a configurable number of queries to occur in parallel but serializes RDBMS updates. By assigning update responsibility of specific tables to specific DBService instances (by convention) parallel updates can be supported which generally increases performance and/

or scalability under system load. TCDBService, MBDBService are examples of this strategy.

ODService - Object Directory Service

ODService registers new objects as well as caches configuration and (optionally) run-time information that is likely to be requested by applications in a particular form and/or on a regular basis. This caching allows the requests to be handled very quickly without directly accessing the database. Cached information is primarily static configuration data, such as object classes, class hierarchy, symbology assignments and (optionally) device alias information.

DDService - Dynamic Data Service

DDService manages real time (dynamic) information required by the system. In addition to command line options DDService utilizes the srs_rules table for run time options.

Examples of dynamic data that DDService manages include:

- Current status of switchable devices
- Special operating conditions of devices (tags, crews, notes, etc.)
- SCADA information (analogs, digitals, quality codes)
- Operating authority (users and control zones)

When you make changes to Oracle Utilities Network Management System control zones (control_zones and/or control_zone_structures tables), you need to tell DDService to update its internal control zone memory structures with the following UpdateDDS command:

```
UpdateDDS –recacheZones
```

When you make changes to SCADA device definitions, you can tell DDService to update itself with the following UpdateDDS command:

```
UpdateDDS –recacheMeasures
```

MTService - Managed Topology Service

Real-time electrical systems are in a constant state of flux of electrical flow. A single device operation could de-energize a model section, create a parallel on one or more phases, ground one or more phases, create a loop condition, or extend some other form of energization/ deenergization. Since the topological state (i.e., energization, ground status, energizing feeder, feeder color, etc.) of a device cannot be accurately determined without taking into account a large number of other devices and operating conditions, it is not possible for each application to independently determine current topological states. Instead, MTService maintains a complete topological copy of the model in memory, which it updates as devices and conditions change. It publishes topological impact updates and services topological data requests from other Network Management System applications and services.

JMService - Job Management Service

JMService is the customer trouble call analysis engine. It relies on MTService to trace device connectivity when determining probable outages in the system. Customer complaints (trouble calls) are fed into the system and JMService groups them using configurable rules to compute and publish the most likely cause of the problem. JMService also manages restoration resources (crews). In addition to command line arguments, JMService uses the srs_rules table for the majority of its run-time configuration options.

TCDBService - Trouble Call Database Service

This is a copy of DBService that runs specifically to improve the performance of JMService by handling database calls for JMService. This lets the main DBService manage database requests from operator activity not directly related to trouble calls.

MBService - Model Build Service

MBService is used in building a data model, which mirrors the customer's existing data model (generally extracted from a Geographic Information System such as ESRI, Intergraph, SmallWorld, or AutoCAD). When changes are made in the GIS a project-specific extractor is used to extract and transform GIS changes into a standard Network Management System format. MBService takes the standardized input, parses and integrates the resulting changes into the Oracle Utilities Network Management System electrical network model. In addition to maintaining the model database, MBService also generates map files, which are optimized for use with Network Management System graphical viewing tools.

SwService - Switching Service

SwService helps manage switch plan state transitions and provides a facility for sending updated plans to interested parties via e-mail.

MBDBService - Model Build Database Service

MBDBService serves the same purpose for MBService as TCDBService does for JMService. It is a copy of DBService that runs specifically to improve the performance of MBService by handling the database calls resulting from model building. It only applies if you use the -mbdbs command line option when starting MBService. This option bypasses DBService and uses MBDBService to perform queries and SQL commands.

MQDBService - MQService Gateway DBService

MQDBService provides direct access to the database for the MQSeries Gateway. This reduces competing throughput for the DBService reserved for operator interactions.

PFService - Power Flow Service

PFService manages real-time operations power flow calculations that allow you to view the complex voltages and currents at points and devices in the electrical network model. These calculations are performed on an electrical island basis by tracing from each energized source and collecting all the energized objects. SCADA measurements at the feeder head and at various points down the feeder are used to accurately distribute load to each load point. PFService sends the real-time power flow solution results, as well as information about voltage and flow violations, to various Oracle Utilities Network Management System windows for you to view.

CORBA Gateway Service

The CORBA Gateway service provides part of the interface between the Java-based applications such as Web Trouble, Storm Management, Web Call Entry, etc. and the other C++-based Oracle Utilities Network Management System services. The CORBA gateway allows the Java Application Servers to get published updates from services like JMService, DDService or MTService and also provides the mechanism to query these services directly on-demand. The Java Application Servers (JBoss or WebLogic) must then take these updates and make them available for the Java (end-user) client applications.

The CORBA Gateway service uses ISIS to communicate with the other Oracle Utilities Network Management System services. The CORBA Gateway service requires that the TAO (TheACE

ORB) CORBA Naming Service be running. Normally TAO is configured to run (be default) on startup. See the $CES_HOME/templates/tao.template script for an annotated example of a tao startup/shutdown script that could be configured to run at system startup.

> **Note:** With Oracle Utilities Network Management System 1.9.0 we now recommend that you run two copies of the CORBA gateway for each Oracle Utilities Network Management System environment.

1. The first instance is a dedicated publisher instance that takes messages published via the Oracle Utilities Network Management System services and publishes them the Java Application Server (JBoss or WebLogic).
2. The second instance is dedicated to handling Java client application requests to Oracle Utilities Network Management System services.

   Examples of how to setup these corbagateway instances can be found in the $CES_HOME/templates/system.dat.template file.

The JBoss or WebLogic Java Application Server must be configured to correctly connect to the appropriate corbagateway(s). See the Oracle Utilities Network Management System installation guide for instructions on configuring the Java Application server.

If you need to run two or more Java Application Servers (JBoss or WebLogic instances) to support two or more Oracle Utilities Network Management System environments on the same UNIX machine you will need a corresponding number of IP addresses. One IP address is needed to support the appropriate instance of the Java Application Server for each Oracle Utilities Network Management System environment. This can be accomplished in one of two ways:

1. If you have two or more (available) Ethernet ports on your server (or can add additional cards), attach that port to the network and assign it a new IP address.
2. On most UNIX systems, you can add a second IP address – known as an "alias" – to your single interface. You will need a second valid IP address for your network in order to do this.

Service Alert Service

Service Alert processes updates from other services such as job/event update information, device operations, as well as receiving notifications from database triggers. These "updates" serve as the triggers for Service Alert to determine when the criteria for sending out a notification have been met. Once triggered, Service Alert gathers relevant data and sends out the desired notifications.

Service Alert Email Administration

How Service Alert Email and Paging Notification Work

When initiating a notification, Service Alert sends email and paging requests to the CORBA gateway. It is the email toolkit code within the CORBA gateway that interfaces with a mail system. The email toolkit uses SMTP to send these message requests. * Therefore, to properly receive Service Alert notifications, an SMTP server needs to be configured and running on the network. All that is left to do is to describe to the email toolkit the configuration settings that it needs in order to communicate with the SMTP server.

> **Note:** Pager notifications are also sent by SMTP, since most major paging providers allow messages to be sent to a pager via an email aliasing system.

Entering Email/Pager Configuration Settings

The following UNIX environment variables need to be set up properly in order to configure the email/pager notifications.

Variable	Description
CES_SMTP_SERVER	This is the fully qualified network hostname of the mail server.
CES_DOMAIN_SUFFIX	Domain Suffix – This value should be a valid domain such as "oracle.com". This value is used in constructing the domain portion of the "From" field for all outbound messages. This field is also used during SMTP communication between the CORBA gateway and the mail server. It is important to set this to a valid domain, as some SMTP servers will verify that the domain exists and is real. If the server does not believe that the domain is legitimate, the email message may be discarded

The **Email Username** setting is a command line parameter on the CORBA gateway. The username is the string that appears after the "-username" command line option. This will appear in all email and pager notifications "From" field. It is probably a good idea to set up an email alias for this username, in case notification recipients attempt to reply to a notification. Note that the "@domain.com" portion of the username should be omitted as this comes from the "CES_DOMAIN_SUFFIX" Environment variable.

Service Alert Printing Administration

After installing the Oracle Utilities Network Management System Web Gateway, three sets of configuration steps need to be performed to allow printing from Service Alert, as described in the following paragraphs.

Adding Printers for Service Alert

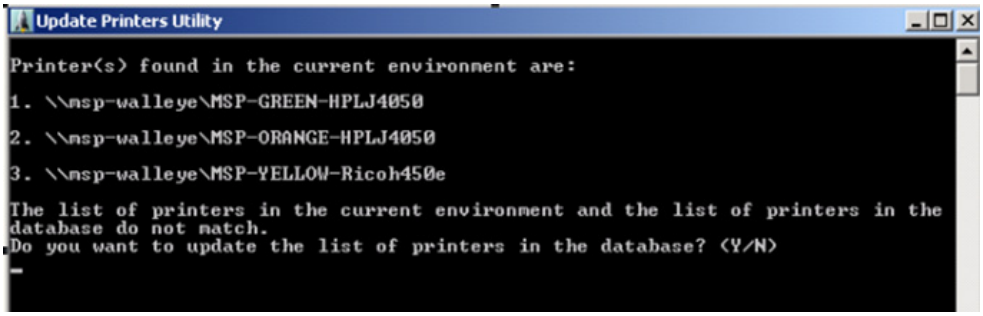
A UNIX System Administration will need to add the printers/queues to the UNIX server where the Service Alert application is executing.

Using the Update Printers Utility

The list of available printers will be kept in the MYC_PRINTERS table in the Oracle Utilities Network Management System database. Running the "UpdatePrinters" utility will populate this table. This utility is installed with the base Oracle Utilities Network Management System. UpdatePrinters will look at the current user's Windows environment to obtain a list of available printers. If the list of printers in the current environment does not match the contents of the MYC_PRINTERS table, the user will be asked if he would like to synchronize the table with the current Windows environment. Also, for any contact that does not have a known printer, the tool will give the user the opportunity to change the printer location for that contact.

The image below shows an initial run of the Update Printers utility. Before running the utility, three printers were added to the current environment. The utility finds these printers in the

current environment, and it prompts the user to determine if they wish to add the list of printers to the database.

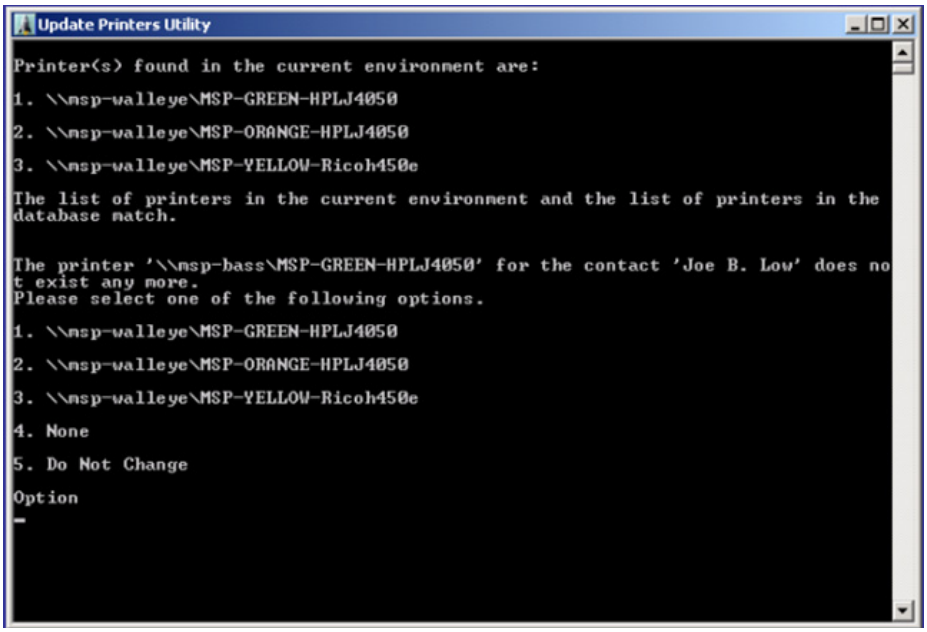


The image below displays another run of the Update Printers utility. In this instance, the utility has determined that:

- The list of printers in the current UNIX environment and the list of printers in the database are the same.
- The contact in the database with name "Joe B. Low" has an associated printer "\\msp-bass\MSP-GREEN-HPLJ4050" that is not listed in the database table that contains the list of all "known" printers.

The Update Printers user can choose to:

- Set the printer associated with contact "Joe B. Low" to one of the known printers (options 1 – 3).
- Reset the printer for contact "Joe B. Low" to "None" (option 4), in which case the contact will no longer receive printer notifications.
- Leave the printer associated with contact "Joe B. Low" unchanged (option 5), in which case an attempt to send a printer notification to this contact will fail.



Services Configuration File

The Services Configuration Data File (system.dat) configures services for operation. It determines how services are defined, which default flags to use, on which computers, and how long the waitfor timer runs. The system.dat file is located in the $CES_HOME/etc directory.

There are a number of sections in the system.dat file. The most critical sections include:

- scripts
- server
- services
- applications
- program
- instances

Scripts

The following table defines the scripts that SMService uses to perform various tasks.

Script	Description

LaunchScript	Used to launch a service. The most widely used mechanism for starting all the services is: sms_start.ces
	The default script to start a single service is sms_start_service.ces. Its syntax is:
	`sms_start_service.ces <host> <service> <process> <options>`
	host – Name of the machine on which to run the service
	service – Name of the service
	process – Name of the executable that launches the service
	options – Command line options that are passed to the process at initialization
	For example, to start DBService, type:
	`sms_start_service.ces train1 DBService  DBService -nodaemon`
	Define the launch script in system.dat as follows:
	`LaunchScript  <script name>`
	If no script is specified, then sms_start_service.ces is assumed.

Script	Description

Notify Script	Announces an event. This script eliminates the need for an ISIS tool as an announcer. It can be used to generate e-mails and logs, or to interface to paging systems. When developing this script, keep in mind that it does not connect to ISIS. The syntax is:
	
	`<script name> <time> <host> <process> <event type> <system state> <old system state> <message>`
	
	time – Date/time stamp.
	
	host – Name of the machine on which the processes are running.
	
	process – Name of the process.
	
	event type – The process state. Valid values are:
	
	STARTING – The process has started.
	
	INITIALIZING – The process has registered and is initializing.
	
	RUNNING – The process reports as initialized.
	
	FAILED – The process has failed.
	
	FAILED_INTERFACE – The process reports a failed interface.
	
	STOPPED – The process intentionally stops.
	
	INFO – The process generates a progress report.
	
	system state – State of the system. Valid values are:
	
	INITIALIZING – SMService is launching processes from system.dat.
	
	NORMAL – All processes are running or are intentionally stopped.
	
	WARNING – A non-critical process has failed. This state also refers to failed critical processes that have another instance running.
	
	CRITICAL – A critical process has failed and there are no other instances running.
	
	old system state – State of the system before the event generating the announcement occurred.
	
	message – Message supplied by SMService or the process that caused the event.
	
	Define the notify script in system.dat as follows:
	
	`NotifyScript <script name>`
	
	There is no default value, so if a script is not defined here, then only ISIS announcements are generated.

CoreScript	SMService looks to this script for instructions when a core file is detected. This script determines what should be done with the file, such as announce the existence of the file, delete it, archive it, or e-mail the administrator. It does not connect to ISIS. Its syntax is:
	
	`<script name> <process> <corefile>`
	
	process - Name of the process that has produced the core file
	
	corefile - Path to the core file
	
	Define the core script in system.dat as follows:
	
	`CoreScript <script name>`
	
	If a script is not defined, the core file remains and will be detected by SMService during the next cycle.

Server

This section of the system.dat file defines all machines that run services. Each server must be assigned a separate server ID number from 1 to 10. The format is:

```
service <hostname> <server id>
```

For example, for services running between machines london and paris:

```
server london 1

server paris 2
```

The value for hostname can be specified literally as <local>. If this is the case, then SMService will automatically substitute the name of the current node as the machine name. For example:

```
server <local> 1
```

While it is possible to configure services to run on different nodes and to have redundant versions of non-database services running on multiple nodes this is generally only done for very specific circumstances. In general it is suggested that you use the <local> syntax and run everything on one server.

Service

These entries in the system.dat file are definitions of services and process groups, such as interfaces, that are launched and monitored by SMService. Below is a sample service section:

# NAME	REQUIRED	START	DELAY	RESTARTS	RESET	MODE
service SMService	Y	60	0	10	86400	
service DBService	Y	90	0	10	86400	
service ODService	Y	180	0	10	86400	
service DDService	Y	180	0	10	86400	
service MTService	Y	180	0	10	86400	
service MBService	Y	180	0	10	86400	
service JMService	Y	280	0	10	86400	
service SwService	Y	280	0	10	86400	
service PFService	Y	4000	0	10	86400	
service corbagateway	Y	120	0	10	86400	
service service_alert	Y	120	0	10	86400	

The following table describes the SMService Service fields.

Field	Description
NAME	The name of the executable for the particular service.
REQUIRED	Indicates whether the instance of the service is required for the system to be functional. Valid values are 'Y', 'Yes', 'N', or 'No'. If there are no instances of a required service, the system locks until an instance is started.
START	The time taken for a service to start.
DELAY	Sets the number of seconds to wait before restarting a failed service. It only applies to processes that failed after they were running. Processes that fail before initialization are restarted based on the period parameter. A negative number indicates that the process is not restarted.
RESTARTS	The number of times to attempt restarting a process. A process is no longer automatically restarted after this value is exhausted until the process is reset (see below).
RESET	The timeout period that controls the rate at which processes are reset. When a process is reset, the restart counters re-initialize. A negative value deactivates this feature.
MODE	An optional argument that specifies the high availability mode of the service. If a mode is specified, the service starts with -<mode> and –number <n>, where <n> is the id defined for the node in the server line. Valid modes are exclusive, redundant, parallel or not specified. Exclusive runs with only one server. Redundant specifies running two servers, each with a database that mirrors the other. Parallel involves using Oracle Parallel Server to run two servers with a shared database.

Program

The program section of the system.dat file defines the executable program and command line options for each service. This section is optional, but can be used for the following:

- Specifying an alternative executable for a particular service. For example, setting TCDBService as an instance of DBService.
- Specifying command line options across all instances of a service. This simplifies the instance definition so that the command line options do not have to be duplicated for each definition.

Below is a sample applications section:

# NAME	EXE	ARGS
program DBService	DBService	-nodaemon
program ODService	ODService	-nodaemon –aggregates

# NAME	EXE	ARGS
program DDService	DDService	-nodaemon –zones –subscribezone –allowReset –alarms ALL
program MTService	MTService	-nodaemon
program MBService	MBService	-nodaemon
program JMService	JMService	-nodaemon –dbs
program SwService	SwService	-nodaemon
program PFService	PFService	-nodaemon
program corbagateway	Corbagateway	-nodaemon –ORBInitRef NameService=iioploc:// <hostname>:1750/NameService –ORBLogFile /users/<username>/dialog_log/orb.log –ORBDebugLevel 3 -implname InterSys_<hostname>_<username> -iorfile /users/<username>/etc/ <username>_vns.ior –publisher –xmldir /users/<username>/dist/wwwroot/xml
program service_alert	Mycentricity	-nodaemon –xmldir /users/<username>/dist/wwwroot/xml

The following table describes the SMService Program fields.

Field	Description
NAME	Specifies the name of the service that the executable belongs to. Valid services for this value are defined in the service section.
EXE	Specifies the name of the executable that runs the service.
ARGS	Defines the command line options that are used in all instances of the service.

Instance

The instance section of the system.dat file defines how the services are started. The format of each line is:

```
instance <node> <service> <database/args>
```

The following example starts nine services on the local node.

#	NODE	SERVICE	DATABASE/ARGS
instance	<local>	SMService	
instance	<local>	DBService	

#	NODE	SERVICE	DATABASE/ARGS
instance	<local>	ODService	
instance	<local>	DDService	
instance	<local>	MTService	
instance	<local>	JMService	
instance	<local>	SwService	
instance	<local>	corbagateway	
instance	<local>	service_alert	

The following table describes the SMService Instance fields.

Field	Description
NODE	Defines the node. Valid nodes for this value are defined in the server section. The value for NODE can be specified literally as <local>. If this is the case, then SMService will automatically substitute the name of the current node as the instance for which the service is to be started. By using "<local>" in place of a specific machine name, you can simplify your effort when replicating a system; you will not need to make changes to the system.dat at all.
SERVICE	The service being defined.
DATABASE/ARGS	Command line arguments that are applied when the service starts at this node. If the program section specifies command line options for a particular service, it applies to all nodes, so the arguments do not need specification here.

Model Build System Data File

The Model Build System Data File (system.dat.model_build) configures services for Model Build/ Configuration operations. It is formatted the same as system.dat.

The system.dat.model_build starts only SMService, DBService, ODService, and MBService. These services are generally executed from configuration scripts, such as ces_setup.ces, which require that some services be running to access the database and object classes.

Starting and Stopping Services

In order to start services, the following configuration files must be updated for the specific site configuration:

```
~/etc/system.dat.model_build

~/etc/system.dat

~/etc/system.dat.init
```

Starting Services

To start services, complete these steps:

1. Login to the server machine as the Oracle Utilities Network Management System Admin user.
2. Type:

   ```
   sms_start.ces
   ```

SMService starts. It reads and caches the system.dat file by default and starts the remaining services based on the data it just cached.

> **Note:** Using the -f <filename> option with sms_start.ces will override the default behavior and SMService will cache the specified file instead (e.g., ~/etc/system.dat.init, or ~/etc/system.dat.model_build.etc.).

Stopping Services

To stop services, type:

```
sms_stop.ces -s
```

When stopping services, you may have other tools running. The services are the core dependencies of all applications, so when services are stopped, all tools should be stopped and then restarted after the services have been re-launched. The best method to stop everything short of stopping ISIS is to stop the process by groups.

1. To stop both clients and services:

   ```
   sms_stop.ces -a
   ```

   > **Note:** Occasionally, there are tools or ISIS processes that may continue to exist as defunct and/or hung processes after the above commands do (or do not) run to completion. Check the process list on the UNIX machines for these processes and kill them prior to restarting. Otherwise, otherwise the system may not restart properly.

Chapter 8

Building the System Data Model

The Model Build process creates the operations data model that mirrors the utility company's Geographic Information System.

This chapter defines the configuration of the model builder and provides an overview of validating and testing tools. It includes the following topics:

- **Model Builder Overview**
- **Data Directory**
- **Model Configuration**
- **Model Build Process**
- **Model Manipulation Applications and Scripts**
- **Model Manipulation Applications and Scripts**
- **Schematics**
- **In Construction Pending / Device Decommissioning (ICP)**
- **Editing Symbology**

Model Builder Overview

The Model Builder Service (MBService) is used in building an Oracle Utilities Network Management System operations data model. The Oracle Utilities Network Management System operations data model is built using the customer's existing "as-built" data model (usually a Geographic Information System such as ArcGIS or graphic files such as AutoCAD). Necessary enhancements are applied to the GIS data model to make the "real-time" data model.

When changes are made in the GIS, MBService then merges them into the Oracle Utilities Network Management System data model. In addition to maintaining the model database, MBService also generates map files that are loaded for visual inspection.

A single spatial grouping of data known as a partition passes through various stages during its incorporation into the Oracle Utilities Network Management System Operations Model:

- GIS Data Extraction – to extract the data from the GIS to Oracle's vendor neutral model preprocessor (MP) file format.
- Preprocessing – to produce model build (.mb) files used by the Model Builder.
- Model Build (MBService) – saves the information into the Oracle Utilities Network Management System Operations Model RDBMS and writes out a set of maps.

The Model Builder service (MBService) is responsible for managing structural changes to the core operations model. Structural changes are largely the creation, deletion, and modification of objects. Non-structural changes involve updating attribute information such as status values.

The core operations model describes a set of interconnected network components with graphical representations and managed statuses. The objects contained within the model are subdivided into partitions with interconnections of partitions managed through the use of boundary nodes.

This data model must initially be obtained from an external source (such as a GIS) to populate the core operations model. Once populated, the core operations model is the basis for support of system services and the construction of diagrams.

The real-time services typically load parts of the model during initialization. These services also update attributes of the model. The process of model edit involves the creation, update, and deletion of objects that require consequential updates within services.

Patches

Import Files are submitted to MBService for processing. Each set of transactions submitted to MBService is considered a model patch and is applied to the current model. Most often, a patch is generated when a single partition is submitted to MBService for building.

The lifetime of a patch includes the following:

- Initial creation of the patch either locally or externally.
- Addition of the patch to the core operations model, where the patch will either be applied and become part of the current operations model or will be deleted if there is a problem with the patch resulting from patch format errors or real-time issues in the operations model (i.e., deleting a device with a call or outage).

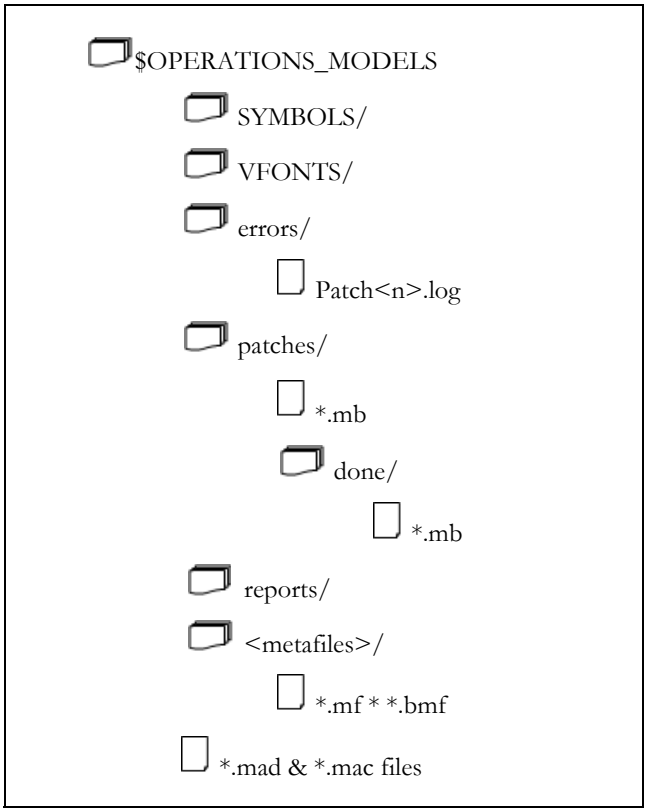
Data Directory

The data directory, which is owned by the UNIX Oracle Utilities Network Management System services user, must be unique for each Oracle Utilities Network Management System implementation. This directory is also referred to by the $OPERATIONS_MODELS environment variable. A unique version of this directory must be present for each Oracle Utilities Network Management System data model. It contains the model map files that the Viewer picks up and loads for the operator. These maps must be consistent with the data model; variations will cause problems with the Viewer's display of maps.

For example, if the same $OPERATIONS_MODELS directory is shared by the PRODUCTION and TEST environments (which have separate data models), then the map files that PRODUCTION accesses can get corrupted by model builds done in the TEST system. Model builds can also cause discontinuity between the data maps that PRODUCTION sees and the database that PRODUCTION uses.

Of course, with UNIX systems the model builds performed on TEST may fail due to an inability to write the resulting maps to the PRODUCTION data directory in the first place, if permissions are established correctly. This is another common mistake in environment construction. When you replicate environments, confirm that important variables like $OPERATIONS_MODELS are defined correctly.

Every UNIX Oracle Utilities Network Management System user must have a local copy of the $OPERATIONS_MODELS directory available for use by the Viewer. Once set up correctly, Oracle Utilities Network Management System maintains various remote copies of the map files by copying them from the database server as they are modified.



The following table describes the Model Builder directories and files.

Directory/ Files	Description
SYMBOLS	Contains the defined symbol sets for the presentation of all objects. (Convention only. May be moved elsewhere.)
VFONTS	Contains the set of supported vector fonts used in the presentation of scalable text. (Convention only. May be moved elsewhere.)
errors	Contains the output files of the model builder specifically related to errors and patch processing. The log files are named in Patch<patch_number>.log format.
*.mac	Textual representations of the background maps. The background map files corresponding to the *.mad files. These files are used by the Viewer to present background graphic information (boundaries, roads, text, etc.).
*.mad	Textual representations of the electrical maps. The map files used by the Viewer when presenting graphic information correlated to the network information stored in the database. It is essential to keep the database and the maps synchronized to ensure proper presentation and map conductivity.

Directory/ Files	Description
metafile dir (binary map files)	The name of the metafile directory varies with the architecture. The exact name of the metafile directory is determined by the string: $(uname).$(uname -m | sed "s/\//-/g") The metafile directory contains a binary copy of the maps that have been built. Whenever possible, the metafile is used instead of the *.mad files because metafiles are smaller and much faster. In a client/server environment, metafiles are distributed to the clients after a model build. (See below for more information)
patches	Contains <mapname>.mb files and/or <mapname>.mbd directories. These files define the model build transactions that will be submitted to the model. Files are moved into the done subdirectory after they have been submitted.
reports	This directory contains difference reports, which list all changes being introduced into the model for each patch. These are only generated if MBService is running with the –report option.
error	Contains the log files of the model build patch commitment process. The log files are named in Patch<patch number>. log format.

Binary Map Files

There is another directory that stores binary versions of all the *.mad and *.mac files. This directory is determined by the formula $OPERATIONS_MODELS/$(uname).$(uname -m \| grep sed -e "s/\//-/g"). Basically, the Model Build Service converts the *.mad and *.mac files into O/S specific binary files stored as *.mf and *.bmf files respectively. The Viewer uses this directory to first search for an appropriate metafile version of the map to load before trying to load the text version. The binary version improves the performance of the Viewer loading the map.

> **Note:** The caveat here is that the metafiles are O/S architecture specific. While the *.mad and *.mac files can be reproduced from a Sun system to a Linux system for example, the metafiles must be rebuilt against the Linux system architecture using the BUILD_METAFILES.ces script. Luckily this process is extremely fast.

For Application server client environments the Viewer only requires metafiles, so the *.mad and *.mac files only need to reside on the Services/Database server. The metafiles are all you need to distribute to client systems. Since the metafiles are significantly smaller in size than the *.mad and *.mac files, the distribution process has a lower impact to bandwidth.

Replicating an Oracle Utilities Network Management System

Essentially an entire Oracle Utilities Network Management System can be replicated across architectures without loss of data content. All you need to take into consideration are the metafiles, binary executables, and Isis executables, all of which are the only O/S specific characteristics of an Oracle Utilities Network Management System. This keeps you from relying upon specific machine architecture.

Model Configuration

This section provides a checklist of the steps to follow before a model build can be performed. There are numerous configuration scripts and SQL files that need to be configured in order to fully set up an operational Oracle Utilities Network Management System.

Conditions

The following conditions must be met for a successful model build.

Environment Variables

Each user of the Oracle Utilities Network Management System must have the environment variables. Environment variables are set in the nmsrc file located in the $NMS_HOME directory. You may edit the file using any text editor. The following environment variables are required to configure and build the model.

Environment Variable	Description
RDBMS_USER	Database login user name.
RDBMS_PASSWD	Database login password.
RDBMS_TYPE	Database type (ORACLE).
RDBMS_HOST	Database host machine. In the case of ORACLE_OCI, append ".world" to the machine name. Dependent upon the Oracle installation.
RDBMS_HOSTS_DIRECT	Usually the same as RDBMS_HOST. If multiple database instantiations are in use (as in software high availability), then the RDBMS_HOST values are concatenated together for RDBMS_HOSTS_DIRECT.
CMM_CELL	The name of the Isis communication "channel". All systems that have the same CMM cell value will communicate. Any value may be used, as long as all the interacting systems have the same value. Other non-interacting systems may not have this value.
CES_DATA_FILES	This environment variable is set to the directory where most configuration data files used by Oracle Utilities Network Management System software are installed. This includes *.dat, *.sym, *.cel files, among others.
NMS_ROOT	This environment variable is set to the directory where the top of the Oracle Utilities Network Management System installation occurs (i.e., ~/nms).

Environment Variable	Description
CES_HOME	This environment variable is set to the product installation directory (i.e., $NMS_HOME/product/1.9.0.0.
CES_LOG_DIR	This environment variable is set to the location of the service log files.
DATEMSK	This environment variable points to the path of the ces_datefmt file.
CES_SERVER	This environment variable contains the hostname of the Oracle Utilities Network Management System server.
CES_SMTP_SERVER	This environment variable points to an SMTP server where mail transactions can occur.
CES_SQL_FILES	This environment variable is set to the directory where most SQL files used by Oracle Utilities Network Management System software are installed.
CES_SITE	This environment variable contains a list of a set of configuration standards. Oracle defines the standard base configuration upon which customer configurations are built. The CES_SITE variable indicates which configurations to use. The setup process looks only for the files containing the values specified in this variable. The syntax is: `CES_SITE = "<project> product ces"` The first argument is the name of the customer or project. The last argument is the name of the default base configuration. There may be multiple configurations specified between the first and last arguments. When the system boots it processes the arguments from right to left, so it first loads the base configuration. Then it moves on to the previous argument and loads the associated configuration files if they exist. The process continues until each argument is processed.
OPERATIONS_MODELS	This variable specifies the directory into which the model will be built. That is, all maps and log files from the model build are located in this directory.
SYMBOLOGY_SET	This environment variable is set to the full path of the Oracle Utilities Network Management System symbol file <project>_SYMBOLS.sym.
CES_BASE_SYMBOLOGY	Directory that contains CES_SYMBOLS.sym, the default symbol file.

Environment Variable	Description
VFONT_DIR	The directory that contains a set of font definitions for vector text. By convention, it is set to $OPERATIONS_MODELS/VFONTS
CES_DATA_TABLESPACE	Contains the name of the primary Oracle tablespace. The installation and setup process uses it to better manage how database tables are set up.
CES_INDEX_TABLESPACE	Contains the name of the Oracle tablespace that is to be used for most indexes. The installation and setup process will attempt to put most indexes into this tablespace.
NMS_LOADPROFILE_DIR	This applies to power flow configurations where individual transformer profiles are used. It is the directory where the CSV files containing the profile data should be placed.

ISIS Configuration

Isis is the messaging backbone used by the Oracle Utilities Network Management System Operations Model, and it is required for every step of a model build. See **Isis Configuration** on page 4-1 for information about setup and configuration.

The CMM_CELL environment variable must be set uniformly over the network in order to communicate with programs on other machines.

To ensure Isis is running, type:

```
ps -ef | grep isis
```

Result: A pid (process id) is returned to confirm that Isis is running.

Verifying Database Connection

The RDBMS_USER, RDBMS_PASSWD, RDBMS_HOST, and RDBMS_TYPE environment variables should be set in the nmsrc file so the ISQL.ces script can be run by an administrative user to connect to an interactive session of the database.

ISQL.ces can make a connection to the database. To verify that a connection is possible to the database, complete these steps:

1. From the <project> user name on the master server, type:

   ```
   ISQL.ces
   ```

   A database prompt ensures that the environment is set up correctly.

2. Type quit to exit the database connection.

Directory

The model builder is primarily concerned with the tables within the selected database and the directory structure located under ${OPERATIONS_MODELS} as shown below.

Verifying Directory Set Up

A directory structure must be set up. To verify that it has been set up, type the following commands:

```
$ cd ${OPERATIONS MODELS}

$ ls
```

Result: A list of all directories will be displayed.

Setting up the Directory Structure

If the directory structure has not been set up, run the script ces_mb_setup.ces to configure it. It requires the OPERATIONS_MODELS environment variable to be set to the user's map data directory.

> **Note:** The ces_mb_setup.ces script is part of the model setup process, so the step listed here is redundant if this has already been completed.

The <project>_mb_setup.ces script creates and cleans the directory structure for customer specific model build setups.

The <project>_mb_preprocessor.ces script is called during the initial setup process to set up any additional directories or database tables that may be required by the model preprocessor. It is only required if special setup is needed.

Cleaning Up the Directory

If the data directory already exists from an obsolete data model, ces_mb_setup.ces -clean should be called to clean up all the residual files.

> **WARNING: If you run this script with the –clean option, you will delete the operational model.**

Class Organization

The operations model is designed around a class hierarchy. At the top of the hierarchy is the superclass, from which all other classes inherit attributes. The hierarchy may have multiple levels, each level having a parent/child relationship. The superclass is the only level that is always a parent and never a child.

Class Inheritance Definition

Classes and inheritance are defined and configured in the <project>_classes.dat and <project>_inheritance.dat files, respectively, located in the <project>/data directory. These files are loaded when the ces_setup.ces command is run to set up the data model.

These files can be individually loaded using the ODLoad command. The syntax to load classes.dat in Classes table via ODLoad is:

```
ODLoad -c <filename>
```

The inheritance relationships file, inheritance.dat, can be loaded into the INHERITANCE table via ODLoad. The syntax is:

```
ODLoad –I <filename>
```

In addition to these base class and inheritance files, special files may be included for dynamic condition classes (<project>_cond_classes.dat, <project>_cond_inheritance.dat) and classes

required for the power flow application (<project>_pf_classes.dat, <project>_pf_inheritance.dat). These additional files would be supplemental to the base files and should not duplicate any entries.

Oracle includes some required classes within the ces_core_classes.dat file. These classes are required in order for the Oracle Utilities Network Management System to work properly. Their inheritance is defined in ces_core_inheritance.dat and is also required. None of the information in these files should be changed, removed, or duplicated.

Attribute Table Configuration

The Oracle Utilities Network Management System attribute table is populated using <project>_attributes.sql. The user attribute table is populated using the <project>_schema_attributes.sql file.

Control Zone Configuration

If you plan to use Oracle Utilities Network Management System control authority functionality, then all electrical devices should have an assigned Network Component Group (NCG). This is usually assigned in the source data or computed in the preprocessor.

Symbology

Oracle Utilities Network Management System Viewer symbol information is stored in <project>_SYMBOLS.sym file. The <project>_ssm.sql file maps classes to the particular symbol.

Service Configuration File

The sms_start.ces.ces script is used to start up Oracle Utilities Network Management System services. It normally reads the system.dat file to determine which services to start up and what arguments to give them. Before a model is built, this configuration must not be used, because it contains startup commands for the Dynamic Data Service (DDService), the Managed Topology Service (MTService), and the Job Management Service (JMService), none of which will execute until a model has been at least partially built. The model build process expects to find another configuration file, system.dat.model_build, in the same directory that has a more limited set of services. In addition, there is a system.dat.init file that starts up only the database service.

Licensed Products File

The Automated Setup script (ces_setup.ces) and related .sql and .ces files will reference a <project>_licensed_products.dat file to properly configure the model to support the products you have licensed. This file is a text file and contains a list of the licensed Oracle Utilities Network Management System options. There is a template version of this file in $CES_HOME/templates/licensed_products.dat.template. The template should be copied to your $NMS_CONFIG/sql directory and renamed to a <project>_licensed_products.dat file. Then you should edit the file to uncomment the options you have licensed and are implementing. This edited template file should then be installed using the nms-install-config installation script prior to running the ces_setup.ces command.

Here is a list of options in the template file, with an indication of the license bundle they are in and the application(s) they affect:

```
#### Outage Management System - Standard Edition ####
## OMS-SE Applications ##
#opws         # Operator's Workspace (also used with DMS-SE)
#crewman      # Trouble Management
#troubleman   # Trouble Management
#webgateway   # Configuration Assistant, any "Web" products
```

```
  ## OMS-SE Adapters ##
  #crsi_gateway # Oracle Utilities CCB-NMS integration
  #oms_mwm      # Oracle Utilities NMS-MWM integration
  #ivr_gateway  # Generic Interactive Voice Recognition integration

  #### Outage Management System – Enterprise Edition ####
  ## OMS-EE Adapters ##
  #mq_gateway   # IBM MQSeries integration
  #mobile       # MQ Mobile integration
  #amr          # Generic AMR/AMI integration

  #### Distribution Management System - Standard Edition ####
  #WebSwitching # Web Switching Management (mutually exclusive with Switchman)
  #Switchman    # Original Switching Management (mutually exclusive with WebSwitching)

#### Distribution Management System - Enterprise Edition ####
#powerflow     # Power Flow, Volt/VAr Optimization, FLM, Suggested Switching
#flm           # Feeder Load Management
#network_analysis # Power Flow, Volt/VAr Optimization, FLM, Suggested Switching
#dynratings    # Dynamic Line Ratings

#### Additional NMS Applications – General ####
#datamart      # NMS BI (Oracle Utilities Business Intelligence integration)

## Additional NMS Applications – OMS focus ##
#crewcentricity# NMS Web Client (Web Workspace/Web Trouble) and Call Center (WCE, WCB)
#mycentricity  # NMS Paging (Service Alert)
#Stormman      # NMS Storm (Storm Management)

## Additional NMS Applications – DMS focus ##
#flisr         # Fault Location, Isolation, and Service Restoration
#fla           # Fault Location Analysis
```

Automated Setup

Oracle has an automated process that sets up the database schema and directory structure. Any scripts, SQL files, or data files that are properly set up, named and installed will automatically get picked up and used by this process. The automated setup process will use various SQL files mentioned in this section to build the initial data model.

ces_setup.ces: This script must be run on the model build host machine, the machine on which MBService is running. This process loads scripts, SQL, and data files that are properly configured and installed. The script makes liberal use of ISQL.ces, which submits all SQL files to DBService to be run. The syntax is:

```
ces_setup.ces [[-clean [-noVerify]] [-reset] | [-offline]] [-showme]
[-o <logFile>]  [-noInherit] [-debug]

[-noMigrations] [-cust]
```

The following table describes the ces_setup.ces command line options.

Option Variable	Description
-clean	Destroys the current model in order to build a new model. A prompt requires the user to verify this option. After this, a rebuilt model will still retain and use the same internal device identifiers (handles). This is useful for continuity of reporting before and after a clean model build.
-noVerify	Bypasses the interactive verification prompt that opens for the –clean option.

Option Variable	Description
-reset	Resets the generation of internal device identifiers (handles). If –reset if used with –clean, then a model built afterward will not be relatable to the previous model, even though they may look the same.
-offline	Preserves the data model, but erases the real-time and historic information concerning the model, such as tags, permits or notes. Configuration changes made directly to the database may be lost. For example, a list of login users maintained with the SqlX tool would be replaced with the login users defined in the CES_USER configuration table located in <project>_ceslogin.sql.
-showme	Prints the complete list in sequential order of scripts, SQL, and data files that are loaded or executed during the model build. Child scripts are indented in the list to easily identify parents. This option must be included in the database table or directory creation scripts in order to work properly.
-o <logFile>	Specifies a file into which the output of the process is written. Search this file for errors or warnings.
-cust	Updates the customers view after the setup is completed.
-noMigrations	Skips the automatic PR migration process. Use this option with caution, as it deviates from the supported process.
-noInherit	Skips base configuration and loads only the customer's configuration. This environment variable contains a list of a set of configuration standards. Oracle defines the standard base configuration upon which customer configurations are built. Use this option with caution, as it deviates from the supported process.

The CES_SITE variable indicates which configurations to use. The setup process looks only for the files containing the values specified in this variable. The syntax is:

```
CES_SITE="<project> product ces"
```

The first argument is the name of the customer or project. The last argument is the name of the default base configuration. There may be multiple configurations specified between the first and last arguments. When the system boots it processes the arguments from right to left, so it first loads the base configuration. Then it moves on to the previous argument and loads the associated configuration files if they exist. The process continues until each argument is processed.

The noInherit option makes sure that only the left-most configuration is loaded. Usually the left-most configuration is the customer's project-specific configuration based on Oracle's standard product configuration.

The setup process runs a large set of shell and SQL scripts that set up all aspects of the Oracle Utilities Network Management System model. The right-most value of the CES_SITE environment variable identifies a "base," or predefined configuration. By default, the setup process sets up the model in the predefined configuration. However, the setup script contains numerous "hooks" that when encountered, install project-specific configuration that overrides the base configuration.

For example, if project XYZ defines a base model stdbase, then the CES_SITE environment variable is set to "xyz stdbase." The stdbase configuration is used by default, with project-specific files overriding stdbase files when encountered. The stdbase configuration may contain a script stdbase_mb_preprocessor.ces that sets up the data model for the stdbase version of the

preprocessor. Project XYZ uses a different preprocessor with a different setup. The Oracle Utilities Network Management System setup process has a hook for a <project>_mb_preprocessor.ces file, so any file of this form with the project prefix as specified by CES_SITE (in this case, xyz_mb_preprocessor.ces) is called in place of the stdbase version. The exact details are dependent upon the nature of the "hook" involved. Some hooks are set up to call both the project script and the base script, while others will only call one or the other.

Linking In Customers

In order for Oracle Utilities Network Management System Trouble Management to run, user information must be linked into the model. This information is assumed to be in the database, whether explicitly loaded or whether linked in as a synonym. Oracle requires that the table that contains the end user information be joined to the SUPPLY_NODES table as a view called CES_CUSTOMERS.

Population of the CES_CUSTOMERS table

CES_CUSTOMERS is a materialized view populated with a join on four tables with details about customers, their meters, and their locations:

- CU_CUSTOMERS
- CU_SERVICE_LOCATIONS
- CU_METERS
- CU_SERVICE_POINTS

To update the Oracle Utilities Network Management System customer model, project-specific customer import processes will drop and rebuild mirror versions of these tables, named:

- CU_CUSTOMERS_CIS
- CU_SERVICE_LOCATIONS_CIS
- CU_METERS_CIS
- CU_SERVICE_POINTS_CIS

They will then run `product_update_customers.ces`, which will perform change detection between the CU_*_CIS tables and their Oracle Utilities Network Management System counterparts, perform incremental updates to them, and re-create the CES_CUSTOMERS table.

> **Note**: If you do not want to update the CU_* tables or you do not have an updated set of CU_*_CIS tables, you should add the "`-no_pre_process`" option to the call to `product_update_customers.ces` and the CU_* tables will remain unchanged.

From the CES_CUSTOMERS view, a smaller table (or view) must be extracted that is called CUSTOMER_SUM.

Population of the CUSTOMER_SUM table

The CUSTOMER_SUM table is a smaller extraction of the CES_CUSTOMERS information in which the customer information is summarized. JMService uses this for faster calculations. Depending on the definition of CUSTOMER_SUM (table or view), a fresh extraction may be required after each model build.

Model Build Process

Model Build with a Preprocessor

In most cases, customers will place source data files into a designated directory and run the ces_model_build.ces script. This script takes no arguments and builds whatever maps are recognized by the <project>_maps_to_build.ces script. When the build completes, any completed maps will have import files automatically placed in a designated directory. In some cases, such as projects using the JCMB version of the preprocessor, models may be built directly from import files.

Customer Model Build Scripts

The following table describes the model build scripts.

Script	Description
ces_model_build.ces	Builds the maps recognized by <project>_maps_to_build.ces. Upon completion, the ${OPERATIONS_MODELS}/patches/done directory contains import files for the built maps.
<project>_build_map.ces	Required for any model build process that has a model preprocessor. Takes a map name and generates an import file for that map. The resulting import file is placed in the $OPERATIONS_MODELS/patches directory.
<project>_build_maps.ces	Exactly the same as the <project>_build_map.ces script, but it is intended to take multiple maps. Any maps supplied will be built as a single model transaction. This is recommended when there is a model transaction involving multiple maps, especially if facilities are being transferred from one map to another.
<project>_maps_to_build.ces	Required for all model build processes. Identifies and prints a list (single line, space separated) of all maps that are queued up to be built.
<project>_postbuild.ces	Although not a required element of the model build, project-specific needs may call for an additional process after each model build. The additional process is carried out by the <project>_postbuild.ces script. It is run after the ces_model_build.ces script builds a complete set of maps. Common reasons for this process include recalculations of control zones, a fresh extraction of the CUSTOMER_SUM table, or a recache of DDService.
<project>_prebuild.ces	Although not a required element of the model build, project-specific needs may call for an additional process before each model build. The <project> _prebuild.ces script carries out the additional process. It is run before the ces_model_build.ces script builds a complete set of maps. This process is rarely needed.

Model Build with a Post-Processor

If a post-processor is needed for the model build, you should create and install the <project>_postbuild.ces script. If the post-processor requires patches to be applied to the model, it will build import files and put them in the patches directory. The ces_build_map.ces script can be called with the -noVerify option to build each patch without user interaction.

Constructing the Model

To ensure correct model construction, complete these steps:

1. To ensure Isis is running, type:

   ```
   ps -ef | grep isis
   ```

   Result: A pid (process ID) should be returned confirming that Isis is running.

2. If a model build preprocessor is being used, make sure that the expected scripts are created and installed. These are <project>_build_map.ces and <project>_maps_to_build.ces.

3. When new files are brought to the system, place them in the appropriate directory on the master server before initiating the model build.

   - Import files should go into the ${OPERATIONS_MODELS}/patches directory.
   - Preprocessor input files will probably go into a project-specific directory. An example of a commonly used directory is ${OPERATIONS_MODELS}/mp.

4. Log into the master server as the administrative user and initiate a model build by typing:

   ```
   ces_model_build.ces
   ```

   Or, if you want to produce a build log, enter the following:

   ```
   ces_model_build.ces | tee model_build.log.$(date+"%d.%m.%Y")
      2>&1
   ```

   Result: Each import file will be processed, updating the Operating Model and Graphic Presentation files.

5. Wait for the user prompt before continuing further model build operations. This process may take some time.

6. Review the error output information contained in the errors directory.

Model Manipulation Applications and Scripts

After a customer has built a model, there may be times when certain scripts or applications may need to be run to clean up errors that have been introduced into the model or to remove obsolete devices or maps. There are several scripts and applications that exist to do this model manipulation. These scripts and applications are described below.

DBCleanup

Most customers should run the DBCleanup application periodically. It examines the modeling database tables and looks for duplicate active rows, orphaned objects, and inconsistencies in the ALTERNATE_VIEWS table. If any of these problems are discovered, the application will attempt to fix the data so that it is consistent with the rest of the database tables.

ces_delete_map.ces

The ces_delete_map.ces script allows the user to remove an obsolete map from the model. It creates a patch that is processed by MBService that will deactivate the map itself and all devices contained in it. This script should be used sparingly.

ces_delete_object.ces

The ces_delete_object.ces script allows the user to deactivate all instances of a single, specified device in all the maps in which it appears. It creates a patch that is processed by MBService to remove all the instances of the device.

ces_delete_branch_obj.ces

The ces_delete_branch_obj.ces script also allows the user to deactivate all instances of a single, specified device from all the maps in which it appears. However, this script directly modifies the modeling database tables, potentially leaving the services in a state that is inconsistent with the current information in the database. After this script is used, either all services should be re-started or MBService should be re-started and then all the maps involved with the deleted object should be re-built. After MBService re-builds the maps, it will send out notifications to the other services to bring them all into sync.

ces_delete_patch.ces

The ces_delete_patch.ces script allows the user to delete a single patch or a range of patches that exist in the database. The script directly modifies the modeling database tables, potentially leaving the services in a state that is inconsistent with the current information in the database. After this script is used, either all services should be re-started or MBService should be re-started and then all the maps involved with the deleted patches should be re-built. After MBService re-builds the maps, it will send out notifications to the other services to bring them all into sync.

mb_purge.ces

The mb_purge.ces script can be used to reduce the size of the modeling tables in the database. It will remove old, inactive rows as specified by the user.

TopVal

The TopVal program checks for common modeling errors and deficiencies, such as deenergized areas, loops, other conditions including tied feeders and larger feeder groups.

TopVal should be used on a regular basis during both initial model construction and routine model maintenance. When used properly and consistently, TopVal can help identify model anomalies before they become production issues.

The TopVal program takes the following command-line options:

Command Line Option	Description
-current	Perform the tests against the current model. The default is the nominal model.
The following four options are mutually exclusive; only one can be used at any time:	
-all	Show all de-energized branches.
-cond	Show de-energized conductors.
-xfmr	Show de-energized transformers.
-snd	Show de-energized supply_nodes. This is the recommended setting.
-loop	Show all loop/parallel objects.
-node	Show FBD/FID/MID node mismatches, including mis-oriented feeders, feeders with shared identifiers, and orphaned identifiers.
-mid <n>	Show MID groups of <n> or more meshed feeders. The default is 3.
-fid <n>	Show FID parallels of <n> or more paralleled feeders. The default is 2.
-ato <n>	Show <n> or more ATOs below an FID. The default is 1 (all ATOs).
-src <n> [-verbose]	List source and feeder groups for islands including <n> objects or more. The default is 1 (all islands). Use the -verbose option to show an additional island breakdown by device object class.
-list <s>	Show classes that inherit from class name <s>.
-stop <space-delimited stop class numbers> [-min <n>]	Lists all the objects of given classes downstream from each feeder identifier. The default is just 994 (supply_node). Use the -min <n> option to show only <n> devices. The default is 1.
-noquick	Show handle aliases for all objects. This setting is not recommended, because it slows down processing.
-debug	Show debug information.

Sample Usage

Below is a sample TopVal command line that is useful in identifying model problems. Note that the TopVal command is followed by a top_view.ces command line, which is used to provide navigation for the TopVal output (see the following section for more info on top_view.ces):

```
TopVal -snd -fid -mid -loop -node >~/tmp/top_val.dat
top_view.ces -f ~/tmp/top_val.dat
```

top_view.ces

The top_view.ces program is a PerlTk script designed to help navigate the output of the TopVal program described in the previous section. TopVal can generate a substantial amount of output, and Top_view.ces can help you navigate from the model objects identified in the TopVal report to the actual model element. Top_view.ces provides navigation through a PerlTk script user interface to the Oracle Utilities Network Management System X11 Viewer tool. Using this interface can help you get a better idea of the reported model conditions.

The top_view.ces script looks for specific character sequences in the TopVal output to understand where one section ends and another begins. These sections are then rendered in a tree table within the top_view.ces PerlTk panel. From this panel, you can:

- Double-click the header for a section to expand or collapse the section. (It works like a toggle.) Inside each section, output is grouped by model partition (*.mad), where appropriate. Each impacted model partition shows the number of reported model elements for that partition.
- Double-click the model partition line to render the actual objects identified for that section of the report.
- Click a line in the top_view.ces PerlTk user interface to parse the line. The following occurs:
    - The program looks for an Oracle Utilities Network Management System handle, which is a series of digits followed by a period, followed by more digits (<d+>.<d+>).
    - If no handle is found, the program looks for a partition name, which is any string followed by ".mad" (as in *sub_1.mad*). If a partition name is found, the program highlights the partition.
    - The program displays the handle or the partition in the Selected Object panel on the upper-right side of the top_view.ces PerlTk user interface, and the Focus button is enabled.
- Click the **Focus** button to find a running Oracle Utilities Network Management System X11/Motif Viewer with the ISIS name VIEW;0;1. (This is the standard name for the first viewer started in a standard X11/Motif user environment.) If no such viewer can be found, the program starts one, if possible. The viewer then focuses on the selected handle or partition.

The top_view.ces script has the following command line arguments:

Command Line Option	Description
-file	Specifies the file to process. The default is: ~/tmp/top_val.dat
-nofocus	Prohibits the program from giving focus to the X11 viewer.
-mingrp	Specifies the minimum number of items necessary to create a group. The default is 1.
-debug	Enables debug output, which is sent to stdout.
-help	Displays a short usage statement

AuditLog

The AuditLog application works with the scripts and applications defined above to keep a persistent record in the database of the data manipulation activities that have been going on when a customer uses any of these scripts or applications. The information is stored in the MODEL_AUDIT_LOG database table and can be useful when trying to help support a customer with corrupted data by helping to provide a better scenario of the activities that might reproduce the problem.

Schematics

Oracle Utilities Network Management System– Schematics can automatically generate orthogonal schematic overviews of the nominal network.

Model Requirements for Schematics

In order to use Oracle Utilities Network Management System Schematics, the following is required of the data model:

- All substations must have the same partition class.
- The substation partition class must only be used by substations.
- All boundaries between feeders and substations are designated with a distinct class of devices.

Schematic Limitations

Since Oracle Utilities Network Management System Schematics uses a splayed-tree representation of the nominal network, it is necessarily geared towards radial networks and will have difficulty representing nominally looped, parallel or meshed areas. Oracle Utilities Network Management System Schematics is also geared towards simple network objects (i.e. a switch) and cannot keep related devices in close proximity (i.e., the internals of a switching cabinet).

Configuring Schematics

- All schematic configuration is controlled via command-line options which are passed to the schematic-generator, schematica. The script that contains the configuration is normally called <project_name>_create_schematics.ces
- The script must perform these three actions:
- Remove any previous schematic import files.
- Call schematica with all of the configured options.
- Process all generated import files.

The following table describes all of the command line options.

Command Line Option	Description
-mapPrefix <prefix>	Prepend all generated schematic maps with this prefix. **Required**.
-ptncls <#>	Partition class to use for all generated schematics. **Required**.
-validFeederStartClass <list of classes>	List of classes that designate the start of a feeder. **Required**.
-addStop <list of classes>	Include these classes as well as those specified via –stop
-balanceSubstations	Shift feeders around a substation until there are similar NUMBERS on each valid side.
-boundingBoxCls <class name>	Create a box of this class to indicate the substation-overviews extents. If unset, the box is not drawn.
-boundingBoxLabelCls <class name>	Create a label of this class, with the substation's name. Default is branch.
-branchWidth <dist>	Distance between two network branches that share a common upstream port. (see Figure 1 below)
-camelHumpHeight	Relative height (in terms of tier-height) of conductor-crossover bumps. Value between 1 and 0. (See Figure 4 below)
-camelHumpWidth	Relative width (in terms of branchWidth) of conductor-crossover bumps. Value between 1 and 0.
-classesToLabel <list of classes>	List of classes for which the schematic-generator should create and place annotation.
-connectionClass <class name>	Device class to use when creating a branch to span two or more non-conductor devices. (Must be a non-electrical branch)
-coordSystem <#>	The coordinate system the schematic generator will assign all schematics. Should not be an existing value. Defaults to the current maximum coord_system + 1

Command Line Option	Description
-db <DBService prefix>	Force the schematic-generator to use the DBService that has the specified prefix (i.e. –db MB will use MBDBService)
-dch	(Disable Camel-humps) Don't create camel-humps where conductors cross
-defaultConductorSymbology <valid symbol class>	Use this symbol class when attempting to write out any conductor that has a symbol class of 0.
-defaultFeederDirection <north|south|east|west>	If the schematic-generator is unable to determine the direction for a feeder, it will use this value. No default. If this option is NOT specified, the schematic-generator will ignore any feeder for which it can not determine a direction.
-deviceGaps <class name> <scale factor>	Scales all diagrams of the specified classes by the specified amounts
-deviceScaling <class name> <scale> <offset>	Scale all diagrams of the specified classes as well as shift them along their parent feeder's axis. Default scaling is 1.0, default offset is 0.0
-deviceHeight <#>	Size of all non-conductor electrical branches. (See figures following this table.)
-fastCrossovers	Use a faster, but less accurate algorithm to determine where conductors intersect.
-fbdBounded	Use this option if all feeder-heads have FID on one port and an FBD on the other.
-feederDirection <north|south|east|west>	Have ALL feeders extend in the specified direction.
-feederNameTable <table name> <column name>	The specified table for each feeder's FID, annotated with the value found in the specified column. For single-circuit schematics, the feeder name is used as part of the generated map's name.
-feederHeight <#>	Minimum distance between a substation and the first device of a feeder. (See figures following this table.)
-feederTextScale <#>	Amount to scale all feeder-name annotation.
-feederOffset <#>	Distance between adjacent feeders. Default is branchWidth*10. (See figures following this table.)
-feederPrefix <comma-delimited list of strings>	Only process substations whose map names contain the specified strings

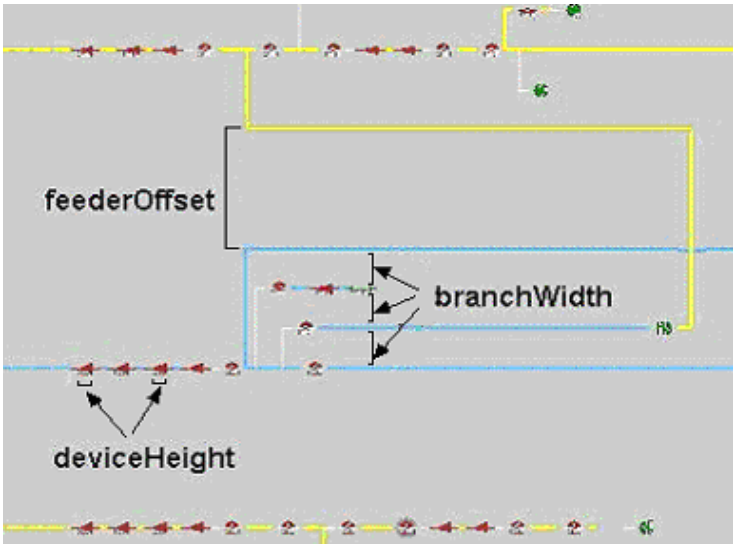
Command Line Option	Description
-geographicSubstations | -gs <table name> <column name>	Use this option when all substations are modeled in the geographic world. Schematica will search the specified table for all classes listed in –substationNodeClasses and set the substation name based on the value in the specified column. (This option must be used in conjunction with –substationNodeClasses)
-globalScaleFactor <#>	Increases the size of all objects and all overviews by this amount.
-invisibleClasses <list of classes>	List of classes (that never have symbology – i.e. zero-impedence conductors) that the schematic-generator should ignore when attempting to connect a feeder to its parent substation.
-intersubOffset <#>	Minimum distance between parallel sub-to-sub conductors. Defaults to tier-height or device-height*2, whichever is greater. (See figures following this table.)
-labelClasses <class name>	Use this text class when creating device annotation if the class <device_class_name>_t2 does not exist. Text class to use for all generated annotation. (See figures following this table.)
-noFeederToFeeder	Do not connect up feeder-to-feeder tie-points. (See figures following this table.)
-noIntraFeederConnections	Do not connect up bypass tie-points
-noPrune	Keep all devices in a feeder, not just those attached to open-points.
-noSubstations	Do not draw substations. Instead draw all feeders in the same map in one or more rows. (See –maxRowWidth)
-noSubToSub	Do not connect up sub-to-sub tie-points.
-orientation <ANY|HORIZONTAL|VERTICAL|ROUND_ROBIN|NONE>	Align all feeders according to the value. (ANY = normal feeder directions, HORIZONTAL = move all north/south-ward feeders to east/west, VERTICAL = move all east/west-ward feeders to north/south sides, ROUND_ROBIN = evenly distribute the feeders around the substation, NONE = move ALL feeders to side specified by "-feederDirection") Default is ANY.
-overviewName <string>	The names of all resulting schematic maps will take the form <map prefix>_<overview name>_<substation name>
-placeSubsByConnection	Attempt to position substations with the greatest number of common connections closest to each other.

Command Line Option	Description
-priorityClasses <list of classes>	Keep the specified list of classes as close to the main trunk of the generated schematic tree as possible.
-reorientDeviceClasses <list of classes>	Ensure that diagrams for the specified classes are always oriented from left to right. (Use this if symbols appear upside-down.)
-scaleFactor <#>	Multiplies the size of all conditions by this amount.
-skipEmptyFeeders	Do not draw feeders that contain an exceedingly small number of devices (< 10 devices)
-sort <GEO|SPAN>	Arrange feeders either geographically (using only the anchor points of each feeder) or arrange them to minimize the distance feeder-to-feeder tiepoint connections must span. Values: GEO|SPAN GEO = geographic ordering, SPAN = minimal spanning tie points. Default is GEO
-startAtFID	Use when all feeder heads are modeled to have an FID attached.
-stop <list of classes>	List of all device classes the schematic-generator should not trace past.
-subSpacing <#>	Minimum distance between substations. No default. (See figures following this table.)
-substationBoxSize <#>	Create a square of the specified size and scale the original substation schematic to place inside. Default is 1000. (See figures following this table.)
-substationBoxCls <class number>	Create a box of this class-type around the substation. No default. If not specified, there will be no visible box around the substation.(See figures following this table.)
-substationName <database table name> <table column name>	Do not model substations. Instead, search the specified table for each feeder's FID and group them into substations based on the values in the specified column
-substationNodeClasses | -snc <list of class names>	When used in conjunction with –geographicSubstations, it specifies what type of nodes to initiate substation tracing from. Generally, the value should be SRC.
-substationPtnCls <#>	Only process substations with this specific partition-class. No default.
-substationTextScale <value>	Amount to scale the size of the substation label.
-substationTransitionClass <list of classes>	The set of classes that designate the transition between feeder and substation. Defaults to hyper_node.

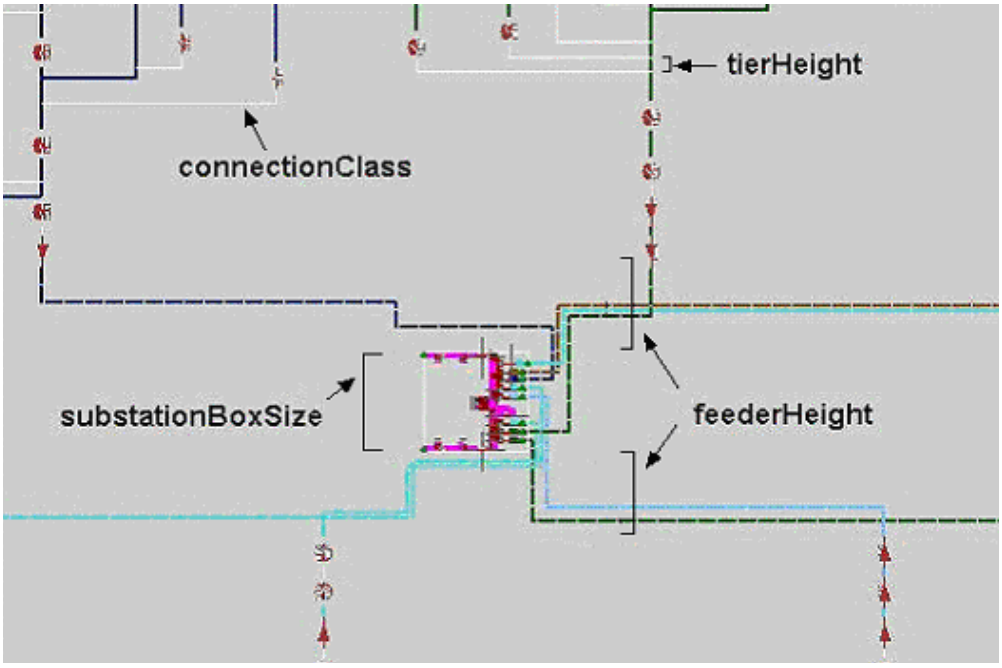
Command Line Option	Description
-tapDeviceOffset <value>	Distance to offset single devices from the main trunk.
-textOffset <#>	Distance (along the feeder's main axis) to pull all device annotation. Default is 0. (See figures following this table.)
-textScale <#>	Scale all device annotation by this amount.
-tierHeight <#>	The distance (along the feeder's axis) a conductor will span. Default is 50. (See figures following this table.)
-tilebasedmaps | -tbm	Calculate the geographic orientation of each feeder based on the coordinates of the all open points, not on the base map's extents.
-voltage <minimum voltage> <max voltage>	Only process devices that fall into the specified voltage range.
-weightClass [<class name> <weight>…>	Tells the schematic-generator to process certain classes of objects sooner when creating its internal schematic tree. If weight < 0, process later. If weight > 0, process sooner.

Note: <list of classes> format: [-]class name[+],[-]class name[+],....] [-] exclude this class (and all descendents if '+' is used) [+] include all descendents.

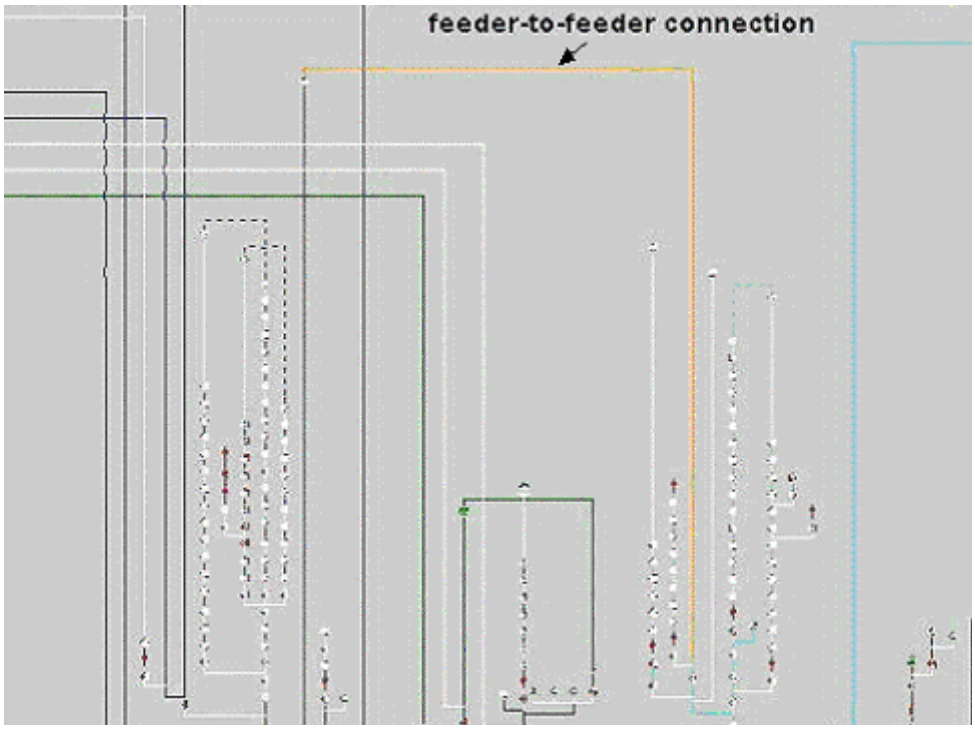
The following figure shows the deviceHeight, branchWidth, and feederOffset.



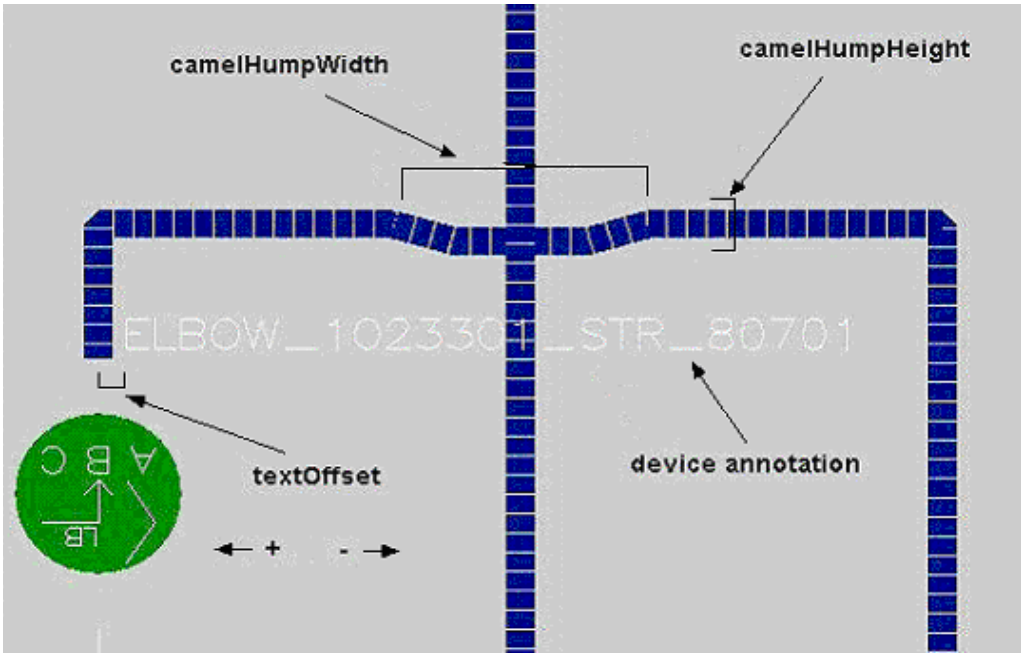
The following figure shows substationBoxSize, feederHeight, tierHeight, and connectionClass.



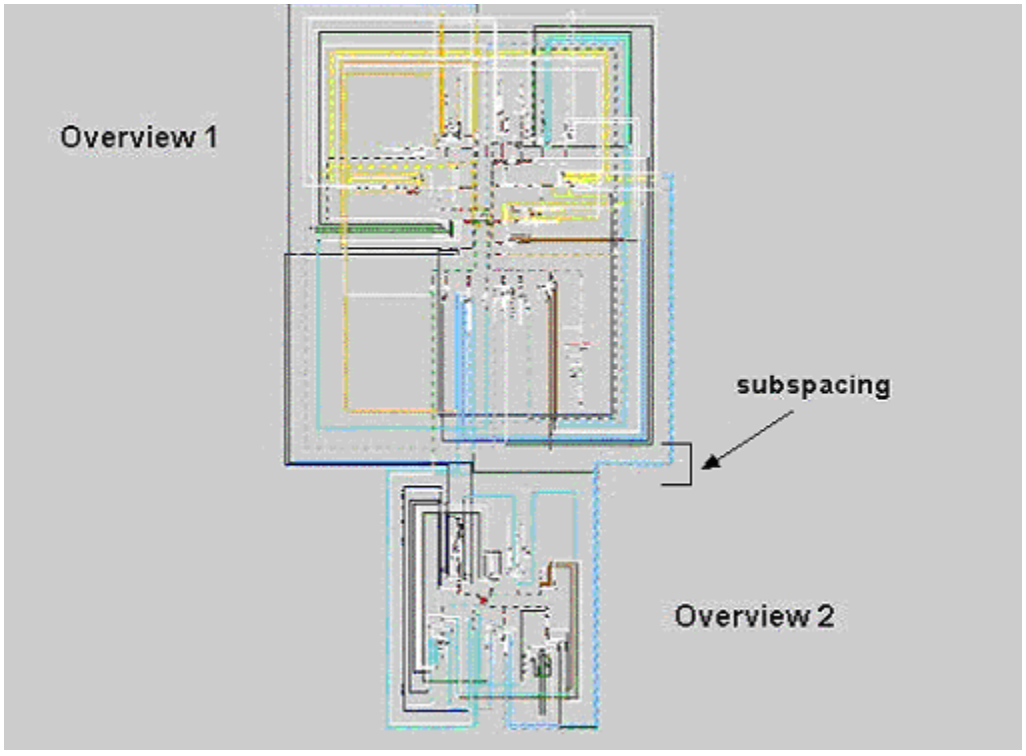
The following figure shows a feeder-to-feeder connection.



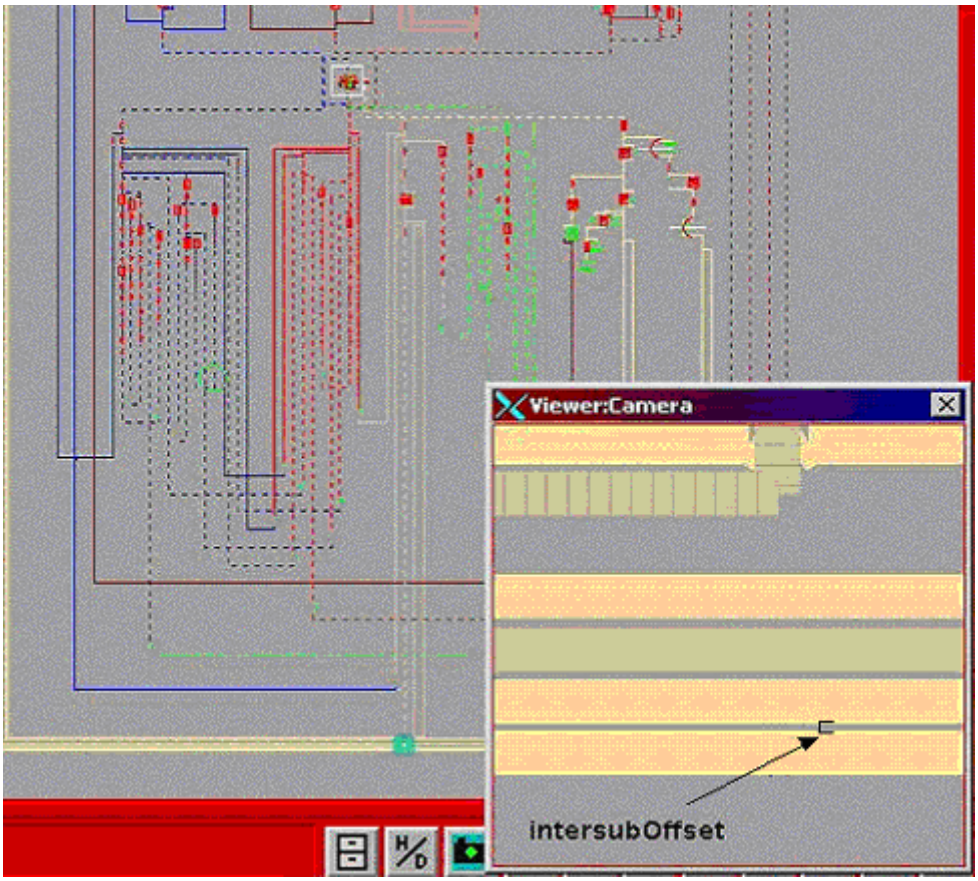
The following figure shows camelHumpWidth, camelHumpHeight, device annotation, and textOffset.



The following figure shows subspacing.



The following figure shows intersubOffset.



Generating Schematics

To create schematics, the customer-specific script <project>_post_build.ces must have a call to <project>_create_schematics.ces.

The Post-Build Process

After the build process has processed the final map, it calls <project>_post_build.ces. If there is an entry for create-schematics, it calls it at this time.

Creating the Import Files

Once invoked, the schematic generator loads in the entire nominal network model and attempts to group all feeders with their parent substations. After it finishes determining the layout and spacing for all feeders and substations, it writes out one import file for each substation.

Processing the Import Files

After the schematic-generator creates the import files, the schematic script compares the most recent previous version of each file. If no changes are detected, it skips the map. Otherwise it proceeds to build the import file as per the normal model-build process.

In Construction Pending / Device Decommissioning (ICP)

Oracle Utilities Network Management System supports the modeling and visualization of devices that are in-construction as well as devices that are marked for decommissioning. ICP can be used for commissioning new construction (such as road widening) and should not be used for nominal-state changes (such as feeder load balancing).

Device Lifecycles

In a GIS system, a device will fall in to one of four possible states:

Device State in GIS	Description
Install	Objects that are proposed construction or new objects to be commissioned at a future date
Existing	All objects that are in the GIS as-built and commissioned
Remove	Objects that are commissioned today and are part of the active model however there is a construction plan to remove these objects
Retired	All objects that have been completely de-commissioned. These devices will not exist in the real-time system.

Model Requirements for ICP

In order to use In Construction Pending (ICP), each affected device must have an additional value listed in their physical_properties entry inside the import file, as shown below:

Device State in GIS	Required Physical_Properties Value in Import files
Install	Construction
Existing	NA
Remove	Decommission
Retired	NA

The model preprocessor calculates these values and writes them out into import files.

> **Note:** Model Extractors must be modified to not filter out devices in the "Install" state.

Model Builds and Commissioned/Decommissioned Devices

The Commissioning Tool moves devices between "Not Commissioned" and "Commissioned" as well as "Not Decommissioned" and "Decommissioned".

If an operator commissions a device, marked as Construction, a model build will not reset the commissioning state (i.e., Subsequent model builds will not undo changes made by the Commissioning Tool).

Effect of ICP Devices on Network Topology

Devices affect the network's topology as follows:

Device State	Commissioned / Decommissioned	Does Device affect Network Model
Install	Not commissioned	No.
Install	Commissioned	Yes. As normal existing device.
Remove	Not decommissioned	Yes. As normal existing device.
Remove	Decommissioned	No.

ICP Device Symbology

The Viewer will hide certain ICP-marked devices and display certain ICP devices with additional symbology.

Device State	Commissioned / Decommissioned	Default Visibility	Special symbology
Install	Not commissioned	Hidden	Yes.
Install	Commissioned	Visible	Yes
Existing	NA	Visible	No
Remove	Not decommissioned	Visible	Yes
Remove	Decommissioned	Hidden	Yes

Note: See the User Guide section on "Using the Operator's Workspace Viewer" for more information on ICP symbology and how to use the Commissioning Tool.

Auto Throw-Over Switch Configuration (ATO)

Oracle Utilities Network Management System supports the modeling and visualization of Auto Throw-Over (ATO) devices. Critical customers such as hospitals, manufacturing, financial and emergency services, require higher level of power quality and reliability. These customers are normally provided with a primary and backup source of power to improve the reliability. Utilities deploy automatic throw over devices to switch the load to backup source when the primary source is not available. Often these devices have automatic restoration feature where the load is fed by the primary source when primary source is energized after an outage.

Model Requirements for ATOs

In order configure ATOs in the Oracle Utilities Network Management System, the Model Build process needs to know what two devices are controlled by the ATO controller. One device must be identified as the primary or preferred feed, which would be normally closed, and the other device would be the secondary or alternate feed, which would be normally open. These relationships and control behaviors are modeled in the ATO_CONTROLLERS table, as shown below:

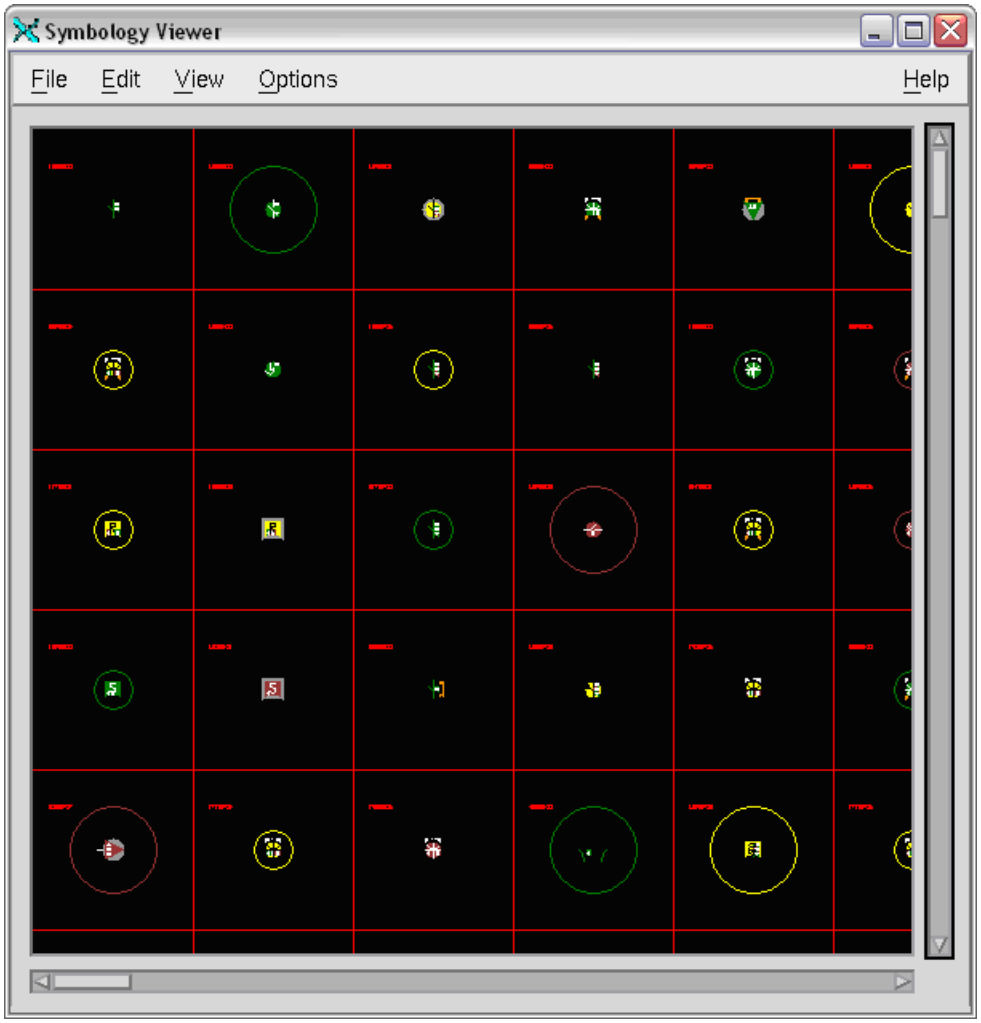
Field	Format	Comments
H_CLS	N	Class part of the ATO controller handle. Required.
H_IDX	N	Index part of the ATO controller handle. Required.
PARTITION	N	ATO controller partition.
CONTROL_FUNCTION	V32	ATO control function identifier. Required. Values: • 2dev – 2 ATO Devices and no auto-restore • 2dev_arc – 2 ATO Devices, auto-restore, no momentary on restore operation • 2dev_momentary_acr – 2 ATO Devices, auto-restore, and will create a momentary on restore operation
ATO1_CLS	N	Class part of the handle of the primary ATO device. Required.
ATO1_IDX	N	Index part of the handle of the primary ATO device. Required.
ATO2_CLS	N	Class part of the handle of the secondary ATO device. Optional.
ATO2_IDX	N	Index part of the handle of the secondary ATO device. Optional.
PARAM1	N	Delay (in seconds) until primary ATO device is opened during throwover - Optional.
PARAM2	N	Delay (in seconds) until secondary ATO device is opened during auto-return (ignored by control function "2dev" but column presence is still required) - Optional.
PARAM3	N	Delay (in seconds) between operating primary and secondary ATO devices. If not configured, there is no delay. -Optional.
BIRTH	D	Birth date of when the object is activated into the model
BIRTH_PATCH	N	Patch which activated this object
DEATH	D	Death date of when the object is de-activated from the model
DEATH_PATCH	N	Patch which de-activated this object
ACTIVE	V1	Active flag

Editing Symbology

The Symbology Editor lets you view and edit the Viewer symbols that represent aspects (devices, crew assignments, etc.) of the operations model.

Symbology Viewer

The Symbology Viewer is the first window you will encounter in the Symbology Editor tool. It displays all the symbols within the symbology file. Open the Symbology Viewer window by typing "symbologyEdit" or "symbologyEdit&" at the prompt. (The ampersand at the end of the command will allow the tool to run in the background, freeing up the shell command line for other commands.)



The following table describes the Symbology Viewer File menu options.

Option	Description
Select Symbol…	Lets you select a symbol by number. Once selected, the symbol is highlighted with a red border in the Symbology Viewer. (The symbol can also be selected directly from the Symbology Viewer by clicking on the symbol in the Symbology Viewer drawing area.)
New	Opens the Symbology Editor window with a blank drawing area and a new symbol number.
Print…	Opens the print dialog box.
Exit	Closes the Symbology Viewer window.

The following table describes the Symbology Viewer Edit menu options.

Option	Description
Symbology Editor	Select this option once you have a symbol selected and the Symbology Editor will open with the symbol pre-loaded. The Symbology Editor lets you alter the symbol or create new symbols. See **Editing Symbology** on page 8-30 for information on editing symbols. You can also press Ctrl+S on the keyboard to open the Symbology Editor window.

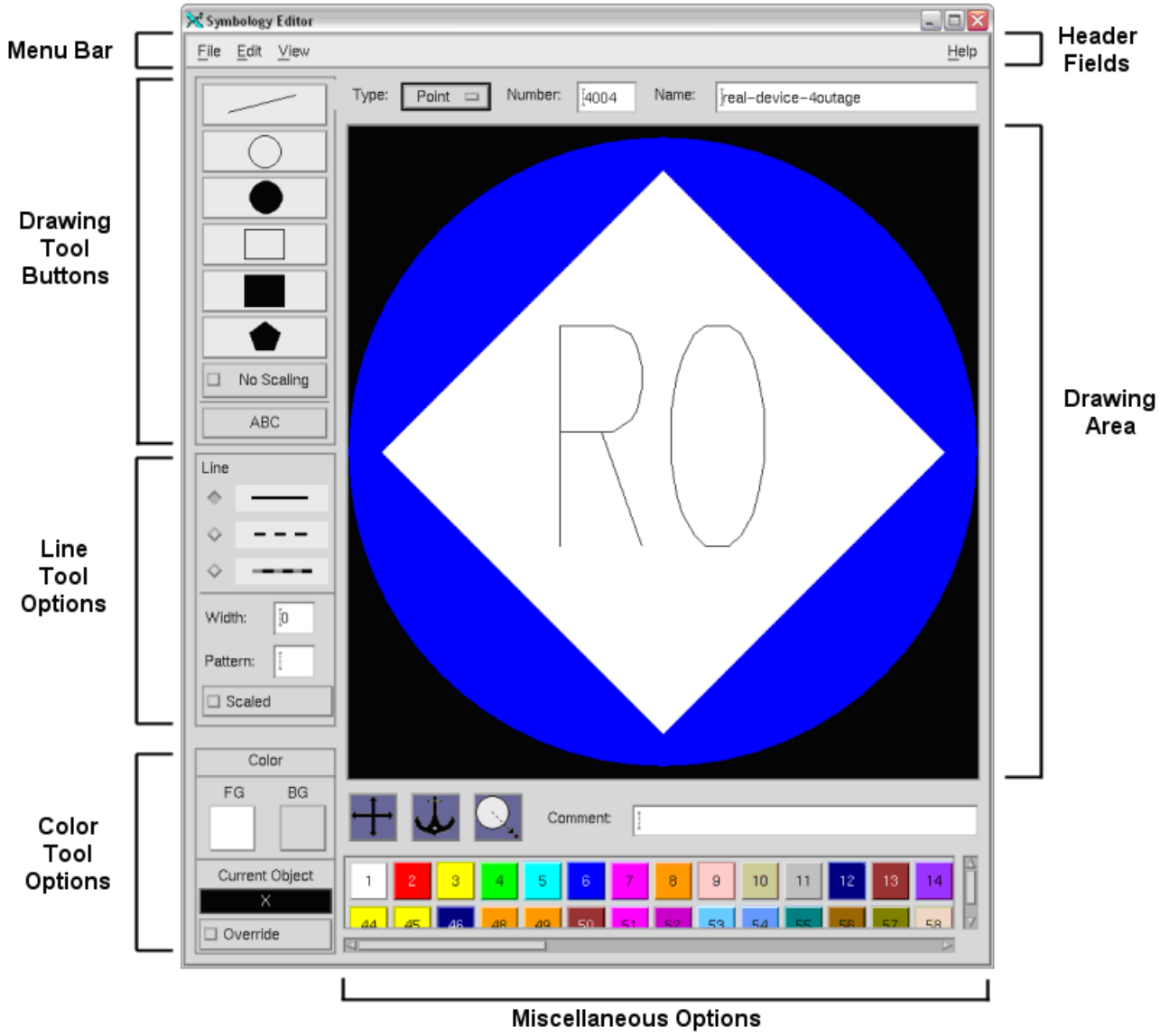
The following table describes the Symbology Viewer View menu options.

Option	Key Stroke	Num Lock	Description
Zoom In	Up Arrow	Off	Zooms in closer to the current view.
Zoom Out	Down Arrow	Off	Zooms out of the current view.
Pan Up	Keypad 8	On	Repositions the drawing area up.
Pan Down	Keypad 2	On	Repositions the drawing area down.
Pan Left	Keypad 4	On	Repositions the drawing area left.
Pan Right	Keypad 6	On	Repositions the drawing area right.
Pan NW	Keypad 7	On	Repositions the drawing area up and to the left.
Pan NE	Keypad 9	On	Repositions the drawing area up and to the right.
Pan SW	Keypad 1	On	Repositions the drawing area down and to the left.
Pan SE	Keypad 3	On	Repositions the drawing area down and to the right.
Maximum View	Keypad 0	Off	Zooms out fully to display all of the symbols.
Rotate Left	Left Arrow	Off	Displays the selected symbol rotated left.
Rotate Right	Right Arrow	Off	Displays the selected symbol rotated right.

Symbology Editor

The Symbology Editor enables you to create, alter, and save edited symbols. It provides options to manipulate lines, patterned lines, circles, filled circles, rectangles, filled rectangles, and polygons. You can also set text within the symbol and define color selections. Open the Symbology Editor from the Symbology Viewer by:

- selecting Symbology Editor from the Edit menu, or
- selecting New from the File menu.



Symbology Editor Window

The **Type** option menu allows you to select the type of drawing. Options are ***Point***, ***Line***, ***Polyline***, and ***Area***.

The **Number** field contains the number assigned to the symbol. You can use this field to assign a different number to the displayed symbol.

The **Name** field contains the name assigned to the symbol. You can use this field to assign a different name to the displayed symbol.

The Drawing Area is used to display and draw the symbol currently being created or edited. You can draw and add features by selecting a drawing tool shape and placing it in this area.

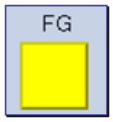
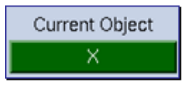
On the upper left-hand side of the Symbology Editor window are several drawing tool buttons, as shown below:

Button	Description
 Line	Draws a line within the area between two mouse clicks.
 Open circle	Draws an open circle with the diameter indicated between two mouse clicks.
 Filled circle	Draws a filled circle with the diameter indicated between two mouse clicks.
 Open square	Draws an open square with two diagonal corners indicated between two mouse clicks.
 Filled square	Draws a filled square with two diagonal corners indicated between two mouse clicks.
 Filled polygon	Draws a filled polygon with each corner indicated by mouse clicks. Clicking near the starting point indicates the final corner.
 Scaling toggle button	Toggles whether the elements drawn will be scaled as the symbol is zoomed in and out.
 ABC (text)	Activates the Enter Text dialog box. If you have text entered in the Text field, each mouse click in the drawing area will place text at that location of the symbol.

On the middle left-hand side of the Symbology Editor window are several line tool options:

Option	Description
 Solid	Causes lines (or shape outlines) to appear as solid lines.
 Dashed	Causes lines (or shape outlines) to appear as dashed lines.
 Two-color Dashed	Causes lines (or shape outlines) to appear as two-color dashed lines.
Width	Defines the width of the line.
Pattern	Used to define the pattern of the dashes. You can specify the pattern as a series of comma-separated integers that represent the length of each dash or space segment in the dashed line.
 Scaled	Indicates whether the dashes scale as the symbol is zoomed in and out or whether the dashes stay the same length as the symbol is zoomed in and out.

On the lower left-hand side of the Symbology Editor window are several color tool options, as shown below:

Option	Description
 FG	The foreground color. This color defines the color of the line or outline of a shape. When using a pattern, it defines the color that begins the pattern. To change the foreground color of a drawing object, click on the FG color box and select a new color on the color palette.
 BG	The background color. This color defines the body color (filled section of a filled shape) or the second color in a pattern. To change the background color of a drawing object, click on the BG color box and select a new color on the color palette.
 Current Object	Lets you change the selected color (foreground) of the current object without changing the settings in the FG and BG color boxes. To change a color, click on the Current Object color box and select a new color on the color palette.
 Override	This toggle button controls whether the color changes or not when the symbol is selected.

At the bottom of the Symbology Editor window are other miscellaneous options:

Option	Description
 Scaling	Opens the Scale dialog box, which lets you select the minimum X, Y coordinates and the maximum X, Y coordinates for the symbol. It also lets you set the size of the grid mesh (the space between grid lines, in number of pixels) and toggle grid snapping on and off.
 Anchor Points	Opens the Anchor Points dialog box for you to specify the locations where the symbol is anchored (connected) to the map graphics.
 Focus Point	Opens the Focus Point dialog box, used to change the X and Y coordinates that designate the symbol's center. You can enter the new X and Y coordinates or reset them to their original values.
Comment	Text field where you can add notes about the symbol.
Color Palette	Set of 100 colors you can choose from when selecting foreground, background, and Current Object colors.

The following table describes the Symbology Editor File menu options.

Option	Description
New	Clears the drawing area and fills the Number field automatically with a new symbol number.
Save	Saves the current symbol. **Note:** When symbols are saved, they are only applied to the local client workstation. In order to use the symbols throughout the network of workstations, the symbology file needs to be distributed to all client workstations that use symbology.
Clear	Clears the drawing area.
Delete	Deletes the current symbol from the database.
Print	Opens the Print dialog box, letting you print the currently displayed symbol.
Exit	Closes the Symbology Editor window.

The following table describes the Symbology Editor Edit menu options.

Option	Description
Edit Text	Opens the Edit Symbology Text dialog box, letting you edit the text file that defines the symbol.
Action step through	Steps through the actions taken to create the symbol.
Delete last action	Deletes or "undoes" the last action. (This does not include color changes.)
Delete last full action	Deletes or "undoes" the last action, including color changes.

The following table describes the Symbology Editor View menu options.

Option	Key Stroke	Num Lock	Description
Zoom Out	-	N/A	Zooms out from the current view.
Zoom In	+	N/A	Zooms in closer to the current view.
Pan Up	Keypad 8	On	Repositions the symbol up.
Pan Down	Keypad 2	On	Repositions the symbol down.
Pan Left	Keypad 4	On	Repositions the symbol left.
Pan Right	Keypad 6	On	Repositions the symbol right.
Pan NW	Keypad 7	On	Repositions the symbol up and to the left.
Pan NE	Keypad 9	On	Repositions the symbol up and to the right.
Pan SW	Keypad 1	On	Repositions the symbol down and to the left.
Pan SE	Keypad 3	On	Repositions the symbol down and to the right.
Maximum View	Keypad 0	Off	Zooms out fully to display the entire symbol.

Editing a Symbol

To edit a symbol, complete these steps:

1. From the Symbology Viewer, select the symbol you want to edit, and then select **Symbology Editor** from the Edit menu. The Symbology Editor opens displaying the symbol.
2. Remove unwanted objects in the symbol using the Edit menu or the **Clear** option on the File menu.
3. Add objects to the symbol using the buttons and dialog boxes.
4. Save the symbol by selecting **Save** from the File menu. A Confirm dialog box opens.
5. Click **Yes** in the Confirm dialog box. Your revisions are saved.

Creating a New Symbol

To create a new symbol, complete these steps:

1. From the Symbology Viewer, select **New** from the File menu. The Symbology Editor opens displaying a new symbol number in the Number field.
2. Add objects to the symbol using the buttons and dialog boxes.
3. Save the symbol by selecting **Save** from the File menu. A Confirm dialog box opens.
4. Click **Yes** in the Confirm dialog box. Your new symbol is saved.

Deleting a Symbol

To delete a symbol, complete these steps:

1. From the Symbology Viewer, select the symbol you want to delete.
2. Select **Symbology Editor** from the Edit menu. The Symbology Editor opens displaying the symbol.
3. Select **Delete** from the File menu. A dialog box opens.
4. Click **OK** in dialog box to confirm that you want to delete the symbol. The Symbology Editor becomes blank and the symbol is deleted from the database.

Scaling a Symbol

To scale a symbol, complete these steps:

1. From the Symbology Viewer, select the symbol you want to scale and select **Symbology Editor** from the Edit menu. The Symbology Editor opens displaying the symbol.
2. Click on the **Scaling** button (). The Scale dialog box opens.
3. If you want to scale the symbol using minimum and maximum X, Y coordinates, enter them into the Min X, Min Y, Max X, and Max Y fields.

Or

If you want to scale the symbol by altering its grid, enter the number of pixels you would like to appear between grid lines into the Grid Mesh field.

4. Click either **Apply** (to save) or **OK** (to save and close the Scale dialog). The symbol resizes in the drawing area.

Setting Anchor Points for a Symbol

An object can have either one or two anchor points. Objects with two anchor points are always scaled on a map to fit between the two anchor points. To set the anchor points for a symbol, complete these steps:

1. From the Symbology Viewer, select the symbol you want to set anchor points for and select **Symbology Editor** from the Edit menu. The Symbology Editor opens displaying the symbol you selected.
2. Click on the **Anchor** button (). The Anchor Points dialog box opens.
3. Enter the initial anchor point coordinates into the **A1 X** and **A1 Y** fields.
4. If you want the symbol to scale itself within two anchor points on a map, enter the second anchor point coordinates in the **A2 X** and **A2 Y** fields.

5. Click either **OK** (to save) or **Apply** (to save and close the Anchor Points dialog). The symbol repositions in the drawing area to the coordinates you have entered and the anchor point(s) are set.

   Note: For point symbols, the first X,Y coordinates define the only anchor point and the second X,Y coordinates are not used.

Setting the Center of a Symbol

To set the center of a symbol, complete these steps:

- From the Symbology Viewer, select the symbol you want to center and select Symbology Editor from the Edit menu. The Symbology Editor opens displaying the symbol.
- Click on the **Focus Point** button (). The Focus Point dialog box opens.
- Enter the X, Y coordinates for the center of the symbol in the **Focus X** and **Focus Y** fields.
- Click either **OK** (to save) or **Apply** (to save and close the Focus Point dialog). The symbol recenters in the drawing area.

  Note: You can reset the original center coordinates at any time by clicking the **Reset** button in the Focus Point dialog box.

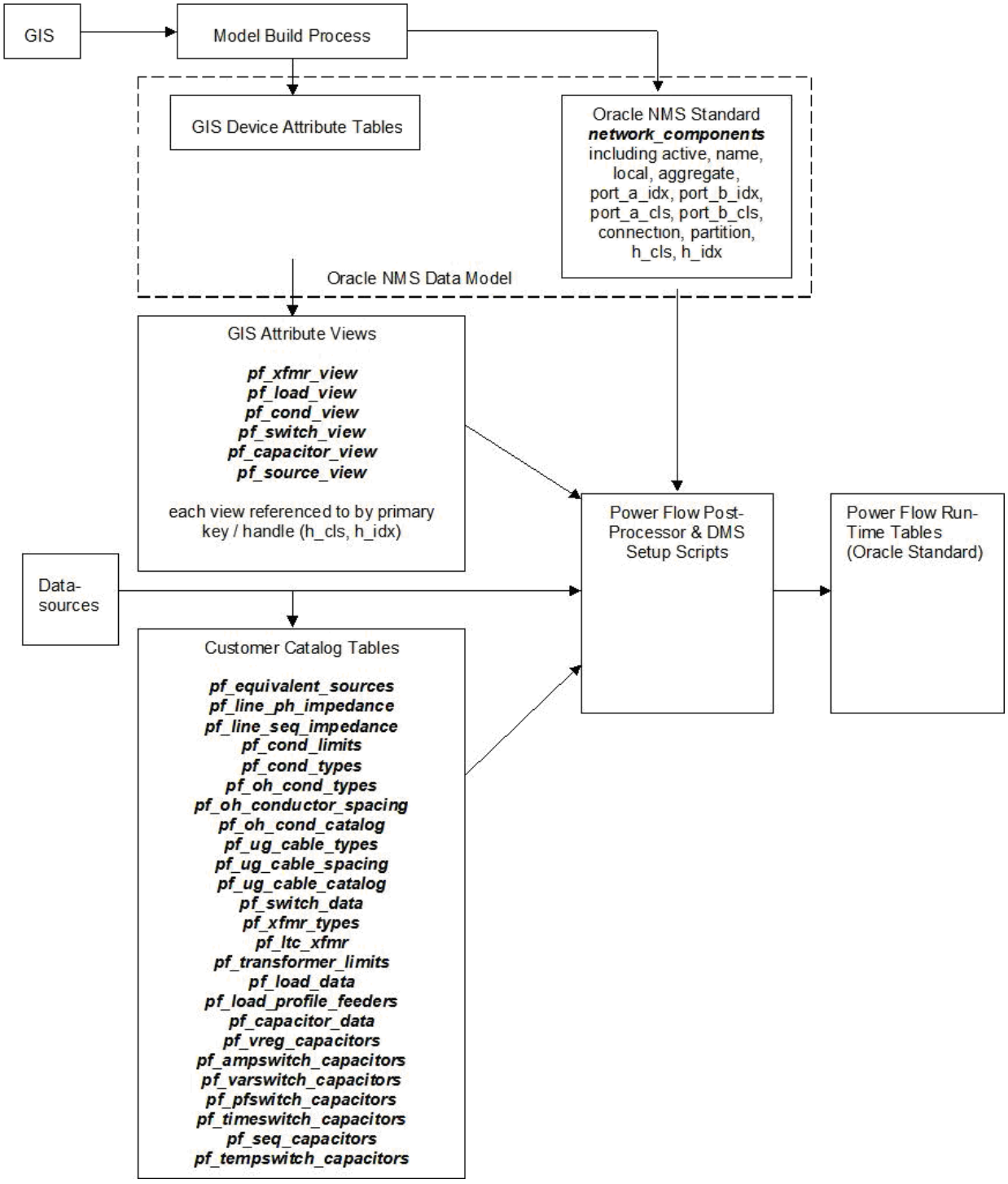
Power Flow Data Requirements and Maintenance

This section describes the Power Flow Extensions Data Input process and the customer data requirements and associated database schema. The data will be used by all DMS applications, including Power Flow Extensions, Suggested Switching, Optimal Power Flow, Feeder Load Management, Short Circuit Analysis, and Fault Location Analysis. It is intended to assist the customer during the Power Flow Extensions data modeling process. It is meant as an introduction, rather than a comprehensive reference. It includes the following subsections:

- Power Flow Extensions Data Import Process – this section describes the basic process and data sources used during the data import process.
- Modeling Device Data – this section explains the basic data requirements and relevant database tables required to model device data.
- Modeling Load Data –this section explains the basic data requirements and relevant database tables required to model load data.
- Catalog tables, Required GIS attributes, Power Flow Attribute Views, Default Data Tables, and Configuration Tables – these sections contain the table schemas required for the Power Flow Extensions.
- Power Flow Engineering Data Workbook – this section describes the power flow engineering workbook contents, and workbook maintenance and update process
- Power Flow Postprocessor – this section describes the power flow post processor functionalities and how to run the postprocessor to populate the run-time tables for power flow extensions, and other DMS applications.
- Power Flow Engineering Data Maintenance – this section provides a list of high-level command line options that can be used with PFService to dynamically update the engineering data

Power Flow Extensions Data Import Process

The figure below details the Power Flow Extensions Data import process. The Power Flow Post-Processor has two dependencies: The device type data in the Catalog Tables and the completion of the model build process.



The model build process creates the Oracle Utilities Network Management System data model, which contains the network_components table and GIS attribute tables.

The catalog tables contain electrical characteristic data for represented device types in the electrical distribution system. Customer data for these catalog tables are captured through the Power Flow Engineering Data workbook

The device attribute views are created based on data in the GIS attribute tables and the customer-provided catalog tables. These views map each relevant network device to class and index attribute keys in the *network_components* table. They may have to be tailored as per the project and the availability of data in both the GIS and catalog tables. The data requirements for each device type are discussed in the *Modeling Device Data* section below.

The Power Flow PostProcessor processes the data provided in the network_components table, the device attribute views, and the catalog tables, performs any unit conversions, and calculates impedances where necessary. If the Power Flow PostProcessor determines that data is missing or incomplete from entries in the catalog or devices attribute tables, a warning message is generated and default data is substituted. The default data values are supplied by Oracle. The Power Flow PostProcessor then writes the results into the Power Flow Extensions run-time tables in the Oracle Utilities Network Management System data model. After successful completion of the Power Flow PostProcessor, Power Flow Extensions may be used in the Oracle Utilities Network Management System user environment.

Modeling Device Data

The creation of device-specific attribute data-sets (either database views or tables) is an important step in the Power Flow Data Modeling process. As mentioned earlier, this process relies on the availability of data from two sources, viz., GIS attribute tables and device catalog tables. This section discusses the device information that is used for this process.

Sources

The customer must provide an equivalent source model for each source node defined in the data model. These source nodes are used to determine energization of the system and generally are located at feeder heads, the substations secondary bus, or the substation primary side bus. The required equivalent source parameters must represent an equivalent impedance looking up into the transmission system from the source node in question. It is possible to model equivalent sources as a zero impedance infinite bus. If sources are modeled with zero impedance (i.e. 'infinite bus'), this will impact the accuracy of short circuit calculations provided by the Power Flow Extensions. During the model build process, source nodes are defined for each feeder head or generation source.

The *pf_source_view* is created based on the data available in the source GIS attribute table (*att_generator*), the device id tables (*network_nodes, feeders*) and the customer catalog table (*pf_equivalent_sources*)

Overhead Lines

The customer must provide overhead line data in one of three ways.

- The first and preferred way is to provide the phase impedance data for each overhead line. The phase impedance data must be the self and mutual impedance and shunt susceptance for each phase. The phase impedance data must be provided in the *pf_line_ph_impedance* table. Oracle Utilities Network Management System Power Flow Extensions supports the modeling of symmetric and asymmetric overhead lines. Lines are considered symmetric when the 3 phases have the same conductor type. Lines are considered asymmetric when the 3 phases have at least two different conductor types.

- The second way is to provide the sequence impedance data for each overhead line. The sequence impedance table data must be the positive and zero sequence impedances and shunt susceptance for each overhead line. The sequence impedance data must be provided in *pf_line_seq_impedance* table.
- The third way is to provide the conductor and construction types for each overhead line. This data must be provided in the *pf_cond_types*, *pf_oh_cond_types*, *pf_oh_conductor_spacing*, and *pf_oh_cond_catalog tables*. *pf_cond_types* contains the conductor characteristics. *pf_oh_cond_types* contains phase-wise conductor type description (as catalogued in the *pf_cond_types* table). The *pf_oh_conductor_spacing* table contains the phase-wise spacing (coordinates) information for an overhead line type. The *pf_oh_cond_catalog* table contains the conductor types (as per *pf_oh_cond_types*) and conductor spacing type (as per *pf_oh_ conductor_spacing*) mappings of all overhead lines.

The GIS attribute table for overhead lines is *att_oh_elec_line_seg.*

In the third option, the Power Flow PostProcessor uses Carson's Modified Equations to calculate the line segment impedances for the overhead line.

Underground Cables

Similarly, the customer must provide underground cable data in one of three ways.

- The first and preferred way is to provide the phase impedance data for each underground cable. The phase impedance data must be the self and mutual impedance and shunt susceptance for each phase. The phase impedance data must be provided in the *pf_line_ph_impedance* table. The Oracle Utilities Network Management System Power Flow Extensions supports the modeling of asymmetric underground cables. Cables are considered asymmetric when the 3 phases have at least two different conductor types. Cables are considered symmetric when the 3 phases have the same conductor type.
- The second way is to provide the sequence impedance data for each cable. This data must be provided in the *pf_line_seq _impedance* table.
- The third way is to provide the conductor and construction types for each underground cable. This data must be provided in the *pf_cond_types*, *pf_ug_cable_types*, *pf_ug_cable_spacing*, and *pf_ug_cable_catalog* tables. *pf_cond_types* contains the conductor characteristics. *pf_ug_cable_types* contains phase & neutral conductor type description (as catalogued in the *pf_cond_types* table). The *pf_ug_cable_spacing* table contains the phase-wise spacing (coordinates) information for an underground cable type. The *pf_ug_cable_catalog* table contains the cable types (as per *pf_ug_cable_types*) and cable spacing type (as per *pf_ug_cable_spacing*) mappings of all underground cables. This data is necessary for the Power Flow PostProcessor to calculate the cable impedances. Calculations based on construction can only be performed for concentric neutral cables.

The GIS attribute table for underground cables is att_ug_elec_line_seg

Transformers and Regulators

Oracle Utilities Network Management System Power Flow Extensions supports explicit modeling of multiple forms of power transformers, such as auto transformers, load-tap-changing transformers, step-up/step-down transformers and regulators. Each of these types of transformers and regulators require transformer characteristic data provided in the *pf_xfmr_types* and *pf_ltc_xfmr* tables. The GIS attribute tables for transformers are *att_power_transformer* and *att_voltage_regulator*, based on which the device attribute *pf_xfmr_view* is created.

Capacitors and Reactors

Shunt (capacitor/reactor) parameters must be provided to the Power Flow PostProcessor for each device in the data extracted from the customer's data source(s), typically the GIS. These parameters will be accessed by the Power Flow PostProcessor through the *pf_capacitor_view*.

Oracle supports the following types of shunt regulation:

- **Voltage switched capacitors**: local or remote bus regulation
- **Current switched capacitors**: local line or cable regulation
- **KVAR switched capacitors**: local line or cable regulation
- **Power-factor switched capacitors**: local line or cable regulation
- **Temperature switched capacitors**: on and off temperature
- **Time of day switched capacitors**: on and off time of day.
- **Fixed capacitor: no regulation**
- **Capacitor sequence control**

The supplied data for each shunt must indicate which type of regulation is to be used and the corresponding control attributes.

Modeling Loads

Load modeling consists of basic load data which is used to determine average loading levels and load profile data which is used to provide detailed information for load variation over time.

Basic Load Data

During modeling efforts, loads must be assigned to specific equipment types. The preferred approach is to insert a load (supply node) at the secondary of each underground and overhead distribution transformer. Supply nodes may also be created at primary metering points for cases where there is no transformation or the transformation is unknown or customer-owned.

For each load, a utilization factor must be determined, which represents the average loading level for the rated size of the transformer. For the most accurate power flow results, this data should be based on per-instance consumption data, which can often be obtained from historical billing information by dividing the total energy consumption of attached customers by the billing period and transformer rating.

The power flow data for load is mapped from the attribute data by using the view pf_load_view, which is based on data in the GIS attribute tables: *att_transformer, att_meter* and *supply_nodes*.

Load Profile Data

Load profile data is used to model how load changes over time, A single load profile represents the change in load levels over a 24-hour period. Multiple profiles may be associated with a single load to represent different load behavior for different types of day (e.g. weekday, weekend) and for different seasons. The use of load profile data improves the accuracy of the DMS applications by providing more realistic loading scenarios for the current or predicted analysis time period. For example, profiles are used to verify switch plans, determine suggested switching recommendations, and to generate daily and seasonal peak limit alarms.

The Oracle Utilities Network Management System supports three different sources of load profile data: feeder profiles, load class profiles, and individual transformer profiles. Customers must choose one of these data sources to be used at any one time. It is not possible to mix different types of load profiles in the same model. The LOAD_PROFILE_DATA_SOURCE parameter in the *pfs_rules* table is set to determine which profile is active. The post-processor tool must also be configured to enable the functionality for updating the currently active type of load profile.

Once processed, all profile data is placed in the *pf_load_interval_data* and *pf_loadtype_data* tables. The profile_id field of the *pf_loads* table points to the data in these tables.

Feeder Profiles

Feeder load profiles are derived from historical feeder-head measurements, and typically obtained from SCADA system data. These profiles will be applied to all downstream transformers which will have the feeder load allocated to them prorated by transformer kVA and the pre-existing utilization factor. In this manner, any predefined relationship between loading levels of different transformers in a feeder will be retained.

In order to use feeder load profiles effectively, a full set of feeder data must be used to populate the *pf_load_profile_feeder* table. A full set of data comprises measurement data for at least one feeder upstream of each load point. For example, if a high voltage feeder supplies a low voltage feeder, then measurement data must be supplied for at least the high voltage feeder and optionally for the attached low voltage feeder (so load can be allocated independently for each feeder).

The biggest challenge to collecting and using feeder profile data is ensuring that the system configuration for which the data is gathered matches the nominal system configuration for the NMS model.

To use feeder profiles, set pfs_rules:LOAD_PROFILE_DATA_SOURCE=0 and configure the product_pf_fdr_profiles.ces script to have ACTIVE='Y' in the *PF_POSTPROCESSOR_PROCS* table.

Load Class Profiles

Load class profiles represent typical load changes over time for a particular type or class of load, such as residential, commercial and industrial. This type of profile data can be obtained from general sources, or the customer can collect this data from typical customers or feeders. When using load class profiles, the load level at each load point is determined by combining the rated kVA with the load utilization factor and the class profile associated with that load. Load class profiles are useful where detailed data for each load is unavailable.

To use feeder profiles, set pfs_rules:LOAD_PROFILE_DATA_SOURCE=1 and configure the prod_model_pf_hourly_load_profileloader_fromsql.ces script to have ACTIVE='Y' in the *PF_POSTPROCESSOR_PROCS* table.

Transformer Profiles

Modern Meter Data Management (MDM) systems make it possible to collect detailed power usage histories for each customer. By aggregating individual meter loads to each service transformer, it is possible to create detailed load profiles for each transformer location. This data can be derived from either representative historical conditions or using predictive values if the MDM system has this capability.

When using transformer profiles, all load data is derived from these profiles and basic load data such as the utilization factor is not used. The load profile input data can include both kW and kVAr values for load over the 24 hour period.

To use feeder profiles, set pfs_rules:LOAD_PROFILE_DATA_SOURCE=2 and configure the prod_model_pf_hourly_load_profileloader_fromcsv.ces script to have ACTIVE='Y' in the *PF_POSTPROCESSOR_PROCS* table. The postprocessor expects the profile data to be input using one or more CSV files placed in the NMS_LOADPROFILE_DIR directory. These CSV files can contain either a full set of data or just partial data (either selected transformers/feeders or selected daytypes).

Catalog Tables

The catalog tables identified in this section must all be populated by the customer. The Power Flow Data Engineering excel workbook should be used as a template to assist the customer to identify source data locations (planning power flow data, database tables etc.), define a data export mechanism, and the Oracle table names, columns, and data formats, into which the source data must be imported.

Equivalent Source Model

pf_equivalent_sources

pf_equivalent_sources: contains the impedance data and voltage set point data representing an equivalent model of transmission system from each source node.

pf_equivalent_sources			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_ID	Varchar2	None	Unique identifier for each equivalent source. Mapping: with feeder::feeder_name (usually with network_nodes::name)
Pos_seq_react	Number	0.09	Positive sequence source reactance in Ohms(X1)
Pos_seq_resist	Number	0.07	Positive sequence source resistance in Ohms (R1)
Zero_seq_react	Number	1	Zero sequence source reactance in Ohms (X0)
Zero_seq_resist	Number	0.1	Zero sequence source resistance in Ohms (R0)
substation_name	Varchar	None	Substation name
kva_rating	Number	10000	KVA rating
nominal_voltage_kv	Number	25	Nominal Voltage in kV
reg_voltage_factor	Number	1.05	Regulated voltage in PU
angle	Number	0	Relative source angle
Connection	Number	1	The connection type for the source: 0: wye (or star) 1: grounded wye (or star)

Phase Impedance Data for Lines and Cables

pf_line_ph_impedance

pf_line_ph_impedance: contains the phase impedances for any conductor (Za, Zb and Zc).

pf_line_ph_impedance

Attribute	Data Type	Default Value	Description
Catalog_Id	Varchar2	None	Line/Cable type alias
RA	Number	0.6295	Resistance of the self impedance on phase A, ohms/ft
RB	Number	1.5876	Resistance of the self impedance on phase B, ohms/ft
RC	Number	1.1075	Resistance of the self impedance on phase C, ohms/ft
XA	Number	1.3106	Reactance of the self impedance on phase A, ohms/ft
XB	Number	1.4818	Reactance of the self impedance on phase B, ohms/ft
XC	Number	1.472	Reactance of the self impedance on phase C, ohms/ft
RAB	Number	0.22	Resistance of the mutual impedance between A and B phase, ohms/ft
RBC	Number	0.2195	Resistance of the mutual impedance and B and C phase, ohms/ft
RAC	Number	0.218	Resistance of the mutual impedance between A and C, ohms/ft
XAB	Number	0.6331	Reactance of the mutual impedance between A and B phase, ohms/ft
XBC	Number	0.6511	Reactance of the mutual impedance between B and C phase, ohms/ft
XAC	Number	0.6511	Reactance of the mutual impedance between A and C ohms/ft
Rmn	Number	0.2195	Resistance of the neutral mutual impedance, ohms/ft
Xmn	Number	0.6331	Reactance of the neutral mutual impedance, ohms/ft
BA	Number	5.2239	Line charging susceptance of phase A for the unbalance line, ohms/ft
BB	Number	4.809	Line charging susceptance of phase B for the unbalance line, ohms/ft
BC	Number	4.9653	Line charging susceptance of phase C for the unbalance line, ohms/ft
Bn	Number	0	Line charging susceptance of neutral for the unbalance line, ohms/ft
BM	Number	0	The mutual line charging susceptance for the unbalance line, ohms/ft

Sequence Impedance Data for Lines and Cables

pf_line_seq_impedance

pf_line_seq_impedance: contains the sequence impedances for any conductor.

pf_line_seq_impedance			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_Id	Varchar2	None	Line/Cable type alias
R1	Number		Positive sequence resistance (ohms/ft)
X1	Number		Positive sequence reactance (ohms/ft)
R0	Number		Zero sequence resistance (ohms/ft)
X0	Number		Zero sequence reactance (ohms/ft)
B1	Number		Positive sequence susceptance (S/ft)
B0	Number		Zero sequence susceptance (S/ft)

Conductor and Construction Type Data for OH Lines

pf_cond_types

pf_cond_types: contains the conductor characteristics.

pf_cond_types			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Conductor type unique identifier (e.g. 336 ACSR)
Diameter	Number	0.721	Conductor Diameter in inches
Resistance	Number	0.306	Conductor Resistance in ohms/mile
GMR	Number	0.0244	Geometric Mean Radius of conductor in feet

pf_oh_cond_types

pf_oh_cond_types: contains phase-wise conductor type description (as catalogued in the pf_cond_types table) and conductor type mappings of all phases of an overhead line type.

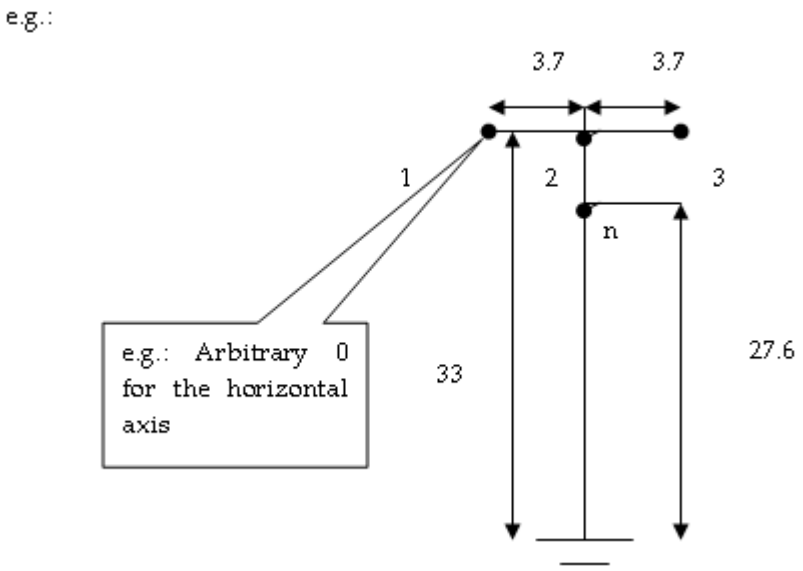
pf_oh_cond_types			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Overhead line construction type unique identifier
Catalog_ID_A	Varchar2	None	Phase A conductor construction type unique identifier
Catalog_ID_B	Varchar2	None	Phase B conductor construction type unique identifier
Catalog_ID_C	Varchar2	None	Phase C conductor construction type unique identifier
Catalog_ID_N	Varchar2	None	Neutral conductor construction type unique identifier

pf_oh_conductor_spacing

pf_oh_conductor_spacing: contains the phase-wise spacing (coordinates) information for an overhead line type.

pf_oh_conductor_spacing			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Conductor spacing type unique identifier
X1	Number	None	Phase A conductor x coordinate (in feet)
Y1	Number	None	Phase A conductor y coordinate
X2	Number	None	Phase B conductor x coordinate
Y2	Number	None	Phase B conductor y coordinate
X3	Number	None	Phase C conductor x coordinate
Y3	Number	None	Phase C conductor x coordinate
X_N	Number	None	Neutral conductor x coordinate
Y_N	Number	None	Neutral conductor y coordinate

The position of conductor defines the horizontal and vertical positions of the centers of the conductors. The vertical distances are always measured from the earth's surface, while the horizontal distances are measured from an arbitrary point, for example: X1=0. The Y_N must be different that zero; otherwise we cannot calculate B0.



pf_oh_cond_catalog

pf_oh_cond_catalog: contains the conductor types (as per pf_oh_cond_types) and conductor spacing type (as per pf_oh_ conductor_spacing) mappings of all overhead lines.

pf_oh_cond_catalog

Attribute	Data Type	Default Value	Description
Catalog_ID	Varchar2	None	Overhead line unique identifier
OH_COND_TYPE_ID	Varchar2	None	Line construction type unique identifier
OH_COND_SPACING_ID	Varchar2	None	Line spacing type unique identifier

Conductor and Construction Type Data for UG Cables

pf_ug_cable_types

The pf_ug_cable_types table contains the cable physical material characteristics and conductor-type mappings for phases and neutral.

Pf_ug_cable_types			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Unique identifier for each cable type
Cond_catalog_ID	Varchar2	None	Unique identifier for phase cable type
Cond_catalog_ID_N	Varchar2	None	Unique identifier for the neutral cable type
neutral_num	Varchar2	13	Number of the concentric neutral in the conductor
Ri	Varchar2	0	Inner radius of lead sheath in inches or in cm
Ro	Varchar2	0	Outer radius of lead sheath in inches or in cm
Dielectric	Varchar2	3	Dielectric constant of the cable insulation
Cable_type	Varchar2	0	0 is a concentric neutral cable. 1 is a parallel-Laid (Single) tape shield cable 2 is a three conductor cable
Idiam	Varchar2	1.29	Diameter cable over the insulation, in inches

pf_ug_cable_spacing

The pf_ug_cable_spacing table contains the cable construction characteristics, such as the phase=wise coordinate information of an underground cable type.

pf_ug_cable_spacing			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Unique identifier for each cable
D_AB	Number	8	Distance between conductor A and B (inches/cm)
D_BC	Number	8	Distance between conductor B and C (inches/cm)
D_AC	Number	16	Distance between conductor C and A (inches/cm)

pf_ug_cable_catalog

The pf_ug_cable_catalog table contains the conductor type and spacing type mappings for all underground cables.

pf_ug_cable_catalog			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2	None	Unique identifier for each cable
UG_Cable_Type_ID	Varchar2	None	Key to the pf_ug_cable_types table
UG_Cable_Spacing_ID	Varchar2	None	Key to the pf_ug_cable_spacing table

Line-Cable Limits

pf_cond_limits

The pf_cond_limits contains the multiple rating or limits for the branch flows based on seasons for the transformers and the conductors.

pf_cond_limits			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	varchar2(40)	None	Unique identifier for each conductor type
Season	Number	None	Season Index: 1 for Winter, 2 for Spring, 3 for Summer and 4 for Fall. Depending upon the number of seasons configured
Limit Type	Number	None	1 for Normal, 2 for Emergency.
Limit	Number	None	Conductor current limit.

Switch-Fuse Limits

pf_switch_data

The pf_switch_data contains the multiple rating or limits for the switches/fuses based on seasons for the transformers and the conductors.

pf_switch_data			
Attribute	**Data Type**	**Default Value**	**Description**
Catalog_ID	Varchar2		Switch/Fuse Identifier
Feeder_ID	Varchar2		ID of the feeder associated with the switch/fuse
Amp_Limit	Number	100	Ampacity of the switch/fuse

Transformers and Regulator Type Data

pf_xfmr_types

The pf_xfmr_types table contains the power, step, auto transformer, and voltage regulator characteristics.

pf_xfmr_types			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_ID	Varchar2	None	Unique identifier for each transformer type
r1	Number	0.035	Positive sequence series resistance in PU on transformer rating
x1	Number	0.024	Positive sequence series reactance in PU on transformer rating
r0	Number	0.0355	Zero sequence series resistance in PU on transformer rating
x0	Number	0.024	Zero sequence series reactance in PU on transformer rating
Prim_ratio	Number	1	Primary ratio, (nominal voltage)/(tap voltage). Range: >=0
sec_ratio	Number	1	Secondary ratio, (nominal voltage)/(tap voltage). Range: >=0
Status	Number	1	0=no taps, 1=fixed, 2= controlled taps, 3=grouped taps
Rated_kva	Number	500	Nominal Kva rating.
Ground_Prim_R	Number	0.01	Grounding resistance in the primary side, ohms
Ground_Prim_X	Number	0.01	Grounding reactance in the primary side, ohms
Ground_Second_R	Number	0.01	Grounding resistance in the secondary side, ohms
Ground_Second_X	Number	0.01	Grounding reactance in the secondary side, ohms
Noloadloss	Number	0.01	The no load losses in the transformers in KW
magpf	Number	0.98	Magnetizing current power factor
pf_LTC_xfmr	Varchar2	None	Key to the pf_ltc_xfmr table
config	Number	D/Y ground (30)	Transformer configuration information. See PSS/U code connection type below.

pf_xfmr_types are referenced from xfmr_in_view xfmr_catalog_id.

pf_xfmr_tanks

The pf__xfmr_tanks table contains the step and auto transformer characteristics.

pf_xfmr_tanks			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_ID	Varchar2	None	Unique identifier for each transformer type
Min_tap_pos	Number	-16	Minimum Tap Position
Num_Steps	Number	32	Number of tap steps. A value of n gives a tap range from 1 to (n+1).
Tapside	Number	1	0 = primary, 1 = secondary
Ldcompr	Number	0	Line drop compensation resistance in PU. Range: -100 to 100
Ldcompx	Number	0	Line drop compensation reactance in PU. Range: -100 to 100
Vtarget	Number	1.0454	Target voltage in PU. Range: 0.5 to 1.5
Tapgroup	Number	-1	Tap Group. Range: >=0 or –1
Tapposition	Number	0	Nominal tap position. Range: Min_tap_pos to (Min_tap_pos+Num_Steps).
Bandwidth	Number	0.00833	Regulation bandwidth
Regbusphase	Number	4	Phase of the regulating bus (A = 1, B = 2, C = 4)

PSS/U Code Connection Type (from PSS/U document)

Xfmr Configuration Type IDs	
2	Wye-Delta -30°
-2	Wye-Delta +30°
3	Delta-Wye +30°
-3	Delta-Wye -30°
4	Open Delta Auto Regulator AB (XY) Open
5	Open Delta Auto Regulator BC (YZ) Open
6	Open Delta Auto Regulator CA (ZX) Open
7	L-L Auto Regulator AB (XY)
8	L-L Auto Regulator BC (YZ)
9	L-L Auto Regulator CA (ZX)
10	Delta-Connected Auto Regulator
11	Delta-Delta

Xfmr Configuration Type IDs	
12	Wye-Connected Auto Regulator
13	Auto Regulator AN (XN)
14	Auto Regulator BN (YN)
15	Auto Regulator CN (ZN)

pf_transformer_limits

The pf_transformer_limits contains the multiple rating or limits for the branch flows based on seasons for the transformers and the conductors.

pf_transformer_limits			
Attribute	**Data Type**	**Default Value**	**Description**
Season	Number	None	Season Index: 1 for Winter, 2 for Spring, 3 for Summer and 4 for Fall. Depending upon the number of seasons configured
Limit_type	Number	None	1 for Normal, 2 for Emergency
Limit	Number	None	Transformer Kva limit

Capacitor Bank Data

pf_capacitor_data

The pf_capacitor_data table contains capacitor size and control type data for each capacitor bank in the data model.

pf_capacitor_data			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_ID	Varchar2	None	Unique identifier for each capacitor bank type
Control_type	Number	1	Type of regulation control: 0 = Fixed 1 = Voltage Regulation 2 = Timer Switched 3 = Temperature Switched 4 = Current Switched 5 = VAR Switched 6 = Powerfactor Switched 7 = Sequence Regulation
kvar_rating_a	Number	400	KVAR rating for phase A
kvar_rating_b	Number	400	KVAR rating for phase B
kvar_rating_c	Number	400	KVAR rating for phase C
nominal_voltage	Number	NA	Base Voltage

pf_capacitor_data			
Attribute	**Data Type**	**Default Value**	**Description**
num_step	Number	1	Number of steps (in case of a switched cap)
nominal_step	Number	1	Nominal step
phase	Number	7	This field indicates which of the phases or neutral are present at the shunt capacitor. Range: 1 to 15 (by using the individual bits)
Connection_config	Number	1	The connection type for the shunt capacitor:
Control_catalog_id	Varchar2	NA	Key to the pf_timeswitch_capacitors, or pf_vreg_capacitors, pf_tempswitch_capacitors, pf_ampswitch_capacitors, pf_varswitch_capacitors, pf_pfswitch_capacitors
reg_branch_id	Varchar2	NA	Regulated Branch ID
reg_branch_terminal	Number	0	Measured Terminal, 0-Port A, 1-Port B
local_or_remote	Number	0	0--local 1--- remote

pf_vreg_capacitors

The pf_vreg_capacitors table contains the regulation settings for each voltage regulated capacitor bank. The PF_VREG_CAPACITORS::CATALOG_ID must match an undue entry in the PF_CAPACITOR_DATA::CATALOG_ID data table column.

PF_VREG_CAPACITORS			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_id	Varchar2	None	Unique identifier for the capacitor type
turn_on_voltage	Number	1.04	Minimum PU voltage before turning on capacitor bank or increasing step position of a multi-step position capacitor bank
turn_off_voltage	Number	1	Maximum PU voltage before turning off capacitor bank or reducing step position of a multi-step position capacitor bank
Reg_phase	Number	7	The phases to be regulated at regbus_idx. It takes zero to regulate all non-neutral phases at the bus. Never pass the neutral phase. Range: 0 to 7

pf_ampswitch_capacitors

The pf_ampswitch_capacitors table contains the regulation settings for each current switched capacitor bank. The PF_AMPSWITCH_CAPACITORS::CATALOG_ID must match a unique entry in the PF_CAPACITOR_DATA::CATALOG_ID data table column.

PF_AMPSWITCH_CAPACITORS

Attribute	Data Type	Default Value	Description
catalog_id	Varchar2	None	Unique identifier for the capacitor type
turn_on_amp	Number	100	Minimum branch current flow before reducing step
turn_off__amp	Number	0	Maximum branch current flow before increasing step
Reg_phase	Number	7	The phases to be regulated at regulated branch. It takes zero to regulate all non-neutral phases at the branch. Range: 0 to 7

pf_varswitch_capacitors

The pf_varswitch _capacitor table contains the regulation settings for each VAR switched capacitor bank. The PF_VARSWITCH_CAPACITORS::CATALOG_ID must match a unique entry in the PF_CAPACITOR_DATA::CATALOG_ID data table column.

PF_VARSWITCH_CAPACITORS

Attribute	Data Type	Default Value	Description
catalog_id	Varchar2	None	Unique identifier for the capacitor type
turn_on_var	Number	0	Minimum branch VAR flow before turning on capacitor bank or increasing step position of a multi-step position capacitor bank
turn_off_var	Number	100	Maximum branch VAR flow before turning off capacitor bank or reducing step position of a multi-step position capacitor bank
Reg_phase	Number	7	The phases to be regulated at regulated branch. It takes zero to regulate all non-neutral phases at the branch. Range: 0 to 7

pf_pfswitch_capacitors

The pf_pfswitch _capacitor table contains the regulation settings for each POWERFACTOR switched capacitor bank. The PF_PFSWITCH_CAPACITORS::CATALOG_ID must match a unique entry in the PF_CAPACITOR_DATA::CATALOG_ID data table column.

PF_PFSWITCH_CAPACITORS

Attribute	Data Type	Default Value	Description
catalog_id	Varchar2	None	Unique identifier for the capacitor type
turn_on_pf	Number	0	Minimum branch power factor before turning on capacitor bank or increasing step position of a multi-step position capacitor bank
turn_off_pf	Number	1	Maximum branch power factor before turning off capacitor bank or reducing step position of a multi-step position capacitor bank
reg_phase	Number	7	The phases to be regulated at regulated branch. It takes zero to regulate all non-neutral phases at the branch. Range: 0 to 7

pf_timeswitch_capacitor

The pf_time_sw_Switches and fuses in the pf_switch_view can be joined to the network_components by h_cls and h_idx, if at the same time, active in ('Y','A'), local <> 'Y', aggregate <> 'A'.

PF_TIMESWITCH_CAPACITORS

Attribute	Data Type	Default Value	Description
catalog_id	Varchar2	None	Unique identifier for the capacitor.
On_hour	Number	8	Hour that capacitor regulation switches ON
On_minute	Number	30	Minute that capacitor regulation switches ON
Off_hour	Number	19	Hour that capacitor regulation switches OFF
Off_minute	Number	30	Minute that capacitor regulation switches OFF
time_var_off_days	Number	65	Days that capacitor regulation will be turned off
Timezone	Varchar2	"CST"	Time zone that capacitor is located in.

pf_seq_capacitor

The pf_seq_capacitor contains electrical characteristics of the sequence capacitor banks and control schemes in the system.

pf_seq_capacitor			
Attribute	**Data Type**	**Default Value**	**Description**
catalog_id	Varchar2	None	Unique identifier for the capacitor
max_kv_factor	Number	1.01	Maximum voltage before reducing step
min_kv_factor	Number	0.9	Minimum voltage before increasing step
sequence_priority	number	1	The priority in which the capacitor will be switched during sequence control.
num_step	Number	10	Total number of steps
Nominal step	Number	1	The initial step of the capacitor
reg_branch_id	Varchar2	None	Regulated Branch ID
reg_branch_terminal	Number	0	Measured Terminal, 0-Port A, 1-Port B
local_or_remote	Number	0	0--local 1--- remote

Load model data

pf_load_data

pf_load_data: contains the customer specific load data including the load utilization factor, load power factor, and load type percents

pf_load_data			
Attribute	**Data Type**	**Default Value**	**Description**
load_ID	Varchar2	None	Unique identifier for each load or customer (Load Transformer Name or ID)
rated_kva_a	Number	100	KVA rating for phase A
rated_kva_b	Number	100	KVA rating for phase B
rated_kva_c	Number	100	KVA rating for phase C
Utilizationfactor	Number	0.6	Utilization factor
Powerfactor	Number	0.9	Power Factor
Const_Imped_Percent	Number	60	Constant impedance load in percent
Const_Power_Percent	Number	20	Constant power load in percent
Const_Current_Percent	Number	20	Constant current load in percent
Profile_id	Varchar2	20	Load profile ID

pf_load_profile_feeder

The *pf_load_profile_feeder* table contains the feeder load data for every defined season, day type and hour of the day. For a typical customer, winter and summer seasons are defined with weekday and weekend day types. The number of periods in each day type is 24. Each feeder head must have an entry for every unique Season: Day_Type, and Period. Entries for lower-voltage feeders supplied from higher voltage feeders (which do have entries) are optional. This data is then used to initialize the *pf_load_interval_data* and *pf_loadtype_data* tables.

pf_load_profile_feeders

Attribute	Data Type	Default Value	Description
Period	Number	None	Number of the period (1-24)
Nom_load	Number	None	KVA nominal load for day type
Day_type	Number	None	Day type
Temperature	Number	None	Nominal temperature

Load Interval data

pf_load_interval_data

pf_load_interval_data: contains the customer specific load profile data for every season, month, day-type and day-interval defined. The values in this table are generated by the post-processor from data derived from either feeder load profiles, load class profiles or individual transformer profiles. For feeder profiles and load class profiles, only the kW field is used (kVAr, KVA, pf are unused). For individual transformer profiles, all fields are populated.

pf_load_interval_data

Attribute	Data Type	Default Value	Description
season	Number	99	Season identifier. Info may be captured in the daytype field itself.
Month	Number	99	Month. Info may be captured in the daytype field itself
daytype	Number	99	Identifier for different day types defined in the powerflow configurations. May or may not be linked with seasonal information
interval	Number	0.6	Time interval (hour of day) – e.g., 0 to 23 for a daily profile with hourly intervals.
kW	Number	0.9	kW interval data, normalized or actual
kVAr	Number	60	kVAr interval data, normalized or actual
kVA	Number	20	kVAr interval data, normalized or actual
Pf	Number	20	Power factor in the interval

Pf_loadtype_data

pf_loadtype_data: contains the definitions of customer-specific load profile types. The values in this table are generated by the post-processor from data derived from either feeder load profiles, load class profiles or individual transformer profiles. There will be one row in this table for each complete profile (24 rows) in the pf_load_interval_data table.

pf_loadtype_data

Attribute	Data Type	Default Value	Description
MPFILE_ID	VARCHAR2(80)	NULL	GIS-based identifier
DAA_ID	VARCHAR2(80)	NULL	DAA (3rd-party system)-based identifier
SOURCE_NORMALIZED	Number	1	Flag that indicates whether the external profile data is normalized.
NORMALIZED	Number	1	Flag that indicates whether the load interval data stored in the OMS/DMS is normalized
SEASON	Number	99	Season identifier
MONTH	Number	99	Month identifier
DAYTYPE	Number	99	Day type identifier
MAX_KW	Number	0	For normalized data, indicates maximum kW interval value in a given profile
MAX_KVAR	Number	0	For normalized data, indicates maximum kVAr interval value in a given profile
MAX_KVA	Number	0	For normalized data, indicates maximum kVA interval value in a given profile
MAX_PF	Number	1	For normalized data, indicates maximum pf interval value in a given profile
UTILFAC	Number	1	Utilization factor for the given profile
INTERVALS	Number	24	Number of intervals in a given profile
ZEROS	Number	0	Number of zero values in a given profile

Required GIS Attributes

The Oracle Utilities Network Management System Power Flow Extensions and Suggested Switching applications require GIS attribution and engineering data for a set of device types. The engineering data requirements are defined above. The GIS attributes importantly map each relevant network device to attribute keys in the Oracle Utilities Network Management System data model and are used to query the engineering data tables.

A definition of the required GIS attribute data used by Power Flow Extensions is provided below for each device type. Typically, required GIS attribution will include the device type identifier, the phases, the voltage, and the device size (KVA rating, KVAR rating, or AMP limit).

The device types used in the Power Flow and other DMS applications are:

- Equivalent sources
- Overhead lines
- Underground cables
- Power transformers and regulators
- Loads (service transformers)
- Switches
- Capacitors

The following sections identify the device attributes that should be mapped from GIS into attribute tables for each device type.

Device Attributes for Equivalent Sources

For Equivalent Sources, the following attributes are required:

Device Attributes for Equivalent Sources

Description	Engineering Data Table::Column	Attribute Data Table::ColumnName	Datatype
Unique identifier for each equivalent source. The PF_EQUIVALENT_SOURCES table contains the impedance data and voltage set point data representing an equivalent model of transmission system from each source node.	Pf_equivalent_sources::catalog_id	Att_generators::feeder_id or Feeders::feeder_name or Network_nodes::h_idx	Varchar2

Device Attributes for Overhead Lines and Underground Cables

For overhead Line and Underground Cable, impedance data can be provided in 1 of 3 formats: phase impedances, sequence impedances, or line and cable conductor and construction types.

Device Attributes			
Description	**Engineering Data Table::Column**	**Attribute Data Table::ColumnName**	**Datatype**
Line Type Identifier: An identifier used to look up the related impedance or conductor/construction type data	Pf_line_ph_impedance::catalog_id or pf_line_seq_impedance::catalog_id or Pf_oh_cond_types::catalog_id Pf_ug_cond_types::catalog_id	Att_oh_elec_line_seg::pf_cond_type or Att_ug_elec_line_seg::pf_cond_type or Att_xx_elec_line_seg::pf_const_type Att_xx_elec_line_seg::u1_pf_cond_type	Varchar2
Phase String (A, B, C, AB, AC, BC, ABC, AN, ABN, ABCN etc)	NA	Att_oh_elec_line_seg::phase_designation Att_ug_elec_line_seg::phase_designation	Varchar2
Nominal voltage	NA	Att_oh_elec_line_seg::nominal_voltage Att_ug_elec_line_seg::nominal_voltage	number
Measured length	NA	Att_oh_elec_line_seg::measured_length Att_ug_elec_line_seg::measured_length	float
Phase Count (1, 2, 3)	NA	Att_oh_elec_line_seg::phase_position Att_ug_elec_line_seg::phase_position	Varchar2

The line type identifier may come from more than one data column in the attribute tables. Below, typical examples are given for the each of the 3 allowed overhead line and underground cable data formats.

Line and Cable Sequence Impedance Attributes

- 3 Phase Conductor Type: ATT_XX_ELEC_LINE_SEG::PF_COND_TYPE
  - Ex 1: “OH: 1/0 CU: 1/0 CU”
  - Ex 2: “OH 795AL”
  - EX 3: “4 ACSR: 2 ACSR” for

Line and Cable Phase Impedance Attributes

- 3 Phase Conductor Type: ATT_XX_ELEC_LINE_SEG::PF_COND_TYPE
  - Ex 1: “OH: 1/0 CU: 1/0 CU” for
  - Ex 2: “OH 795AL”
  - EX 3: “4 ACSR: 2 ACSR”

Line and Cable Conductor and Construction Type Attributes

- Construction Type: ATT_XX_ELEC_LINE_SEG::PF_CONST_TYPE
- 3 Phase Conductors: ATT_XX_ELEC_LINE_SEG::PF_COND_TYPE (Same conductor type for each phase)
- Phase A Conductor Type: ATT_XX_ELEC_LINE_SEG::U1_PF_COND_TYPE
- Phase B Conductor Type: ATT_XX_ELEC_LINE_SEG::U2_PF_COND_TYPE
- Phase C Conductor Type: ATT_XX_ELEC_LINE_SEG::U3_PF_COND_TYPE
- Neutral Conductor Type: ATT_XX_ELEC_LINE_SEG::Neutral_PF_COND_TYPE.

Device Attributes for Power Transformers

For Power Transformer, the following attributes are required:

Device Attributes			
Description	**Engineering Data Table::Column**	**Attribute Data Table::ColumnName**	**Datatype**
Transformer Type Identifier	Pf_xfmr_types::catalog_id	Att_power_transformes::pf_xfmr_type	Varchar2
XFMR Primary Side Volts	NA	Att_power_transformes::nom_prim_voltage_kv	number
XFMR Secondary Side Volts	NA	Att_power_transformes::nom_sec_voltage_kv	number
Substation Name	NA	Att_power_transformes::substation_name	Varchar2
Substation Bus Name	NA	Att_power_transformes::facility_id	Varchar2
Xfmr Configuration Type IDs	Pf_xfmr_types::catalog_id	NA	number

Device Attributes for Switches

For Switches, the following Attributes are required:

Device Attributes			
Description	**Engineering Data Table::Column**	**Attribute Data Table::ColumnName**	**Datatype**
Switch Id	NA	ATT_SWITCH ::FACILITY_ID or ATT_FUSE:: FACILITY_ID or ATT_RECLOSER:: FACILITY_ID or ATT_SECTIONALIZER:: FACILITY_ID	Varchar2
Amp Limit	NA	ATT_XXX ::AMP_SIZE	number
Phase designation	NA	Att_XXX::phase_designation	Varchar2

Device Attributes for Capacitors

For Capacitor, the following attributes are required:

Device Attributes			
Description	**Engineering Data Table::Column**	**Attribute Data Table::ColumnName**	**Datatype**
Capacitor Id	PF_CAPACITOR_DATA:: CAPACITOR_ID	ATT_FIXED_CAPACITOR:: FACILITY_ID or ATT_SWITCH_CAPACITOR:: FACILITY_ID	Varchar2
Control Type	PF_CAPACITOR_DATA:: Control_catalog_id	NA	number
Rated Phase A KVAR	NA	ATT_XXX ::KVAR_RATING_A	number
Rated Phase B KVAR	NA	ATT_XXX ::KVAR_RATING_B	number
Rated Phase C KVAR	NA	ATT_XXX ::KVAR_RATING_C	number
Voltage	NA	Att_XXX::nominal_voltage	Varchar2

Device Attributes for Load Models

For loads, the following attributes are required:

Device Attributes			
Description	**Engineering Data Table::Column**	**Attribute Data Table::ColumnName**	**Datatype**
Load ID	PF_LOAD_DATA::LOAD_ID	ATT_TRANSFORMER ::FACILITY_ID or ATT_METER:: FACILITY_ID	Varchar2
Rated Phase A KVA	NA	ATT_ TRANSFORMER:: RATED_KVA_A or ATT_METER:: METER_RATING	number
Rated Phase A KVA	NA	ATT_ TRANSFORMER:: RATED_KVA_B or ATT_METER:: METER_RATING	number
Rated Phase A KVA	NA	ATT_ TRANSFORMER::RATED_KVA_C or ATT_METER:: METER_RATING	number
Voltage	NA	Att_XXX::NOMINAL_VOLTAGE	Varchar2

Power Flow Attribute Views

The Oracle Utilities DMS Project Engineer creates customer specific attribute data views used to identify the engineering data devices for equivalent sources, overhead lines, underground cables, power transformers, regulators, loads (service transformers), primary meters, switches, and capacitors.

A description of each Power Flow Attribute view is provided below

pf_source_view

The pf_source_view contains the source data. Information about the injection sources is contained in this table.

Attribute Column Name	Data Type	Default	Description
H_CLS	Number	None	Handle class for this source (always 299)
H_IDX	Number	None	Handle index for this source
Pos_seq_react	Number	0.09	Positive sequence source reactance in Ohms(X1)
Pos_seq_resist	Number	0.07	Positive sequence source resistance in Ohms (R1)
Zero_seq_react	Number	1	Zero sequence source reactance in Ohms (X0)
Zero_seq_resist	Number	0.1	Zero sequence source resistance in Ohms (R0)
substation_name	Varchar	None	Substation name
kva_rating	Number	10000	KVA rating
nominal_voltage_kv	Number	25	Nominal Voltage in kV
reg_voltage_factor	Number	1	Regulated voltage in PU
Connection	Number	1	The connection type for the source: 0: wye (or star) 1: grounded wye (or star)

pf_source_view is referenced from network_components by the h_cls, and h_idx, if Active in ('A','Y')

pf_cond_view

The pf_cond_view contains the catalog ID of the three phase conductors and neutral. Also, the catalog ID is used to link with the line and cable catalog table.

Column Name	Data Type	Default	Description
H_cls	Number	None	Handle class for this conductor
H_idx	Number	None	Handle index for this conductor
Feeder_id	Varchar2	None	Feeder id

Column Name	Data Type	Default	Description
Partition	Number	None	Partition
Cond_catalog_id_a	Varchar2	None	Key to the *pf_cond_types* for phase a wire – Required unless default ratings are acceptable
Cond_catalog_id_b	Varchar2	None	Key to the *pf_cond_types* for phase b wire wire – Required unless default ratings are acceptable
Cond_catalog_id_c	Varchar2	None	Key to the *pf_cond_types* for phase c wire wire – Required unless default ratings are acceptable
Cond_catalog_id_n	Varchar2	None	Key to the *pf_cond_types* for phase n wire wire – Required unless default ratings are acceptable
Cond_imped_catalog_id	Varchar2	None	This field is required unless default impedances are acceptable
			Key to the pf_oh_cond_catalog,
			Key to the pf_ug_cabe_catalog,
Length	Number	100	Length of a line/cable in feet/meter. Defaults 100 feet/meters
oh_ug_type		None	0: OH/UG Phase impedance is provided
			1: Overhead spacing is provided
			2: OH/UG Sequence impedance is provided
			3: Underground cable spacing is provided
nominal_voltage_kv	Number	12.47	Base KVoltage on conductor or cable

Conductors and Cables in the pf_cond_view can be joined to the network_components by h_cls and h_idx, if at the same time, active in ('Y','A'), local <> 'Y', aggregate <> 'A'.

The Cond_catalog_id_a/b/c/n fields are keys into the pf_cond_types if the conductor is overhead or the pf_ug_cable_types table if the conductor is an underground cable. These same fields (cond_catalog_id) are also keys into the pf_cond_limits table.

The Cond_imped_catalog_id is a key into any one of the following tables: pf_line_ph_impedance, pf_line_seq_impedance, pf_oh_conductor_spacing, or pf_ug_const_type tables. The table chosen will be based on the method the customer has decided to provide power flow impedance data to the system.

pf_xfmr_view

The pf_xfmr_view contains the substation and bus names that the transformer is connected.

Attribute Column Name	Data Type	Default	Description
h_cls	Number	None	Handle class for this transformer
h_idx	Number	None	Handle index for this transformer
Port_a_cls	Number	None	Handle cls for local node
Port_a_idx	Number	None	Handle idx for local node
Port_b_cls	Number	None	Handle cls for local node
Port_b_idx	Number	None	Handle idx for local node
Nominal_voltage_kv	Number	25.00	Base KVoltage on conductor or cable
Nominal_sec_voltage_kv	Number	12.40	Base Voltage on Secondary1 side of transformer
Substation_name	Varchar2	None	Substation Name
Bus_name	Varchar2	None	Bus Name
Xfmr_catalog_id	Varchar2	None	Key to the pf_xfmr_types

pf_xfmr_view can be joined to the network_components by h_cls = "The Class Number used for transformers", and the h_idx, if at the same time, active in ('Y','A'), local <> 'Y', aggregate <> 'A'.

pf_switch_view

The pf_switch_view contains the switch nominal current information.

Attribute Column Name	Data Type	Default	Description
h_cls	Number	None	Handle class for this conductor or cable
h_idx	Number	None	Handle index for this conductor or cable
Amp_Limit	Number	100	Ampere limit on the switch

Switches and fuses in the pf_switch_view can be joined to the network_components by h_cls and h_idx, if at the same time, active in ('Y','A'), local <> 'Y', aggregate <> 'A'.

pf_load_view

The pf_load_view contains the characteristics on the customer load points, including distribution transformers and primary meters.

Attribute		Default	Description
h_cls	Number	None	Handle Class
h_idx	Number	None	Handle Index
load_zone	Varchar2	1	The zone that the load belongs to
Conforming_load	Number	1	1 if conforming load. 0 if non conforming load
rated_kva_a	Number	100	KVA rating for phase A
rated_kva_b	Number	100	KVA rating for phase B
rated_kva_c	Number	100	KVA rating for phase C
Utilfactor	Number	0.63	Utilization factor (Default 0.6)
Powerfactor	Number	0.95	Power Factor (Default 0.8)
Nominal_voltage_kv	Number	12.0	Base Voltage
Const_Imped_Percent	Number	60	Constant impedance load in percent
Const_Power_Percent	Number	20	Constant power load in percent
Const_Current_Percent	Number	20	Constant current load in percent
Connection	Number	1	The connection type for the load: 0: wye 1: grounded wye 2: delta.

pf_capacitor_view

The pf_capacitor_view contains electrical characteristics of capacitors.

Attribute Column Name	Data Type	Default	Description
h_cls	Number	None	Handle class for this capacitor
h_idx	Number	None	Handle index for this capacitor
Control_type	Number	0	Type of regulation control: 0 = Fixed 1 = Voltage Regulation 2 = Timer Switched 3 = Temperature Switched 4 = Current Switched 5 = VAR Switched 6 = Powerfactor Switched 7 = Sequence Regulation
rated_kvar_a	Number	100	KVAR rating for phase A

Attribute Column Name	Data Type	Default	Description
rated_kvar_b	Number	100	KVAR rating for phase B
rated_kvar_c	Number	100	KVAR rating for phase C
Nominal_voltage_kv	Number	12.00	Base Voltage
Phase		7	This field indicates which of the phases or neutral are present at the shunt capacitor,. Range: 1 to 15 (by using the individual bits) Comment: also, we can get the phase from the network_components table.
Connection_config	Number	1	The connection type for the shunt capacitor: 0: wye 1: grounded wye 2: delta:
Cap_catalog_id	Varchar2	None	Key to the pf_time_sw_capacitor, or pf_volt_reg_capacitor, or pf_seq_capacitor

pf_capacitors_view can be joined to the network_components by h_cls, and h_idx., if at the same time, active in ('Y','A'), local <> 'Y', aggregate <> 'A'.

Default Data Tables

pf_default_data

Attribute Column Name	Description	Data	Unit
Shunt Capacitor			
Shunt_Reg_Bus	Regulated bus	1	N/A
Shunt_Type_Control	Control type	0	N/A
Shunt_Num_Step	Total number of steps	10.0000	N/A
Shunt_Steps	Steps	5.0000	N/A
Shunt_Max_KV	Max voltage factor before reducing step	1.0100	N/A
Shunt_Min_KV	Minimum voltage factor before increasing step	0.9000	N/A
Shunt_On_Hour	Hour that capacitor regulation switches ON	8.0000	hr
Shunt_On_Min	Minute that capacitor regulation switches ON	30.0000	min

Attribute Column Name	Description	Data	Unit
Shunt_Off_Hour	Hour that capacitor regulation switches OFF	19.0000	hr
Shunt_Off_Min	Minute that capacitor regulation switches OFF	30.0000	min
Shunt_TIME_VAR_OFF_DAYS	Days that capacitor regulation will be turned off	65.0000	day
Shunt_Time_Zone	Time zone that capacitor is located in.	"CST"	N/A
Shunt_SEQUENCE_PRIRIORITY	The priority in which the capacitor will be switched during sequence control.	1	N/A
Shunt_KVAR_RATING_A	KVAR rating for phase A	100.0000	kVar
Shunt_KVAR_RATING_B	KVAR rating for phase B	100.0000	kVar
Shunt_KVAR_RATING_C	KVAR rating for phase C	100.0000	kVar
Shunt_Nom_Volt	Base Voltage	12.0000	kV
Shunt_Steps	The initial step of the capacitor	1	N/A
Shunt_Connection	The connection type for the shunt capacitor:	1: grounded wye	
Source			
Source_Nom_Volt	Nominal voltage	12.4700	kV
Source_Reg_Volt	Regulated voltage factor in PU	1.0000	PU
Source_X1	Positive sequence source reactance (X1)	0.0900	Ohms
Source_R1	Positive sequence source reactance (R1)	0.0700	Ohms
Source_X0	Zero sequence source reactance (X0)	1.0000	Ohms
Source_R0	Zero sequence source resistance (R0)	0.1000	Ohms
Source_KVA_Rating	KVA rating	1000.0000	kVA
Source_Connection	The connection type	1 (for grounded wye)	N/A
Load			
Load_Zone	The zone that the load belongs to	1	N/A
Load_Conforming	Conforming load	1	N/A
Load_Kva_rating_A	KVA rating for phase A	100.0000	kVA
Load_Kva_rating_B	KVA rating for phase B	100.0000	kVA
Load_Kva_rating_C	KVA rating for phase C	100.0000	kVA
Load_Utility_factor	Utilization factor	0.6300	N/A
Load_Power_Factor	Power Factor	0.9500	N/A

Attribute Column Name	Description	Data	Unit
Load_Nom_Voltage	Base Voltage	12.0000	kVA
Load_Const_Imped_Percent	Constant impedance load in percent	60.0000	%
Load_Const_Power_Percent	Constant Power load in percent	20.0000	%
Load_Const_Current_Percent	Constant current load in percent	20.0000	%
Load_Connection	The connection type for the load:	1	N/A
Load_Num_Phase	Number of phase	3.0000	N/A
Load_Phases	Phases	7.0000	N/A
Switch			
Switch_Amp_limit	Ampere limit on the switch	100.0000	Ampers
Cable			
Cable_Lenth	Length	100.0000	feet
Cable_Nom_Volt	Nominal voltage	12.0000	kV
Cable_R1	Positive sequence resistance	0.0180	ohms/ mile
Cable_X1	Positive sequence reactance	0.0380	ohms/ mile
Cable_R0	Zero sequence resistance	0.2810	ohms/ mile
Cable_X0	Zero sequence reactance	0.0187	ohms/ mile
Cable_B1	Positive sequence shunt susceptance	0.0000	uS/mile
Cable_B0	Zero sequence shunt susceptance	0.0000	uS/mile
Cable parameters			
Cable_Num_Neutral	Number of concentric neutral in the conductor	13.0000	N/A
Cable_Inner_Radius	Inner radius of lead sheath	0.0000	inches
Cable_Outer_Radius	Outer radius of lead sheath	0.0000	inches
Cable_Dielectric	Dielectric constant of the cable insulation	3.0000	N/A
Cable_Diameter	Diameter Cable over the insulation	1.2900	inches
Cable_Type	Cable type	0 (concentric neutral)	
Cable construction type			
Cable_Distance_AB	Distance between conductor A and B	8.0000	inches
Cable_Distance_BC	Distance between conductor B and C	8.0000	inches

Attribute Column Name	Description	Data	Unit
Cable_Distance_AC	Distance between conductor A and C	16.0000	inches
Line			
Line_Length	Length	100.0000	feet
Line_Nom_Volt	Nominal voltage	12.0000	kV
Symmetric Line impedance			
Line_R1	Positive sequence resistance	0.0180	ohms/ mile
Line_X1	Positive sequence reactance	0.0380	ohms/ mile
Line_R0	Zero sequence resistance	0.2810	ohms/ mile
Line_X0	Zero sequence reactance	0.0187	ohms/ mile
Line_B1	Positive sequence shunt susceptance	0.0000	uS/mile
Line_B0	Zero sequence shunt susceptance	0.0000	uS/mile
Asymmetric Line Impedance			
Line_RA	Resistance of the self impedance on phase A	0.6295	ohms/ mile
Line_RB	Resistance of the self impedance on phase B	1.5876	ohms/ mile
Line_RC	Resistance of the self impedance on phase C	1.1075	ohms/ mile
Line_XA	Reactance of the self impedance on phase A	1.3106	ohms/ mile
Line_XB	Reactance of the self impedance on phase B	1.4818	ohms/ mile
Line_XC	Reactance of the self impedance on phase C	1.4720	ohms/ mile
Line_Rn	Resistance of the mutual impedance	0.0000	ohms/ mile
Line_Xn	Reactance of the mutual impedance	0.0000	ohms/ mile
Line_BA	Line charging susceptance of phase A	5.2239	uS/mile
Line_BB	Line charging susceptance of phase B	4.8090	uS/mile
Line_BC	Line charging susceptance of phase C	4.9653	uS/mile
Line_BN	The mutual line charging susceptance	0.0000	uS/mile

Attribute Column Name	Description	Data	Unit
Line_RAB	Resistance of the mutual impedance between phases A and B	0.2200	ohms/ mile
Line_RBC	Resistance of the mutual impedance between phases A and B	0.2195	ohms/ mile
Line_RAC	Resistance of the mutual impedance between phases A and B	0.2180	ohms/ mile
Line_XAB	Reactance of the mutual impedance between phases A and B	0.6331	ohms/ mile
Line_XBC	Reactance of the mutual impedance between phases B and C	0.6511	ohms/ mile
Line_XAC	Reactance of the mutual impedance between phases A and C	0.6441	ohms/ mile
Line_RMN	Resistance of the mutual impedance between phases and Neutral	0.2195	ohms/ mile
Line_XMN	Reactance of the mutual impedance between phases and Neutral	0.6331	ohms/ mile
Line parameters			
Line_GMD_Phase	Geometric mean distance between phase conductor	4.5428	feet
Line_GMD_Neutral	Geometric mean distance between phase and neutral conductors	6.7194	feet
Line_Distance_Phase	Average distance between phase conductors and earth	24.0000	feet
Line_Distance_Neutral	Average distance between neutral conductor and earth	18.0000	feet
Line spacing parameters			
Line_DX_A	Position of the conductor phase A on the horizontal axis(X)	6.0000	feet
Line_DY_A	Position of the conductor phase A on the vertical axis(Y)	2.4000	feet
Line_DX_B	Position of the conductor phase B on the horizontal axis(X)	1.3000	feet
Line_DY_B	Position of the conductor phase B on the vertical axis(Y)	0.2000	feet
Line_DX_C	Position of the conductor phase C on the horizontal axis(X)	1.2000	feet
Line_DY_C	Position of the conductor phase C on the vertical axis(Y)	3.2000	feet
Line_DX_N	Position of the conductor neutral on the horizontal axis(X)	0.7500	feet

Attribute Column Name	Description	Data	Unit
Line_DY_N	Position of the conductor neutral on the vertical axis(Y)	0.9800	feet
Conductor			
Conductor_Resistance	Resistance	1.3515	Ohms
Conductor_Diameter	Diameter of the conductor	0.6350	inches
Conductor_GMR	geometric mean radius of the conductor	1.3320	inches
Transformer			
Transformer_MVA	Transformer Capacity in MVA	0.5000	MVA
Transformer_Prim_Volt	Primary voltage	25.0000	Kv
Transformer_Second_Volt	Secondary voltage	12.4000	Kv
Transformer_R0	R0	0.0355	PU
Transformer_X0	X0	0.0240	PU
Transformer_R1	R1	0.0355	PU
Transformer_X1	X1	0.0240	PU
Transformer_Configuration	Configuration	D/Y ground (-30)	N/A
Transformer_Type	Transformer type	3 phases	N/A
Transformer_Prim_Phase	Primary phase	Triphase ABC	N/A
Transformer_Second_Phase	Secondary phase	Triphase ABC	N/A
Transformer_Power_Factor	Power factor	0.9800	N/A
Transformer_Groud_Prim_R	Ground Resistance in primary side	0.0100	Ohms
Transformer_Groud_Prim_X	Ground Reactance in primary side	0.0100	Ohms
Transformer_Groud_Second_R	Ground Resistance in secondary side	0.0100	Ohms
Transformer_Groud_Second_X	Ground Reactance in secondary side	0.0100	Ohms
Transformer_NoLoad_Losses	No load losses	10.0000	KVA
Transformer_Ratio	Transformer ratio	1.0000	N/A
Transformer_Ration_Second	Ratio on the secondary side	1.0000	N/A
Transformer_Ratio_Max	TAP ratio max	1.0500	N/A
Transformer_Ratio_Min	TAP ratio min	0.9000	N/A
Transformer_Min_TapStep	Minimum TAP steps	0.0000	N/A
Transformer_Tanks_Number	Number of tanks of the transformer	3.0000	N/A
Transformer_Magnnet	Magnetizing current power factor	0.9500	N/A

Attribute Column Name	Description	Data	Unit
Transformer_Sec2Volt	Secondary voltage transformer	0	KV
Transformer_GroundPrim	This fields indicate whether a ground is present on a primary winding. 0 or 1	0	N/A
Transformer_GroundSec1	This fields indicate whether a ground is present on a secondary winding. 0 or	1	N/A
Transformer_GroundSec2	This fields indicate whether a ground is present on a third winding. 0 or 1	0	N/A
Transformer_RatioSecond2	Voltage ratio for the secondary winding transformer	0	N/A
Transformer_TapSide	0 if the tap is on the primary side, or 1 if the tap is on the secondary side	1	N/A
Transformer_LDCompR	The line drop compensation resistance	0	
Transformer_LDCompX	The line drop compensation reactance	0	
Transformer_TargetVolt	Target voltage in Per Unit	12.4	KV
Transformer_TapGroup	Indicates a group to which the primative transformer belongs. A negative values indicates that it does not belong to a group	-1	N/A
Transformer_TapPosition	Current tap position	10	N/A
Transformer_TapStatus	Current tap status. 0= No tap present 1= Fixed tap 2= Controlled tap 3= Grouped taps	1	N/A
Transformer_Phaserotation	Indicates the rotation phase	1	N/A
Transformer_NormRatingKva_A	KVA rating on phase A	160	KVA
transformer_NormRatingKva_B	KVA rating on phase B	160	KVA
Transformer_NormRatingKva_C	KVA rating on phase C	160	KVA
Transformer_EmergKva	Emergency KVA limit	180	KVA

Configuration Tables

pf_seasons

This table will store the seasonal peak load information and define the seasons. One entry in this table is required for every season of every load zone. A load zone consists of a group of all loads that have their load profiles maintained according to the same temperature measurement point. There could be only one load zone for the entire system, but also several. The customer directly populates this table from seasonal data.

Attributes			Description
season_Number	Varchar2	20	Season number
Zone	Varchar2	20	Zone number
Season_start_day	Varchar2	20	Season start day
Season_start_month	Varchar2	20	Season start month
Season_end_day	Varchar2	20	Season end day
Season_end_month	Varchar2	20	Season end month
Season_peak	Varchar2	20	Season peak load in KVA
peak_day	Varchar2	20	Day of seasonal peak load
peak_month	Varchar2	20	Month of seasonal peak load
peak_load_period	Varchar2	20	Load period of seasonal peak
peak_day_type	Varchar2	20	Day type of seasonal peak
peak_temp	Varchar2	20	Peak temperature in °F or °C
nom_temp	Varchar2	20	Nominal temperature in °F or °C
temperature_alias	Varchar2	20	Alias for temperature point in °F or °C

pfs_rules

Attributes			Description
Nom_Temp	Varchar2	50	Nominal temperature for use when maintaining the load profile
DAYTYPE_X	Varchar2	100	To identify the day type of the current day, where 'X' is the number of the day type represented by this field.

DAYTYPE_X Attribute

The following table provides examples for the DAYTYPE_X attributes of the pfs_rules table.

Parameter	Value
DAYTYPE_0	WEEKDAY, SEASON_1, SEASON_2, SEASON_3, SEASON_4, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
DAYTYPE_1	WEEKEND, SEASON_1, SEASON_2, SEASON_3, SEASON_4, SUNDAY, SATURDAY
DAYTYPE_2	SEASONALPEAK,SEASON_1, SEASON_2, SEASON_3, SEASON_4
DAYTYPE_3	DATE, 5-17, 7-4, 12-25
DAYTYPE_4	DEFAULT

Day types can be configured by day name and season or date. Days not specified by date or day name in any of the day type configuration strings are of the default day type, which is DAYTYPE_4.

In the table above:

- All weekdays of season 1 through season 4 are of day DAYTYPE_0
- All weekend days of season 1 through season 4 are of DAYTYPE_1,
- Seasonal peaks of season 1 through season 4 are of DAYTYPE_2
- May 17, July 4, December 25 are of DAYTYPE_3, since date configuration takes precedence over day name configuration.

The customer must specify the names of each day type, which will be used on the graphical user interface where appropriate. For example, DAYTYPE_0 could be 'Typical Weekday', DAYTYPE_1 could be 'Typical Weekend', DAYTYPE_2 could be 'ANNUAL PEAK=', and DAYTYPE_3 could be 'Holiday'.

For more information on pfs_rules table configuration, refer to the **Chapter 14**, **Distribution Management Application Configuration**.

PowerFlow Engineering Data Workbook

The PowerFlow Engineering Data Workbook is an Excel spreadsheet used to gather and manage data required by the PowerFlow extensions and other DMS applications that is not generally available within the GIS and Oracle Utilities Network Management System. The Power Flow Engineering Data Workbook maintains data required to run the Power Flow Extensions, Suggested Switching, Optimal Power Flow, Feeder Load Management, and Fault Location Analysis applications.

The Power Flow Engineering Data Workbook defines the required data types, the data tables, and the table schemas. An MS Excel spreadsheet is used for each data type and its corresponding data table. Tabs (worksheets) in the Excel spreadsheet contain a description of the data table and the data table columns. Each data worksheet also contains one or more user-editable tables the user fills for each device type in the data model. The user simply edits the enterable table, adding a new row for each unique device type. For each completed worksheet, the user generates an SQL formatted ASCII text file from a push button on the TOOLS worksheet. The SQL formatted ASCII text files are used to import the Power Flow Engineering data into the Oracle Utilities Network Management System data model.

Workbook Contents

The contents of the Powerful Engineering Data Workbook are as follows.

- Tools (contains the GUI and underlying macros that generate the customer catalog data)
- Power Flow Engineering Data Dictionary
- GIS Attribute Mapping
- Equivalent Sources
- Line and Cable Phase Impedances
- Line and Cable Sequence Impedances
- Line and Cable Limits
- Conductors Construction
- Line Construction
- Cable Construction
- Switch-Fuse Limits
- Power Transformer Impedance
- Power Transformer Tap Data
- Power Transformer Limits
- Customer Load Models
- Feeder Load Profiles
- Capacitor Data
- Voltage Regulated Capacitors
- Current Switched Capacitors
- VAR Switched Capacitors
- Powerfactor Switched Capacitors
- Time Switched Capacitors
- Sequence Controlled Capacitors

- Temperature Switched Capacitors
- Customer Hourly Load Profiles

Catalog Data SQLs to be Generated

The Power Flow Engineering Data Workbook will generate a set of customer catalog data SQLs, and those files should be installed in the $CES_DATA_FILES directory (~/sql).

Data file	Description
~/sql/<project>_powerflowengineeringdata.xls	This is the latest checked-in version of the Power Flow Engineering Data workbook. To be used for generating the customer catalog data sql files
~/sql/<project>_pf_sources.sql	Contains data pertaining to equivalent source models for the source nodes in the network
~/sql/<project>_pf_phimped.sql	Contains unbalanced phase impedance data for overhead line / underground cable conductors
~/sql/<project>_pf_seqimped.sql	Contains sequence impedance data for overhead line / underground cable conductors
~/sql/<project>_pf_condlimits.sql	Contains multiple ratings/limits for branch flows based on seasons for lines and cables
~/sql/<project>_pf_conductors.sql	Contains construction details of individual conductors
~/sql/<project>_pf_lines.sql	Contains construction details of overhead lines (conductor type mappings for every phase and phasewise coordinate spacing information)
~/sql/<project>_pf_cables.sql	Contains construction details of underground cables (conductor type mappings for every phase and phasewise coordinate spacing information)
~/sql/<project>_pf_switches.sql	Contains nominal ampacity data for switches
~/sql/<project>_pf_load_data.sql	Contains electrical characteristics of customer loads
~/sql/<project>_pf_load_profile_feeder.sql	Contains profiles for a full set of feeders. This data is only used if load profiles are configured to use feeder load profiles.
~/sql/<project>_pf_xfmrtypes.sql	Contains electrical characteristics data for power, step and auto transformers
~/sql/<project>_pf_xfmrtaps.sql	Contains electrical characteristics data for power, step and auto transformers

Data file	Description
~/sql/<project>_pf_xfmrlimits.sql	Contains multiple ratings/limits for branch flows based on seasons for transformers
~/sql/<project>_pf_capacitors.sql	Contains electrical characteristics of capacitors
~/sql/<project>_pf_vregcap.sql	Contains electrical characteristics of voltage-regulated capacitors
~/sql/<project>_pf_timeswitchcap.sql	Contains electrical characteristics of time-switched capacitors
~/sql/<project>_pf_ampswitchcap.sql	Contains electrical characteristics of current-regulated capacitors
~/sql/<project>_pf_varswitchcap.sql	Contains electrical characteristics of reactive power-regulated capacitors
~/sql/<project>_pf_pfswitchcap.sql	Contains electrical characteristics of power factor-regulated capacitors
~/sql/<project>_pf_tempswitchcap.sql	Contains electrical characteristics of temperature-regulated capacitors
~/sql/<project>_pf_hourly_load_profiles.sql	Contains load profiles for load classes (e.g., res, comm, ind). This data is only used if load profiles are configured to use load class profiles.

Workbook Maintenance and Update Process

The following process is to be followed both for first-time installation and for workbook updates in the Engineering data.

1. The user updates the engineering data for each device type in the corresponding worksheet.
2. From the TOOLS worksheet, the user executes the defined Excel Macro to generate an ASCII text file in a SQL format.
3. The user copies the SQL formatted ASCII text files to the appropriate Oracle Utilities Network Management System server(s) where the Oracle Utilities Network Management System data model is maintained and updated for the real-time operations environment. The files should be installed in the ~/sql directory on the server(s).
4. The user imports the data into the Oracle Utilities Network Management System database by either:
   - Executing ces_network_analysis.ces with the –clean option: This script will re-create and populate all the PowerFlow Oracle tables including the customer catalog ones.
   - Executing <project>_pf_setup.ces with the –clean option: This script will re-enter the configuration data for all customer catalog tables.
   - Executing ISQL.ces <file>. This lets the user import individual files and saves time since fewer files are processed.
   - Execute "product_pfpp_Tk.pl –startUI". This invokes the user interface of the Power Flow Post-processor program. Then execute the scripts from the Setup tab.

5. The Oracle Utilities Network Management System Power Flow Post-Processor reads the imported engineering data using the device attribute views, processes the data and builds the run-time model used by the DMS applications.

PowerFlow Post Processor

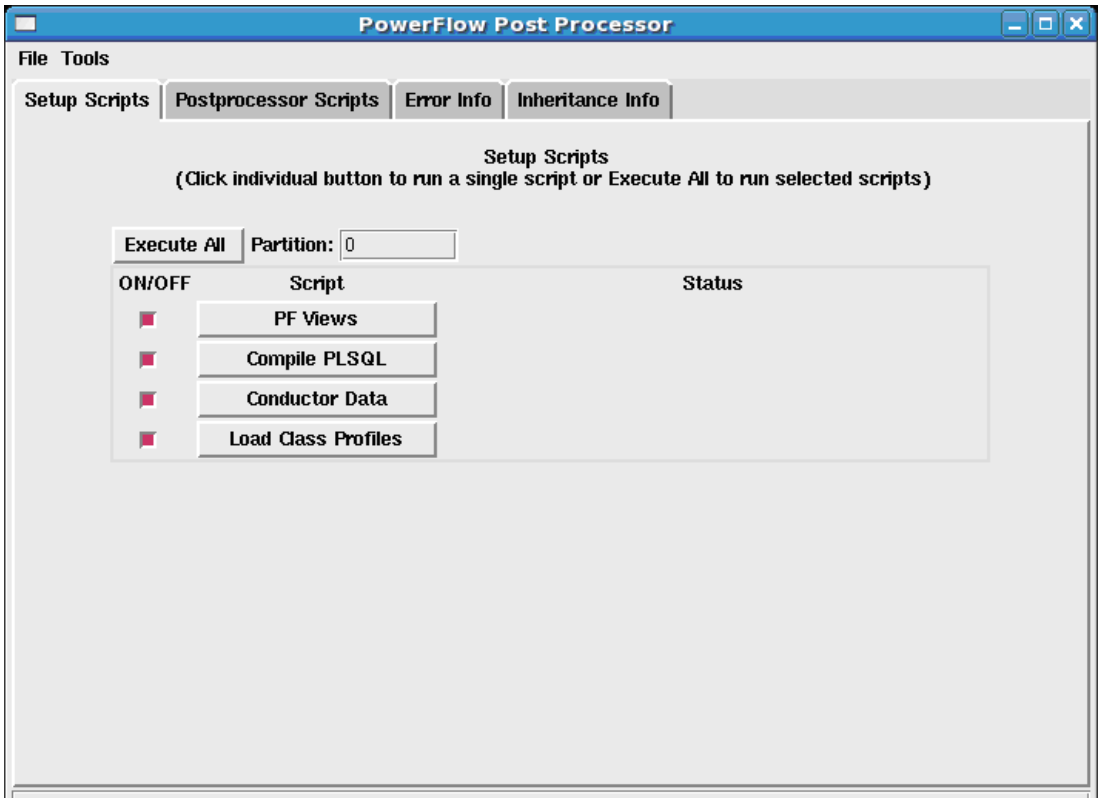
The Post Processor tool ensures that all data required for running the Oracle Utilities Network Management System power flow applications is correctly updated and loaded into the model database. It is designed to be run following both initial and incremental model build operations and provides a menu of operations (different scripts and stored procedures) that can be run to update different types of power flow data.

The Post Processor has a user interface which can be invoked with the following Unix command:

```
product_pfpp_Tk.pl -startUI
```

Much of the functionality in the postprocessor can also be run from the command line without bringing up the GUI. To see a list of these options, run `product_pfpp_Tk.pl` with no command line arguments.

The PowerFlow Post Processor window is shown in the following figure:



The PowerFlow Post Processor Tools menu displays the following options:

- **Check_DBService**: Enables user to check whether DBService is running
- **Check_PFService**: Enables user to check whether PFService is running
- **Start_PFService**: Enables user to start PFService if it is not running.
- **Check_DBI_Drivers**: Enables user to check whether the PerlTK DBI drivers are okay so that the postprocessor can be run successfully.

When you click on a Tools menu option, the status of the service displays at the bottom of the Post Processor window.

GUI Configuration

The Post Processor tool has a tabbed format.

The Setup Scripts and Post Processor Scripts tabs have buttons for executing various parts of the DMS installation and setup. These buttons are configured to execute either Unix shell scripts, PL/SQL stored procedures or load Oracle SQL files. The configuration for these actions is done in the database table pf_postprocessor_procs.

The configuration of the *pf_postprocessor_procs* table is done through the following SQL scripts:

- `product_pf_defaults.sql`
- `<project>_pf_defaults.sql`

Setup Scripts Tab

The Setup Scripts tab on the PowerFlow Post Processor window provides a GUI to execute a subset of the ces_powerflow.ces script. The actions on this tab are usually only performed in association with major changes to the model, such as after the initial model build. The actions shown on this display can be executed independently or as a group by using the **Execute All** button. As discussed, each button is configured to run a particular DMS script.

The following table describes the buttons and fields on the Setup Tab:

Field	Description
Partition	User-editable text box. The user may enter the non-zero number of the partition for which setup needs to be run. The default is 0, in which case all partitions are processed
PF Views	Button rebuilds the PowerFlow views. These views provide a flexible way to modify the source or default for certain PowerFlow attributes. This button is usually configured to invoke the *<project>_pf_views.sql* file. If this file is modified, executing this button will re-create the views in the Oracle database.
Compile PLSQL	Button recompiles the PFPostprocessor PL/SQL package. It invokes the *product_pf_plsql.ces* script. Note: This needs to be done whenever there is a change to the schema (e.g. PF Views are re-created) or when the package itself is updated.
Load Class Profiles	This command loads class profiles (e.g., residential, industrial, commercial profiles) using the `<project>_pf_hourly_load_profileloader_fromsql.ces` script, which loads the profiles stored in `<project>_pf_hourly_load_profiles.sql`.
ON/OFF	Toggle buttons enable the user to select the scripts that have to be run
Execute All	Button enables the user to run all selected scripts (selected by the *ON/OFF* toggle button) together
Status	Shows the start and end times of the selected script run

Post Processor Scripts Tab

The Post Processor Scripts tab functions similarly to the Setup tab. It provides a GUI to execute one or more update actions. The buttons on this display can be executed individually or as a group using the **Execute All** button.

To update catalog data, follow these steps:

1. Navigate to the Post Processor Scripts tab on the Post Processor screen.
2. Click on one or more script buttons corresponding to the device types that have had their catalog data updated through the Power Flow Engineering Data workbook or where load profiles have been updated.

Check the property pages or the Error Info tab to evaluate the correctness of the data populated into the DMS run-time tables by the post-processor scripts.

The following table describes the available buttons and fields on the Post Processor Scripts tab.

Field	Descriptions
Partition	User-editable text box. The user may enter the non-zero number of the partition for which setup needs to be run. The default is 0, in which case all partitions are processed
Sources	Invokes the *processSources* procedure in the PFPostprocessor package. This populates the DMS run-time tables with equivalent model data for source nodes in the network.
Line Catalogs	Invokes the *processLineCatalogs* procedure in the PFPostprocessor package. This processes and populates data for overhead lines and underground cables
Lines	Invokes the *processLines* procedure in the PFPostprocessor package. This updates the *network_components* table to correctly associate every network branch with a corresponding line/cable catalog
Line Limits	Invokes the *processLineLimits* procedure in the PFPostprocessor package. This processes ratings data for lines/cables
Loads	Invokes the *processLoads* procedure in the PFPostprocessor package. This populates the DMS run-time tables with equivalent model data for loads/supply nodes in the network
Switches	Invokes the *processSwitches* procedure in the PFPostprocessor package. This populates the DMS run-time tables with electrical characteristics data for switches
Transformers	Invokes the *processTransformers* procedure in the PFPostprocessor package. This populates the DMS run-time tables with electrical characteristics data for transformers
Transformer Limits	Invokes the *processTransformerLimits* procedure in the PFPostprocessor package. This processes ratings data for transformers
Capacitors	Invokes the *processCapacitors* procedure in the PFPostprocessor package. This populates the DMS run-time tables with electrical characteristics data for capacitors

Field	Descriptions
Load Trans Profiles	This action runs the `<project>_pf_hourly_load_profileloader_fromcsv.ces` script. It reads flat CSV files with transformer profile data from the `NMS_LOADPROFILE_DIR` directory and updates load profile data for affected loads. This procedure should be run whenever updated transformer profile data (either partial data or a full set of data) is available and will update any affected loads.
ON/OFF	Toggle buttons enable the user to select the scripts that have to be run
Execute All	*Execute All* enables the user to run all scripts (selected by the *ON/OFF* toggle button) together
Status	Shows the start and end times of the selected script run

Inheritance Info Tab

An important part of configuring the DMS applications is defining class inheritance. This display shows a scrollable inheritance tree for PowerFlow classes (actually, all classes that begin with "pf_"). Important classes are:

pf_lines – all children of this class are considered to be conductors (either lines or cables) by the DMS applications. They should have non-zero length and should have a valid catalog_id.

pf_auto_switch – all children of this class are considered candidates for switching by the Suggested Switching application.

Error Info Tab

Errors may occur at any stage in the transformation of customer/OMS data and subsequent loading into the DMS run-time tables. Typically errors arise due to data that's incomplete, inaccurate or duplicated. It would be useful for the user to be able to have a behind-the-scenes view into the running of the PFPostProcessor scripts in order to be aware of any data/process deficiencies if required. This is the purpose of the Error Info Tab.

PowerFlow Post Processor logs errors and warning in an Oracle table, pf_postprocessor_log. This GUI provides a convenient means of viewing the error logs and enables the user to take corrective action if required. Old errors are cleared each time the post-processor is run

Field	Description
Equipment Type	Drop-down list from which the user may select one or all device types of interest. This filters and presents error/warning messages pertaining only to the selected device type

Field	Description
Error Level	Drop-down list from which the user may select the severity level of the error/warning message for the selected device type. There are 4 configured severity levels in PFPostProcessor: 0: Error (Most Severe) 1: Warning 2: Informational message 3: Debug message **Note**: If the user selects 0, messages of all severity levels are displayed For more details on error messages refer to the section below
Clear Error Log	Button empties the PFPostprocessor error log table (*pf_postprocessor_log*). Typically used between clean runs of the postprocessor
Class	ODS Class of the device. Note: For error/warning messages pertaining to line catalogs, the class is displayed as 9999. Also for error messages captured directly from the PL/SQL system and where class information is not available, the class is displayed as 0
Index	ODS Index of the device. Note: For error messages captured directly from the PL/SQL system and where index information is not available, the index is displayed as 0
Severity	Displays the severity level of the message
Equipment	Displays the device type
Message	Displays the error/warning message
Time	Shows the time when the error/warning message was generated

To understand the error/warning messages generated by the Post Processor, it is necessary to have a general understanding of how the postprocessor PL/SQL procedures work. In the context of error message generation, this may be summarized into a 3-step process. For each device:

1. The script attempts to do a bulk insert of device attributes in to the relevant DMS run-time table.

   Note: 'Bulk insert' here implies one row - multiple columns, i.e one complete set of device attributes for one particular device

2. If step 1 is not successful, the script attempts to do a column-by-column insert of the device attribute data
3. If for any particular device attribute (column) step 2 fails, the script attempts to insert default data for that particular attribute.

During each of the above steps, the Post Processor generates information/warning/error messages and also captures error messages from the PL/SQL system. These messages are displayed on the screen.

The following table lists common error/warning message types generated by a run of Post Processor:

Message	Description
"Populating <device type> partition: <num>"	Level 2: Informational message indicating that the processing of devices of a particular type is about to commence
"trying data insert column-by-column"	Level 1: Clear indicator that the bulk insert described above has failed for the device of the displayed class and index
"inserted the handle.."	Level 1: Generated during the columnwise insert for the displayed device. Indicates that the device handle (actually, primary key attributes for the device type) have been inserted into the DMS run-time table
"all insert attempts failed.."	Level 0: Indicates that either: bulk insert, columnwise insert, and columnwise insert of default data have all failed for a particular device (a row of device attributes). Indicators as to which column(s) are failing may be seen from other Level 1 warning messages for that device. Or, bulk insert failed and then immediately after that, the insertion of primary key attributes for that device failed. This usually manifests as a constraint violation indicating duplicate data.
"<attribute name> failed with data = <value> where <device handle info>"	Level 1: Indicates that columnwise insert (with actual data) for the device has failed for the attribute mentioned. Also indicates that the postprocessor would now attempt to insert default data for this attribute
"<attribute name> failed with data = <value>"	Level 1: Similar to the earlier message
"<attribute name> failed with default data = <value> where <device handle info>"	Level 1: Indicates that columnwise insert with both, actual as well as default data for the device has failed for the attribute mentioned. This is a precursor to the Level 0 message "*all insert attempts failed..*"
"<attribute name> failed with default data = <value>"	Level 1: Similar to the earlier message
"<attribute name> could not be inserted"	Level 0: Indicates that the columnwise insert has failed and that the error could not be captured by the above 4 Level 1 warning messages
"constraint <constraint name> violated for <device id>"	Level 1: Generated either during bulk or columnwise insert of a particular attribute. Usually the constraint name gives a good indicator of the device attribute being processed. Similar SQL system error messages are captured and displayed as Level 1 or Level 0 messages. A Level 0 system message implies a catastrophic failure in the particular PL/SQL procedure
"ERROR IN SEQ IMP PROCESSING"	Level 0: Indicates catastrophic failure in the conductor sequence impedance processing part of the post-processor

Message	Description
"ERROR IN PHASE IMP PROCESSING"	Level 0: Indicates catastrophic failure in the conductor phase impedance processing part of the post-processor
"ERROR IN OH LINE PROCESSING"	Level 0: Indicates catastrophic failure in the overhead line construction processing part of the post-processor
"ERROR IN UG CABLE PROCESSING"	Level 0: Indicates catastrophic failure in the underground cable construction processing part of the post-processor
"ERROR IN UPDATING NETWORK_COMP"	Level 0: Indicates catastrophic failure in the procedure that updates the *network_components* table with information associating network branches with specific line/cable catalog IDs
"getColumnData FAILED : <system error msg> "	Level 0: Indicates that the procedure that fetches default data for the attributes of a particular device type has had a catastrophic failure

Power Flow Engineering Data Maintenance

To refresh the DMS applications with the updated data, you must restart PFService. PFService may be restarted at the Unix command prompt with:

```
Action any.PFService restart
```

The users are also able to update some type of DMS data with PFService running:

- To re-initialize system source data with PFService running, execute:

  ```
  Action any.PFService updatesystemsources
  ```

- To re-initialize reload system load data with PFService running, execute

  ```
  Action any.PFService updatesystemloads
  ```

- To re-initialize system capacitor data with PFService running, execute:

  ```
  Action any.PFService updatesystemshunts
  ```

- To re-initialize system transformer data with PFService running, execute

  ```
  Action any.PFService updatesystemxfmrs
  ```

- To re-forecast Feeder Load Management data with new equipment ratings after a model build, execute:

  ```
  Action any.PFService flm reforecast
  ```

Chapter 9

Database Maintenance

As general maintenance, you should establish a schedule to analyze tables, defragment your database, and purge historical/unnecessary data (then re-analyze the tables). You should also set up a schedule to backup your database and archive the backups.

This chapter describes all of these processes as well as the process of reconciling differences in database requirements when you upgrade your model to a new release of Oracle Utilities Network Management System.

It includes the following topics:

- **Oracle Configuration**
- **Purging Historical Data**
- **Applying Migrations**

Oracle Configuration

The following database settings are suggested for at least a minimum level of performance for an Oracle database. Any of these suggestions can be disregarded if an experienced Oracle DBA determines that other settings may offer better overall system performance. However, if any changes are made to any suggested parameters, performance of the system may be affected.

Indexes

Indexes should not be placed on the same physical disk as the data resides. If disk striping is being used then this requirement is not as critical, and may be ignored if enough disks are being employed.

Generating Statistics

As mentioned in a previous section of this chapter, tables should be analyzed periodically. The frequency can be determined by an experienced DBA, but it is suggested that this be done at least weekly. This ensures that the Oracle statistics will be kept up to date for all of the database tables.

Oracle Parameter settings

The Oracle Utilities Network Management System requires the Oracle RDBMS has enough memory to support the expected end user performance. Oracle RDBMS can automatically manage shared memory, but it is suggested that the following parameter be set to define the total amount of memory that is available.

- sga_target – This parameter should be set to at least 1G and could be set higher depending on the size of the database

Make Tablespaces Locally Managed

Dictionary managed tablespaces are more expensive on performance. It is suggested that the Oracle Utilities Network Management System tablespaces be setup as locally managed.

Block Size

If possible, the disk block size of the database should be a minimum of16K, but could be set larger on recommendations from an experienced DBA.

Purging Historical Data

As tables continue to grow, many of their rows become "inactive." The "inactive" data could be historical outage data (completed and/or cancelled outages) or old model build data that is no longer needed.

You should develop a plan to purge the extraneous data from the operational tablespaces (back it up or delete it) on a regular basis. After the data is purged, re-analyze these tables. This process requires proper planning and design because you do not want to lose important information required for reporting or troubleshooting.

Guidelines and Considerations

When developing your plan, it is helpful to understand how the purging process works.

From the facilities tables in the operations database, the usage statement is:

```
mb_purge.ces [-runfile] [-rows <integer>]
             [-days <integer> | -date <MMDDYY> ]
             [-purge] [-analyze] [-table <table_name>]
             [-debug] [-showme]
```

This script can purge obsolete data from model build tables or update Oracle statistics for the tables. The age of data is determined by the DEATH column.

Parameters are described below:

- If the **–purge** option is provided, an SQL file called mb_purge.sql is generated. When this SQL file is executed (by using the **-runfile** option described below), all obsolete rows are deleted.
- If the **–runfile** option is provided, the mb_purge.sql file is executed. If this parameter is not specified, no tables are purged.
- Use the **–rows** parameter to limit how much data is deleted with each command, to prevent filling up the rollback segment.
- Use either of the following options to specify the range of data:
  - **-days**: All rows older than this many days are removed.
  - **-date**: All rows older than the given date are removed.

- If **–analyz**e is provided, the Oracle statistics are updated.
- If a table is provided to the script in **-table**, then the script is only executed against that table. Otherwise, all tables that have a DEATH_PATCH column are processed.
- The **–debug** option prints out extra information for debugging purposes.
- If the **–showme** option is used, the script just prints out the number of rows that could be purged from each eligible table.

Compatibility

An Oracle Utilities Network Management System schema is not backward compatible with Oracle Utilities Network Management System applications. Schema changes occur and are modified as the code and database move forward in time.

For example, it is unlikely that a database which has been migrated or built at version 1.7.10 code level will work with version 1.8.0 code level. However, data models are forward compatible, because Oracle Utilities Network Management System applications can migrate the database forward, making the necessary changes.

Thus, when backing up the database, you should note the Oracle Utilities Network Management System release level that was last operating against the database dump. That way, if there are other systems with older code, the data model is not imported into those systems and problems are not introduced.

Software

The Oracle Utilities Network Management System software is likely to be the most static data on the system. It should only be changing with upgrades. The need for software backup is generally low if the software is installed on several machines locally, but a weekly backup may be needed if there are maintenance scripts and SQL files being updated.

Map Files

Map files are replicated on a number of machines throughout the network, but they will change frequently. Data model files should be backed up once per week at minimum or nightly for frequently changing files.

Applying Migrations

The Apply Migrations process migrates the model of an older Oracle Utilities Network Management System release to that of a new software version. Based on a release level identifier, the migration process determines the differences between the current model and that of a new release. After the installation of a new release of software, and the loading of a copy of your existing production database, you will need to do the following:

- Execute the $CES_HOME/bin/ces_setup.ces script

This script will call another script called ces_apply_migrations.ces, which determines the differences between the release level of the software and the model database. This script then determines the required and optional migrations by accounting for differences in the release database requirements.

Manual Migrations

If a manual migration is required, the ces_setup.ces script will stop at that point and alert the user of the required manual migration. When this occurs, please see the corresponding manual migration file in the $NMS_HOME/migration/manual directory for details on what is required for this migration. The files in this directory are named XXXXX.txt where XXXXX = the bug or PR number.

The $NMS_CONFIG/migration/data/<project>_config_ready.dat file serves as a "sign-off" document for the Oracle Utilities Network Management System project team. As you determine that a manual migration has been completed (or is not needed for your system), you must add the corresponding Bug numbers to the $NMS_CONFIG/migration/data/<project>_config_ready.dat file putting one Bug number per line. Once you have edited this file, you can run $CES_HOME/bin/nms-install-config to copy it to the $NMS_HOME/migration/data directory or manually copy the file there if you prefer. This signals the migration script that this particular manual migration has been completed. Once the file has been properly copied to $NMS_HOME/migration/data, you need to rerun the ces_setup.ces script. Continue this process until all manual and automated migrations are executed.

Command Line Options

The ces_apply_migrations.ces script can be initiated directly from the command line in order to view some of the things that it will be doing when started from the ces_setup.ces script. The following table describes all of the command line options for this script.

Option	Description
-debug	Displays debug information.
-showme	List all processes that would be executed, but do not actually execute any programs or SQL files.
-needConfig	Displays a list of migrations that are required by a project.
-listMigrations	Displays a list of migrations needed without applying them.

Note: The `ces_apply_migrations` script should not be run without any command-line arguments since that would cause the migrations to actually be executed. The command-line arguments listed above are to be used with the script so that it can be run in a "show only" mode but won't actually do the migrations.

Installing Migration Files

The data files that are required for the migration process are installed in the $NMS_HOME/migration/data directory. After making changes to the project-specific $NMS_CONFIG/migration/data/<project>_config_ready.dat file and an optional special $NMS_CONFIG/migration/data/<project>_migration.dat file, run nms-install-config script to install them into the $NMS_HOME/migration/data directory.

The Migration Process

The ces_apply_migrations.ces script determines the database differences by comparing the database release level in the ces_parameters table with the software release levels found in the software_release_id.dat and software_release_levels.dat files. Based on these differences, it will create a list containing all of the necessary migrations.

The migration process, or ces_apply_migrations.ces, finds the necessary migrations in the $NMS_HOME/migration/data/pr_migration.dat file and the $NMS_HOME/migration/data/product_pr_migration.dat file, which contains the list of PRs, releases, patch levels, and configuration types. If there are project-specific migrations, then a optional <project>_pr_migration.dat file is also used.

The pr_migration.dat files resemble the following example:

PR	Release	Patch	Required	Config Required	Script Exists	ConfigType
----	-------	-----	--------	------	------	----------
19254	5.5	3	Y	Y	Y	config_sql
19831	6.0	3	Y	N	Y	schema_sql

The following table describes the pr_migration.dat file columns.

Column	Description
PR	Bug or Problem Report (PR) number for the migration.
Release	Migration release level, two numbers not including the first digit. For example, release 1.8.1 would be just 8.1 in this field.
Patch	Migration patch level. If the release if 1.8.1.2, then the Patch would be 2.
Required	Whether or not this migration is required for the system to function properly. If set to Y, all projects would be forced to execute this migration when encountered. A value of N means that the migration is optional, and it would be skipped for any projects that do not list it within their <proj>_config_ready.dat file.
Config Required	Whether or not configuration is required by a project for the system to function properly. This value is set to Y whenever a change is made that requires configuration work. For instance, if a new required column is added to a configuration table, the population of this new column properly is the domain of the project engineer, not the developer. Setting this field to Y will flag to all project engineers that this migration requires their attention before the migration can be executed. The specific instructions for configuration migration must be documented in the PR's Migration section in gnats. Project engineers signify that the configuration has been examined and completed by added this migration PR to the <proj>_config_ready.dat file.
Script Exists	Indicates whether a script exists for the migration. For example, if a script exists for PR 19254, then there is a script pr19254_migration.ces that performs the migration. Not all migrations involve explicit scripts. As an example, a configuration table change would normally not require a migration. However, if it is important that a new configuration column be properly populated, this must be flagged for project engineers. This is done by adding the PR to pr_migration.dat, setting Config Required to Y and Script Exists to N. Even though there is no migration script, the migration process will not proceed until the project engineer has signified that the configuration is complete by adding the PR to the <proj>_config_ready.dat file.

Column	Description
Config Type	Describes the type of configuration change. Valid values are: • config_sql - A configuration SQL file has changed. • schema_sql - A schema SQL file has changed. • retain_sql - A retain SQL file has changed. • core_sql - A core (required) data SQL file has changed. • data - Model (facilities) data is being migrated. • app_defaults - New or obsolete application default options. • map_rebuild - The migration script will regenerate map files. • metafile_rebuild - The script will regenerate all map metafiles. • service_restart - Services must be restarted. • environment_restart - All user environments must be restarted.

Correcting Warnings and Errors

The table below shows the corrections for some possible errors you might receive when running the ces_apply_migrations.ces script.

Warning	Remedy
WARNING THE FOLLOWING MIGRATIONS NEED CONFIGURATION PR_NUMBER RELEASE_PATCH	This warning is displayed when migrations requiring manual changes are found. To determine the necessary changes, refer to the corresponding file in the $NMS_HOME/migration/manual directory. After making the manual changes, add the PR number to the $NMS_CONFIG/migration/<project>_config_ready.dat file.
DATABASE RELEASE LEVEL IS GREATER THAN SOURCE RELEASE LEVEL MIGRATING BACKWARDS NOT SUPPORTED	This error indicates that the schema level of the database is greater than the runtime executables that are being used. You can return to a prior release if you execute the ces_setup.ces script with the -clean command line option and perform a model build. You should not return to a prior release without running a ces_setup.ces -clean and a model build, for there may be unresolved problems that could cause system instability.

Chapter 10

Troubleshooting and Support

On rare occasion, you may experience problems with your Oracle Utilities Network Management System. There are a number of tools available to help you identify problems, including log files, core files, and Oracle Support.

This chapter includes the following topics:

- **Log Files**
- **Core Files**
- **Locating Information to Troubleshoot an Issue**
- **Contacting Oracle Support**

Log Files

- The log files are the best tools for tracking down the source of a problem. Very seldom does something crash or a tool behave strangely without an entry being logged. There are many different types of log files created by the application software or other 3rd party products. The sections below describe the locations and naming conventions for these logs. Before reporting an issue to Oracle Customer Support, please review the log files for critical information that may help support solve your problem.

Oracle Utilities Network Management System Log Files

Application log files are located in the directory specified by the CES_LOG_DIR environment variable located in the ~/.nmsrc file.

Note: CES_LOG_DIR = $NMS_HOME/log by default.

- If the UNIX server is running Oracle Utilities Network Management services, there should be one log file in this directory for each actively running service.
- If the UNIX server is running client environments, there should be a single log for all of the client applications that are being run for an individual client user.
- After a process has been stopped and restarted, the old log file for that particular server is moved to the old_log subdirectory within the log directory.

Oracle Utilities Network Management System Log File Naming Conventions

Within the log directory, the following naming conventions apply:

- There is one log file for each Service actively executing on the server. Service logs are named <Service Name>.<date>.<time>.log. Example log files would be:

  ```
  DBService.092598.1117.log
  DDService.092598.1118.log
  MTService.092598.1118.log
  ODService.092598.1118.log
  JMService.092598.1118.log
  TCDBService.092598.1118.log
  ```

- Client application logs are named ces_<User Type>.ces.<date>.<time>.log and are stored in a subdirectory that corresponds to the username used to login to the environment (e.g. if you logged into the Unix environment as "oms1," the applications logs would be stored at $CES_LOG_DIR/oms1). Example log files would be:

  ```
  ces_admin.ces.033109.1135.log
  ces_dispatch.ces.033109.1135.log
  ces_crewman.ces.033109.1135.log
  ```

Trimming and Archiving Application Oracle Utilities Network Management System Log Files

As log files grow, they need to be trimmed to save disk space and periodically backed up and archived. This may be accomplished with any text editor.

When determining the maximum size and content of log files, consider your company's needs:

- If accounting files need to be kept for an audit, a larger log file is justifiable. Backups of those files might even be in order.
- If the need for security is high, keep log entries from several weeks ago.

Issues like these should be carefully assessed, and you should develop a policy around your company's specific needs.

Java Application Server Log Files

The location of the Java application server log files depends on the type of the Java application server being used.

Java Application Server	Log file directory
Oracle WebLogic Server	BEA_HOME/user_projects/DOMAIN_NAME
JBoss Application Server	JBOSS_HOME/server/default/log

BEA_HOME - Oracle WebLogic Server installation directory

DOMAIN_NAME - WebLogic domain name used for Oracle Utilities Network Management System

SERVER_NAME - WebLogic server name used for Oracle Utilities Network Management System

JBOSS_HOME - JBoss Application Server installation directory

Java Client Application Logs

Java client applications that execute on a user's desktop do not generate log files. To obtain their output (error messages, exceptions and debug information) the Windows Java console can be used but it must first be enabled.

Use following steps to enable the Java console in Windows:

1. Open the Control Panel (Start -> Settings -> Control Panel).
2. Open the Java Control Panel by double-clicking on the Java icon in the Control Panel.
3. Select the Advanced tab of the Java Control Panel.
4. Set Java control parameter to 'Show console' (Java console will be started maximized) or 'Hide console' (Java console will be started minimized).
5. Press **OK.**

Isis Log Files

There are two types of Isis log files:

- A startup log, which logs everything before protos is completely started, should it exit for some reason. The isisboot program starts protos using the nohup command, which makes protos immune to hang-ups, like exiting the terminal after starting Isis. The startup log is called isis.log and can be found in `$NMS_HOME/oms/run_isis`.
- The protos log, which contains information on running the protos process. This file is site-specific, and the name is based on the site number and port number of the machine on which protos is running. The log for the protos process can be found in `$NMS_HOME/oms/run_isis/<site #>.logdir.<port #>`.

Oracle RDBMS Log Files

Many times, there is an error in an Application log file that points to some sort of database problem. DBService may log that at a certain time the database was unavailable to answer queries. Look in the database logs to find the answer. These logs can alert you to problems with the RDBMS configuration, software, and operations. Refer to the Oracle RDBMS documentation for locations and instructions for viewing Oracle RDBMS logs.

Operating System Log Files

Another place to look for problems is in the operating system logs. Refer to the operating system specific documentation for locations and instructions for viewing operating system logs.

Core Files

On UNIX, if a process has either committed an error or over-taxed the system resources, the O/S will kill it rather than letting it take down the operating system. When this happens, the operating system dumps the contents of the memory occupied by the process into a file named “core.” These files can sometimes be analyzed to better understand the reason for the failure.

Normally, you should question the production of a core file to see if there are any extraneous reasons why the O/S is dumping a process. If you do not find anything, retrieve the core file and analyze it.

See **Core File Naming Configuration** on page 3-6 for OS specific information about core file naming.

Searching for Core Files

To search for core files, complete these steps:

1. Search for core files with the find command:

   ```
   $ find . -name core* -exec ls -l {} \;
   ```

   Expected result:

   ```
   -rw------- 1 ces users 32216692 Oct 15 16:05 ./core
   ```

   This executes an "ls -l" on any files found in the tree starting from the current working directory.

   If a service cores, the core file can be found in the ~/$CES_LOG_DIR/run.<service> directory.

2. Type:

   ```
   $ file ./core to determine where a core file came from.
   ```

   Example result from an AIX server:

   ```
   core: AIX core file fulldump 64-bit, JMService - received SIGBUS
   ```

   The core file referenced above is the result of a JMService core dump. The output gives:

   - the file name (which is always "core"),
   - which program/process the file came from (JMService), and
   - optionally, the message that the program received from the OS (SIGBUS).

3. Use the strings command to get some more information out of the file, if possible. Type:

   ```
   $ strings core | head
   ```

   Sometimes the messages returned, such as "Out of memory" or "I/O error," give an idea of what might have happened.

Locating Information to Troubleshoot an Issue

A good first diagnosis is to run the UNIX "top" command or equivalent. This will display information such as what processes are running, current memory usage, and free memory.

There are several logs that are useful for troubleshooting. These include environment logs, service logs, SMService logs, and PID logs. The directory specified by CES_LOG_DIR environment variable provides information for where the logs are located.

The logs generated from a user logging in to Oracle Utilities Network Management System Operator's Workspace are referred to as environment logs. The log files will be in the UNIX user directory or under the Unix user's name in the CES_LOG_DIR.

Customer Support will often ask about service logs. Looking for DBService errors is a common starting place in determining if the problem is a database issue or a services issue. DBService errors can appear in DBService, TCDBService, and MBDBService depending upon which service is having a problem interacting with the database.

If a particular service cores, Customer Support will want to know if the service has any error messages in the log file right before it went down. The most relevant portion of the log is the text concerning what happened right before the dump. Often, there are important messages explaining why the service exited.

Another key service log is the SMService log. This log records if SMService attempts to restart other services.

PID logs are files with an integer value suffixed by .log. When they are generated, they also create a <pid>.out file. The .out file is unnecessary and can be removed. <pid> logs are generated one of two ways.

- **cmd snapshot command**. This will create <pid> logs for all Isis processes currently running, whether they are services or tools. They appear in the following locations: services will appear in the ~/CES_LOG_DIR/run.<service> directory of the user that starts services. Tools will appear in the CES_HOME directory for a tool started by the ceslogin. If a tool is started from the command line, it will appear in the directory where the tool was started.
- **kill -usr2 <pid>**. This will NOT actually kill the tool. It will create a <pid> log for that one PID, however. It will not create logs for other processes under this method.

  Note: You can do this multiple times, and the logs will append additional dumps into the same log file as long as the process continues to run. It will not remove or replace logs upon additional snapshots of the same process. Customer Service recommends that these logs be cleaned up upon the end of investigating an issue.

There is also a feature that uses the Action command to start a new log file without stopping anything. This can be very useful in isolating a portion of the log file when recreating a problem. The command is:

```
Action any.<service> relog
```

This can also be executed on a tool. When doing this with a tool, e.g., Work Agenda, you first export the display to the screen the Work Agenda is being displayed to. Then use the command Action this.workAgenda relog. This will cause the messages from the Work Agenda to go to a new Work Agenda log instead of the environment log, where the messages for all of the other tools are going.

The Action command can also be used to turn debug on and off for services or tools. This can also be used with the relog feature to isolate the debug portion.

The command "Action any.<service> debug 1" will turn debug on, while the command "Action any.<service> debug 0" will turn debug off.

Core files are other useful tools for troubleshooting. Core files are located in the CES_LOG_DIR/run.<service> directory in the username that started services, or in the directory where a tool was started (usually the home directory of the user). After performing a "kill –usr2" on a hung process, following it up with a "kill –abrt <pid>" is often useful. This will cause the process to dump core and the process will be dead. One thing to remember is to always "kill –usr2" before "kill –abrt", because the –abrt option terminates the process.

The command "file core" will identify which process generated the core. Later core files can overwrite earlier core files. Renaming the core file to something like core.<process> can prevent this.

When a tool or service cores, the investigation is helped by sending the stack trace in the incident report. A stack trace can be generated using the dbx tool. The syntax is as follows:

```
dbx <path to binary directory> <path to corefile>
```

For example:

```
dbx ~/bin/JMService ~/run.JMService/core.
```

Press the space bar until you get a prompt and then type the word "where". Then include the results of the "where" command when you report the incident.

monitor_ps_sizes.ces

The monitor_ps_sizes.ces script monitors the size of processes to identify potential leaks. It performs periodic snapshots of all running processes and warns the user of any processes that have grown greater than the specified size. It supports the following command-line options:

Option	Description
-n <program names>	A comma-separated list of program names to monitor
-l <line number>	The line number that specifies the stable size in the process-size log file. Default: 3 (line numbers begin counting with 1)
-l <line number>	The line number that specifies the stable size in the process-size log file. Default: 3 (line numbers begin counting with 1)
-p <number>	The number of seconds to wait between snapshots. Default: 3600 (seconds)
-g <number>	The growth factor that triggers a report. Default: 1.75 (floating point numbers greater than 1 are valid)
-R <number>	The minimum process size that can be reported. Default: 5000 (units reported by ps)
-G <number>	A warning about a process is guaranteed to be generated if the process exceeds this size. Default: 40000 (units reported by ps)
-P <number>	The minimum number of seconds to wait between warnings. Default: 0 (seconds)
-O <number>	The maximum number of seconds to retain log files.Default: 172800 (seconds) if 0, old log files are not erased.
-u <email names>	A comma-separated list of users to email when there are processes warnings. Default: no email sent.

Option	Description
-s <email subject line>	The subject line to use to title email warnings about processes that are too big. Default: “process size warning for prod_model”
-a <command>	Command to perform on process when generating a warning. You can pass the program's name and/or PID via #PID# and #PROGRAM#
-A	Log the command's output

For example, to monitor JMService and MTService for user 'nms' when either gets larger than 500 meg or grows by 10%, use:

```
monitor_ps_sizes.ces -n MTService,JMService -f nms -p 30 -R 500000 -g 1.1
```

Contacting Oracle Support

For support please contact Oracle Support at:

```
http://www.oracle.com/support/index.html
```

Chapter 11

Setting Up Oracle Utilities Business Intelligence

This chapter describes how to set up the Oracle Utilities Business Intelligence for the Oracle Utilities Network Management System. It includes the following topics:

- **Installing Business Intelligence**
- **Installing Oracle Utilities Network Management System Business Intelligence Extractors**
- **Running Oracle Utilities Network Management System Business Intelligence Extractors**
- **Migrating from Performance Mart to Oracle Utilities Business Intelligence**

Installing Business Intelligence

Installation of the Business Intelligence component is covered in a separate installation guide that comes with the Business Intelligence Media Pack download. For the purposes of Oracle Utilities Network Management System integration, however, you must download the latest Business Intelligence patches from Metalink, as well as service pack 2 for Oracle/SPL Framework v2.0.10.

> **Note**: If you are upgrading from a previous PerformanceMart data warehouse, please reference the **Migrating from Performance Mart to Oracle Utilities Business Intelligence** on page 11-4 for details on the upgrade process.

Oracle Utilities Business Intelligence must be properly installed before you can perform the remaining procedures described in this chapter.

Installing Oracle Utilities Network Management System Business Intelligence Extractors

If extracting from separate Oracle Utilities Network Management System environments into a common BI environment, export the optional CES_BI_DATA_SOURCE environment variable to distinguish between them. The default setting of this is 4. See **Migrating from Performance Mart to Oracle Utilities Business Intelligence** on page 11-4 for more details.

To install the business intelligence extractors, run the install_business_intelligence script. Once this script has been run, use the refresh_business_intelligence script for any subsequent configuration and schema changes. This script generates a log file, create_bi_extractors.log, which lists any errors.

Running Oracle Utilities Network Management System Business Intelligence Extractors

This section explains the following scripts, which should be configured to run in scheduled cron jobs.

- **bi_common_extractor**
- **bi_event_extractor**
- **bi_customer_extractor**
- **bi_feeder_extractor**
- **bi_switch_extractor**
- **nrt_extractor**

bi_common_extractor

This extracts the model-related information like devices and control zones. This script is designed to be run daily, after model changes.

bi_event_extractor

This extracts completed outages and call information. This script is designed to be run daily.

bi_customer_extractor

This extracts customer information. This script is designed to be run daily, after customer data changes.

bi_feeder_extractor

This extracts feeder load information. This script is designed to be run hourly to report average hourly loads.

bi_switch_extractor

This extracts planned switching information. This script is designed to be run daily to report switching activity.

nrt_extractor

This extracts current outage, call, and storm information. This script is designed to be run 3 to 4 times an hour, throughout the day.

General Notes

These scripts create extract .DAT and .CTL files in the configured bi_extract_dir directory (recommended as $HOME/extract). These files will be read by the Business Intelligence import process.

Each script generates a log file named, for example, bi_common_extractor.log, which should list any errors.

To schedule the daily extracts (bi_common_extractor, bi_event_extractor, bi_switch_extractor and bi_event_extractor), they schedule them to run in the following order:

1. bi_event_extractor
2. bi_switch_extractor
3. bi_common_extractor
4. bi_customer_extractor

Note: The bi_feeder_extractor should not be run more frequently than once an hour, and the nrt_extractor can be scheduled to run every 15 minutes. The order that these two extractors run does not matter.

Importing Oracle Utilities Network Management System Extract Files

The extract files created by running the Oracle Utilities Network Management System Extractors must be moved to the directory specified in the EditFP.tcl script that is executed when Business Intelligence is installed. There are various mechanisms that a System Administrator can use to copy these files, including FTP scripts and Cross Mounting hard drives. However, Oracle does not provide any scripts to copy extract files, so a customer is responsible for putting these in place.

Once the extract files have been copied to the appropriate import directory, the Oracle Utilities Network Management System Process Flows described in the Oracle Utilities Network Management System Facts and Dimensions chapter of the Business Intelligence documentation need to be run to load the data contained in the files. The process flows corresponding to each extract program is documented in this chapter, and the import process and how to automate it is described in the Oracle Warehouse Builder chapter of the Business Intelligence documentation.

After importing the data, then the various Oracle Utilities Network Management System zones and portals that a customer has created can be opened or refreshed to view the Oracle Utilities Network Management System data in Business Intelligence.

For more information, use the Help system in Oracle Utilities Business Intelligence to view the Oracle Utilities Business Intelligence documentation.

Press the help button () located in the Business Intelligence Action Bar at the top of any Oracle Utilities Business Intelligence portal screen.

The next steps are a method to import from the extracted files using a function call in sqlplus.

1. Install the Function NMS_EXEC_WF_FNC to Execute Process Flows from SQLPLUS

   - For 10g, install the script nms_exec_wf_fnc_10.sql

   ```
   sqlplus birepownuser/birepownpasswd@birepown_instance
   < nms_exec_wf_fnc_10.sql > nms_exec_wf_fnc_10.sql.log
   ```

   - For 11g install the script nms_exec_wf_fnc_11.sql

   ```
   sqlplus birepownuser/birepownpasswd@birepown_instance
   < nms_exec_wf_fnc_11.sql > nms_exec_wf_fnc_11.sql.log
   ```

2. Make sure the following environment variables are set:

   - BIREPOWN_USER - BI Repository User
   - BIREPOWN_PASSWD - BI Repository Password
   - BIREPOWN_INSTANCE - SQL*Net connection to the BI Repository Database

3. Run the Import Into DWADM Schema from the Extracted Files. For the daily extracts, set the following scripts to run on schedule after the entire daily extract has run, in the following order:

   1. bi_customer_import - call this script after the bi_customer_extractor runs.
   2. bi_common_import - call this script after the bi_common_extractor runs.
   3. bi_switch_import - call this script after the bi_switch_extractor runs.
   4. bi_event_import - call this script after the bi_event_extractor runs.

For the other two extracts, set the import to run after the extract has taken place:

- bi_feeder_import - call this script after the bi_feeder_extractor runs.
- bi_nrt_import - call this script after the nrt_extractor runs.

Migrating from Performance Mart to Oracle Utilities Business Intelligence

This section provides an overview of the schema differences that you must be aware of when migrating from Performance Mart to Oracle Utilities Business Intelligence.

With version 1.9 of the Oracle Utilities Network Management System, the Performance Mart and Executive Dashboard modules have been replaced with Oracle Utilities Business Intelligence version 2.2.1. This section describes the differences between the two products, how to migrate an existing 1.7.10 Performance Mart database to Oracle Utilities Business Intelligence, and provides some guidelines on how to easily migrate existing reports to run against the Oracle Utilities Business Intelligence database.

For information not covered in this document, the Oracle Utilities Business Intelligence documentation is available for all supported releases, including the Oracle Utilities Network Management System Facts and Dimensions chapter that describes the schema and extraction processes that will be covered in this guide.

Schema Differences

The Oracle Utilities Business Intelligence database naming system is different than the Performance Mart schema, so every Oracle Utilities Network Management System object has a new name. Also, Oracle Utilities Business Intelligence utilizes a very strict star-schema approach, so many of the Command Centricity foreign key relationships do not exist.

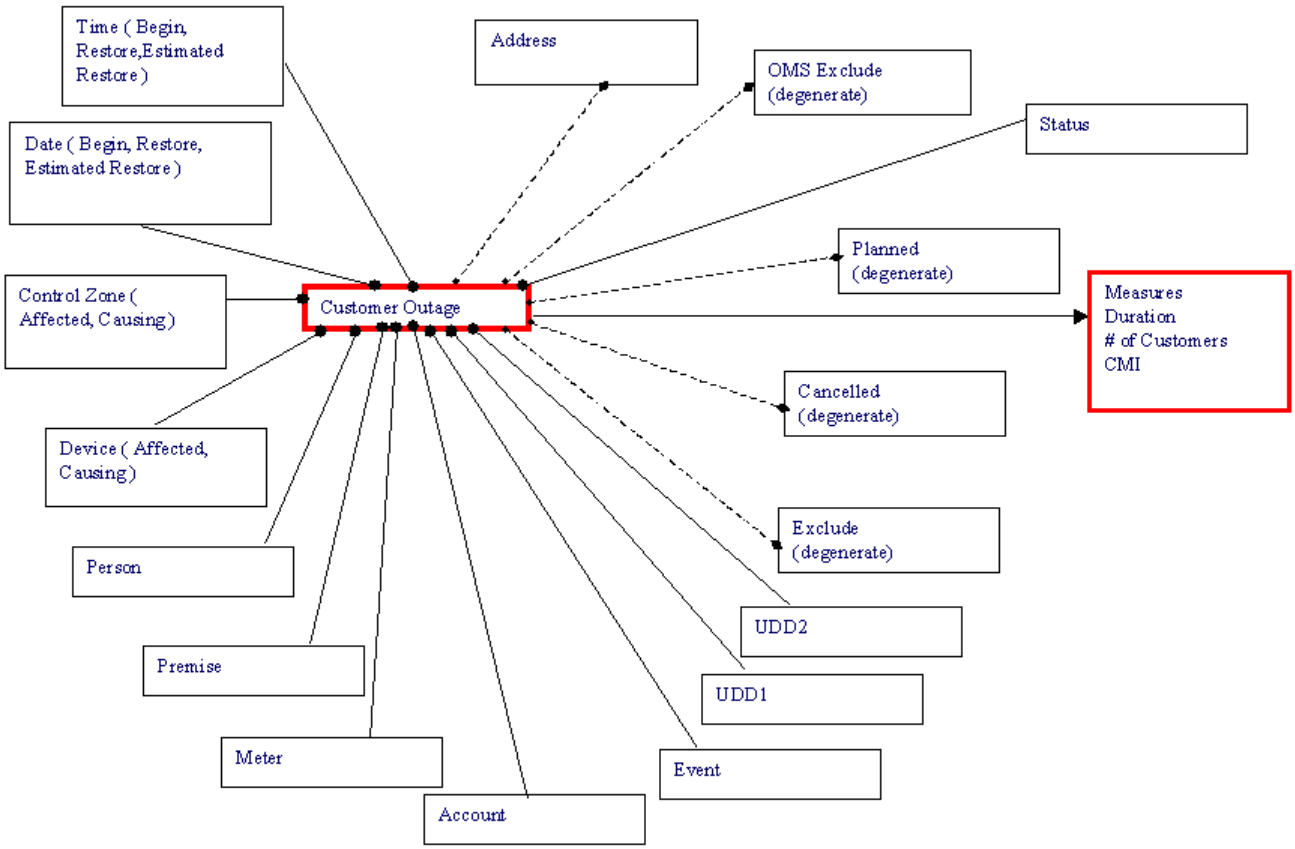
Performance Mart Schema

The Performance Mart schema is a hybrid star-schema/relational model that was convenient for use with Executive Dashboard, and detail trouble reporting.

Oracle Utilities Business Intelligence Schema

Unlike Performance Mart, the Oracle Utilities Business Intelligence Schema utilizes exclusively a Star schema representation. This enables the Oracle Utilities Business Intelligence framework to efficiently create queries against the database tables, and allows for an efficient generic load process.

The following figure shows the star schema diagram for the Customer Outage Fact. This fact corresponds with the SERVICE_POINT_SUPPLY_NODES table in the Performance Mart schema. If you compare the relationships here with the relationships above, you will notice a lot more foreign keys in this document, but nothing related more than one step away from the basic fact table.



The other major difference between Performance Mart and Oracle Utilities Business Intelligence is the use of generic field names in the tables. This is done to allow different customers to extract different fields without having to change the user interface or extractor code. For example, the Device information is stored in the DEVICE_DETAILS table in Performance Mart and in the CD_DEVICE table in Oracle Utilities Business Intelligence.

The following table lists the fields in each table and how they map from one to another

Device_Details	CD_Device
DV_CLS	SRC_DEVICE_CLS
DV_IDX	SRC_DEVICE_IDX
DV_CODE	DEVICE_NAME
DV_VOLTAGE	Unmapped
DV_TYPE	DEVICE_CLASS_CD
DV_DESC	DEVICE_CLASS_DESCR
DV_ACTIVE	Unmapped
	DEVICE_TYPE_CD
	DEVICE_TYPE_DESCR
	UDF1_CD
	UDF1_DESCR

Device_Details	CD_Device
	UDF2_CD
	UDF2_DESCR
	UDF3_CD
	UDF3_DESCR
	UDF4_CD
	UDF4_DESCR
	UDF5_CD
	UDF5_DESCR
	UDF6_CD
	UDF6_DESCR
	UDF7_CD
	UDF7_DESCR
	UDF8_CD
	UDF8_DESCR
	UDF9_CD
	UDF9_DESCR
	UDF10_CD
	UDF10_DESCR

Performance Mart to BI Mapping

The following tables show how the default migration routine will move data from Performance Mart tables to Oracle Utilities Business Intelligence tables. Performance Mart tables not listed here will not be migrated to Oracle Utilities Business Intelligence. Project configuration changes done during the actual migration can change how these columns are migrated, so this list should not be used as a definitive guide to a specific project implementation.

CU_SERVICE_LOCATION_DETAILS

The data in the CU_SERVICE_LOCATION_DETAILS table is migrated to three different BI tables: CD_ACCT, CD_ADDR and CD_PREM. The following table shows which fields go into which table. The CU_SERV_LOC_KEY is used as the primary key in each of these tables.

CU_SERVICE_LOCATION_DETAILS Field	BI Table Name	BI Field Name
cu_serv_loc_key	cd_acct	acct_key
cu_serv_account_number	cd_acct	src_acct_id
cu_serv_loc_id	cd_acct	acct_info

CU_SERVICE_LOCATION_DETAILS Field	BI Table Name	BI Field Name
record_birth_time	cd_addr	eff_start_dttm
record_death_time	cd_addr	eff_end_dttm
cu_serv_loc_key	cd_addr	addr_key
cu_serv_addr_1	cd_addr	addr_line1
cu_serv_addr_2	cd_addr	addr_line3
cu_serv_addr_3	cd_addr	addr_line4
cu_serv_city	cd_addr	udf1_cd, udf1_descr
cu_serv_postcode_1	cd_addr	udf3_cd
cu_serv_postcode_1 || cu_serv_postcode_2	cd_addr	udf3_descr
cu_serv_state	cd_addr	udf4_cd, udf4_descr
cu_serv_loc_id	cd_addr	src_addr_id
record_birth_time	cd_addr	eff_start_dttm
record_death_time	cd_addr	eff_end_dttm
cu_serv_loc_key	cd_prem	prem_key
cu_serv_loc_id	cd_prem	src_prem_id
cu_serv_type	cd_prem	udf2_cd, udf2_descr
cu_serv_life_support	cd_prem	udf3_cd, udf3_descr
cu_serv_c_priority	cd_prem	udf6_cd, udf6_descr
cu_serv_d_priority	cd_prem	udf7_cd, udf7_descr
cu_serv_k_priority	cd_prem	udf8_cd, udf8_descr
record_birth_time	cd_addr	eff_start_dttm
record_death_time	cd_addr	eff_end_dttm

CU_CUSTOMER_DETAILS

The data in the CU_CUSTOMER_DETAILS table is migrated to the CD_PER table in BI. The CU_CUST_KEY is used as the primary key in this table.

CU_CUSTOMER_DETAILS Field	BI Table Name	BI Field Name
cu_cust_key	cd_per	per_key
cu_cust_id	cd_per	src_per_id
cu_cust_name	cd_per	per_name, per_info
cu_cust_home_ac || cu_cust_home_phone	cd_per	per_phone_nbr
record_birth_time	cd_per	eff_start_dttm

CU_CUSTOMER_DETAILS Field	BI Table Name	BI Field Name
record_death_time	cd_per	eff_end_dttm

CU_METER_DETAILS

The data in the CU_METER_DETAILS table is migrated to the CD_METER table in BI. The CU_METER_KEY is used as the primary key in this table.

CU_METER_DETAILS Field	BI Table Name	BI Field Name
cu_meter_key	cd_meter	meter_key
cu_meter_id	cd_meter	src_meter_id, meter_info
record_birth_time	cd_meter	eff_start_dttm
record_death_time	cd_meter	eff_end_dttm

REPORTING_ELEMENTS - Cities

The data in the REPORTING_ELEMENTS table where the RE_TYPE = 'CITY' is migrated to the CD_CITY table in BI. The RE_KEY is used as the primary key in this table

REPORTING_ELEMENTS Field	BI Table Name	BI Field Name
re_key	cd_city	city_key
substr(re_name, 1, instr(re_name, ','))	cd_city	src_city
substr(re_name, instr(re_name, ',') + 2)	cd_city	src_state
'United States of America'	cd_city	src_country
record_birth_time	cd_city	update_dttm

REPORTING_ELEMENTS/REPORTING_HIERARCHY - Control Zones

The data in the REPORTING_ELEMENTS table where the RE_TYPE = 'CIR' is joined to the REPORTING_HIERARCHY_V view and this data is migrated to the CD_CONTROL_ZONE table in BI. The RH_KEY in the REPORTING_HIERARCHY table is used as the primary key in this table.

Performance Mart Field	BI Table Name	BI Field Name
reporting_hierarhcy_v.rh_key	cd_ctrl_zone	ctrl_zone_key
reporting_elements.re_number	cd_ctrl_zone	src_ncg_id
reporting_elements.re_type	cd_ctrl_zone	hierarchy_type
reporting_elements.re_name	cd_ctrl_zone	ctrl_zone_name
reporting_elements_1.re_number	cd_ctrl_zone	uf1_cd

Performance Mart Field	BI Table Name	BI Field Name
reporting_hierarhcy_v.level1_name	cd_ctrl_zone	udf1_descr
reporting_elements_2.re_number	cd_ctrl_zone	udf2_cd
reporting_hierarhcy_v.level2_name	cd_ctrl_zone	udf2_descr
reporting_elements_3.re_number	cd_ctrl_zone	udf3_cd
reporting_hierarhcy_v.level3_name	cd_ctrl_zone	udf3_descr
reporting_elements_4.re_number	cd_ctrl_zone	udf4_cd
reporting_hierarhcy_v.level4_name	cd_ctrl_zone	udf4_descr
reporting_elements_5.re_number	cd_ctrl_zone	udf5_cd
reporting_hierarhcy_v.level5_name	cd_ctrl_zone	udf5_descr
reporting_elements_6.re_number	cd_ctrl_zone	udf6_cd
reporting_hierarhcy_v.level6_name	cd_ctrl_zone	udf6_descr
record_birth_time	cd_ctrl_zone	update_dttm
record_death_time	cd_meter	eff_end_dttm

CREW_DETAILS

The data in the CREW_DETAILS table is migrated to the CD_CREW table in BI. The CR_KEY is used as the primary key in this table.

CREW_DETAILS	BI Table Name	BI Field Name
cr_key	cd_crew	crew_key, src_crew_id
cr_crew_code	cd_crew	crew_cd
record_birth_time	cd_crew	eff_start_dttm
record_death_time	cd_crew	eff_end_dttm

DEVICE_DETAILS

The data in the DEVICE_DETAILS table is migrated to the CD_DEVICE table in BI. The DV_KEY is used as the primary key in this table. Also, during the population, data for the Device Type fields that is not in Performance Mart is queried from the CLASSES table in the Oracle Utilities Network Management System database and populated into BI. If historical data does not exist for a specific class in Oracle Utilities Network Management System anymore, then these fields will be left blank.

DEVICE_DETAILS Field	BI Table Name	BI Field Name
dv_key	cd_device	device_key
dv_cls	cd_device	src_device_cls
dv_idx	cd_device	src_device_idx

DEVICE_DETAILS Field	BI Table Name	BI Field Name
dv_code	cd_device	device_name
dv_type	cd_device	device_class_cd
dv_desc	cd_device	device_class_descr
classes.c_type	cd_device	device_type_cd, device_type_descr
record_birth_time	cd_device	eff_start_dttm
record_death_time	cd_device	eff_end_dttm

Oracle Utilities Network Management System Users

No data exists in Performance Mart for Oracle Utilities Network Management System Users, so during the migration process, the current records in the CES_USERS table will be migrated to the CD_USER table in BI. The primary key will be populated from the SPL_USER_SEQ.NEXTVAL sequence that is normally used by the BI load process.

CES_USERS Field	BI Table Name	BI Field Name
user_name	cd_user	user_cd
full_name	cd_user	user_descr
sysdate	cd_user	eff_start_dttm
31-DEC-4000	cd_user	eff_end_dttm

Event Statuses

No data exists in Performance Mart for Event Statuses, so during the migration process, the current records in the TE_STATUSES and TE_STATUS_GROUPS tables will be migrated to the CD_USER table in BI. The primary key will be populated from the TRANS_STATUS field in the TE_STATUSES table.

NMS Field	BI Table Name	BI Field Name
te_statuses.trans_status + 1	cd_event_status	event_status_key
te_statuses.trans_status	cd_event_status	src_status
te_status_groups.description	cd_event_status	event_status_cd
te_statuses.description	cd_event_status	event_status_descr
Sysdate	cd_event_status	update_dttm

EVENT_CALL_FACTS

The data in the EVENT_CALL_FACTS table is migrated to two different BI tables, one dimension and one fact: CD_CALL_INFO and CF_RST_CALL. The following table shows which fields go into which table. The ECF_KEY is used as the primary key in each of these tables. For the BI tables below that are not CD_CALL_INFO or CF_RST_CALL, the mapping is done by using the foreign key in the CF_RST_CALL table. For example, to get the

ECF_ACCOUNT_NUMBER, the CF_RST_CALL table would be joined to the CD_ACCT table by ACCT_KEY.

EVENT_CALL_FACTS Field	BI Table Name	BI Field Name
ecf_key	cd_call_info	call_info_key
ecf_incident_number	cd_call_info	src_incident_id
ecf_last_name	cd_call_info	caller_name
ecf_phone_number	cd_call_info	phone_nbr
ecf_complaint	cd_call_info	Complaint
ecf_operator_comment	cd_call_info	Comments
sysdate	cd_call_info	update_dttm
ecf_key	cf_rst_call	rst_call_key, call_info_key
ecf_incident_number	cf_rst_call	src_incident_id
e_key	cf_rst_call	event_key
ecf_account_number	cd_acct	src_acct_id
ecf_total_priority	cf_rst_call	priority_ind
ecf_called_time (Date)	cd_date	cal_dt
ecf_called_time (Time)	cd_time	src_time
ecf_user_name	cd_user	user_cd

EVENT_DETAILS

The data in the EVENT_DETAILS table is migrated to two different BI tables, one dimension and one fact: CD_EVENT and CF_RST_JOB. The EVENT_PICKLIST table is also joined to the EVENT_DETAILS table and data in this table is migrated to the CD_EVENT table. The following table shows which fields go into which table. The E_KEY is used as the primary key in each of these tables. For the BI tables below that are not either CD_EVENT or CF_RST_JOB, the mapping is done by using the foreign key in the CF_RST_JOB table. For example, to get the ECF_ACCOUNT_NUMBER, the CF_RST_CALL table would be joined to the CD_ACCT table by ACCT_KEY.

EVENT_DETAILS Field	BI Table Name	BI Field Name
e_key	cd_event	event_key
e_outage_number	cd_event	src_nbr
e_event_idx	cd_event	event_nbr
e_ops_exclude_reason	cd_event	exclude_reason
e_operator_comment	cd_event	operator_comment

EVENT_DETAILS Field	BI Table Name	BI Field Name
e_valid_state_key	cd_event	event_state_descr
e_event_status	cd_event	event_state_cd
e_street_address || ',' || e_city_state	cd_event	first_call_addr
event_picklist.remedy_om	cd_event	remedy_cd
e_trouble_code	cd_event	trouble_cd_list
e_outage_cause_selection1	cd_event	udf1_cd, udf1_descr
e_outage_cause_selection2	cd_event	udf2_cd, udf2_descr
e_outage_cause_selection3	cd_event	udf3_cd, udf3_descr
e_outage_cause_selection4	cd_event	udf4_cd, udf4_descr
e_outage_cause_selection5	cd_event	udf5_cd, udf5_descr
e_outage_cause_selection6	cd_event	udf6_cd, udf6_descr
e_outage_cause_selection7	cd_event	udf7_cd, udf7_descr
e_outage_cause_selection8	cd_event	udf8_cd, udf8_descr
e_outage_cause	cd_event	udf9_cd, udf9_descr
e_outage_cause_selection	cd_event	udf10_cd, udf10_descr
e_key	cf_rst_job	rst_job_key, event_key
e_outage_number	cf_rst_job	src_job_nbr
e_status + 1	cf_rst_job	event_status_key
e_begin_time	cf_rst_job	begin_dttm
e_completion_time	cf_rst_job	rst_dttm
e_est_restore_time (est_rst_date_key)	cd_date	cal_dt
e_est_restore_time (est_rst_time_key)	cd_time	src_time
e_ops_exclude_flag	cf_rst_job	oms_exclude_ind
e_cancel_flag	cf_rst_job	cancelled_ind
re_key	cf_rst_job	ctrl_zone_key
dv_key	cf_rst_job	device_key
e_crew_id1	cd_crew	src_crew_id
e_est_num_cust	cf_rst_job	udm1

Customer Outage

Customer Outage information is stored in three key tables in Performance Mart: SERVICE_POINT_SUPPLY_NODES, EVENT_SUPPLY_NODES and EVENT_DETAILS. Data from each of these tables as well as Customer Keys in the CUSTOMER_SERVICE_POINTS table will be migrated to the CF_CUST_RST_OUTG table

in BI. The primary key will be populated from the SPL_CUST_RST_OUTG_SEQ.NEXTVAL sequence that is normally used by the BI load process.

Performance Mart Field	BI Table Name	BI Field Name
service_point_supply_nodes.e_key	cf_cust_rst_outg	event_key
customer_service_points.cu_serv_loc_key	cf_cust_rst_outg	acct_key, prem_key, addr_key
customer_service_points.cu_cust_key	cf_cust_rst_outg	per_key
customer_service_points.cu_meter_key	cf_cust_rst_outg	meter_key
service_point_supply_nodes.cu_begin_time	cf_cust_rst_outg	begin_dttm
service_point_supply_nodes.cu_completion_time	cf_cust_rst_outg	rst_dttm
event_supply_nodes.re_key	cf_cust_rst_outg	ctrl_zone_key
event_details.re_key	cf_cust_rst_outg	cause_ctrl_zone_key
service_point_supply_nodes.cu_duration	cf_cust_rst_outg	outg_duration, cmi
event_details.e_num_momentaries	cf_cust_rst_outg	num_momentary
event_supply_nodes.dv_key	cf_cust_rst_outg	aff_device_key
event_details.dv_key	cf_cust_rst_outg	cause_device_key

EVENT_CREWS

The data in the EVENT_CREWS table is migrated to the CF_RST_CREW table. The primary key will be populated from the SPL_RST_CREW_SEQ.NEXTVAL sequence that is normally used by the BI load process.

EVENT_CREWS Field	BI Table Name	BI Field Name
cr_key	cf_rst_crew	crew_key
e_key	cf_rst_crew	event_key
ecr_crew_assn_time (assign_date_key)	cd_date	cal_dt
ecr_crew_assn_time (assign_time_key)	cd_time	src_time
ecr_crew_uassn_time (unassign_date_key)	cd_date	cal_dt
ecr_crew_uassn_time (unassign_time_key)	cd_time	src_time
ecr_crew_acpt_time (accept_date_key)	cd_date	cal_dt
ecr_crew_acpt_time (accept_time_key)	cd_time	src_time
ecr_crew_arrv_time (arrive_date_key)	cd_date	cal_dt
ecr_crew_arrv_time (arrive_time_key)	cd_time	src_time

EVENT_CREWS Field	BI Table Name	BI Field Name
ecr_crew_cmpl_time (cmpl_date_key)	cd_date	cal_dt
ecr_crew_cmpl_time (cmpl_time_key)	cd_time	src_time
ecr_crew_assn_user (assign_user_key)	cd_user	user_cd
ecr_crew_uassn_user (unassign_user_key)	cd_user	user_cd
ecr_crew_acpt_user (accept_user_key)	cd_user	user_cd
ecr_crew_arrv_user (arrive_user_key)	cd_user	user_cd
ecr_crew_cmpl_user (cmpl_user_key)	cd_user	user_cd
ecr_crew_work_dur	cf_rst_crew	WORK_ DURATION
ecr_crew_assn_dur	cf_rst_crew	ASSIGN_ DURATION
ecr_crew_disp_dur	cf_rst_crew	DISPATCH_ DURATION
ecr_crew_inroute_dur	cf_rst_crew	INROUTE_ DURATION

INDICE

The INIDICE table in Performance Mart is not migrated in the normal migration script. This is because the Indice calculations can be performed for a specific month by running this SQL*Plus command, replacing the 31-JAN-2004 with a month to calculate indice data for:

```
declare temp NUMBER;
begin
    temp := SPL_OMS_SNAPSHOT_PKG.spl_ctrl_zone_outg_snap_fnc( FALSE,
'M', to_date( '31-JAN-2004', 'DD-MON-YYYY' ), 4, 1, NULL, 3, 5, 'NORM'
);
    commit;
    temp := SPL_OMS_SNAPSHOT_PKG.spl_city_outg_snap_fnc( FALSE, 'M',
to_date( '31-JAN-2004', 'DD-MON-YYYY' ), 4, 1, NULL, 3, 5, 'NORM' );
    commit;
end;
/
```

The INDICE data is now stored in two BI tables: CF_CTRL_ZONE_OUTG and CF_CITY_OUTG. The records in the INDICE table that have an RE_KEY with a 'CIR' type will be stored in the CF_CTRL_ZONE_OUTG table, and those with a 'CITY' type will be stored in the CF_CITY_OUTG table.

These two tables also store the data that was stored in the REPORTING_ELEMENT_FACTS table for customer counts.

The following table defines the BI CF_CTRL_ZONE_OUTG table, and describes if possible where the corresponding data use to exist in Performance Mart. The fields in the CF_CITY_OUTG table have similar descriptions, so they will not be described here.

BI Field Name	Description	Corresponding Performance Mart Field
CTRL_ZONE_KEY	Foreign Key to the Control Zone Table.	INDICE.RE_KEY
TMED_IND	Does this calculation include data that was excluded due to occurring during a Major Event	INDICE..TMED_EXCLUDED
SNAP_TYPE_CD	Snapshot Type (M – Month, Y – Year, …)	N/A
SNAPSHOT_DATE_KEY	Date that the Indice data was calculated	INDICE.INDICE_DATE
BEGIN_DATE_KEY	Begin Date of the Period for which Indice calculations were performed	N/A
END_DATE_KEY	End Date of the Period for which Indice calculations were performed	N/A
NUM_CUST_SERVED	Average Number of Customers that were present in the Region during the Period	REPORTING_ELEMENTS_ FACTS. REF_CUSTOMERS_SERVED
NUM_SUST_INTRPT	Total Number of Sustained Interruptions during the snapshot period	SUM(INDICE. INTERRUPTIONS) where DURATION > 5
NUM_MOM_INTRPT	Total Number of Momentary Interruptions during the snapshot period	SUM(INDICE. INTERRUPTIONS) where DURATION < 5
CMI	Total Customer Minutes Interrupted during the snapshot period	SUM(INDICE. INTERRUPTIONS * INDICE.DURATION)

BI Field Name	Description	Corresponding Performance Mart Field
NUM_MULT_SUST_INTRPT	Total number of Customers that Experienced more than a certain number of Sustained interruptions during the snapshot period.	Calculated when a CEMI report is run.
NUM_MULT_CUST_INTRPT	Total number of Customers that Experienced more than a certain number of sustained or momentary interruptions during the snapshot period.	Calculated when a CEMSMI report is run.
SAIDI	SAIDI	Calculated when a SAIDI report is run.
CAIDI	CAIDI	Calculated when a CAIDI report is run.
SAIFI	SAIFI	Calculated when a SAIFI report is run.
CEMI	CEMI	Calculated when a CEMI report is run.
CEMSMI	CEMSMI	Calculated when a CEMSMI report is run.
CAIFI	CAIFI	Calculated when a CAIFI report is run.
MAIFI	MAIFI	Calculated when a MAIFI report is run.
MAIFIE	MAIFIe	Calculated when a MAIFIE report is run.
ASAI	ASAI	Calculated when a ASAI report is run.
ACI	ACI	Calculated when a ACI report is run.
MSAIFI	MSAIFI	Calculated when a MSAIFI report is run.
NUM_EVENT	Number of Distinct Events in Oracle Utilities Network Management System during the snapshot period	COUNT(DISTINCT INDICE.EVENT_KEY)

BI Field Name	Description	Corresponding Performance Mart Field
NUM_CUST_INTRPT	Total number of Customers that experienced one or more interruptions during the period	COUNT(DISTINCT INDICE.CUSTOMER)
NUM_MOM_E_INTRPT	Total number of Momentary Events that proceeded a lockout	SUM(INDICE.MAIFIE_INTERRUPTIONS)

NRT Table Mapping

The NRT data will not be migrated from the Performance Mart database, as this is transitional data and will need to be populated from the Oracle Utilities Network Management System database once a system is upgraded to support the BI extraction process.

However, the following table mappings are here to help with report conversion projects, and will map how the data would have been migrated if the Performance Mart NRT tables were migrated. Most the data from the NRT tables will be mapped to CF*RECENT* tables, with the exception that some textual data will be stored in either the CD_EVENT or CD_CALL_INFO tables, as described in the following sections.

Also, if a field is not listed in a mapping, then the data is not extracted from the Network Management System database to the BI database with the default product extractors. If missing data is required, then a project configuration change to the Oracle Utilities Network Management System extractors will have to be made to get the data into one of the UDF/UDM fields available in BI.

NRT_EVENT_CALL_FACTS

The data in the NRT_EVENT_CALL_FACTS table exists to two different BI tables, one dimension and one fact: CD_CALL_INFO and CF_RECENT_CALL. The following table shows which fields go into which table. For the BI tables below that are not CD_CALL_INFO or CF_RST_CALL, the mapping is done by using the foreign key in the CF_RECENT_CALL table. For example, to get the NRT_ECF_ACCOUNT_NUMBER, the CF_RECENT_CALL table would be joined to the CD_ACCT table by ACCT_KEY.

NRT_EVENT_CALL_FACTS Field	BI Table Name	BI Field Name
nrt_ecf_incident_number	cd_call_info	src_incident_id
nrt_ecf_last_name and nrt_ecf_first_name	cd_call_info	caller_name
nrt_ecf_area_cod and nrt_ecf_phone_number and nrt_ecf_phone_extension	cd_call_info	phone_nbr
nrt_ecf_complaint	cd_call_info	Complaint
nrt_ecf_operator_comment	cd_call_info	Comments
nrt_ech_short_desc	cd_call_info	udf3_descr

NRT_EVENT_CALL_FACTS Field	BI Table Name	BI Field Name
nrt_active	cd_call_info	udf1_cd
nrt_ecf_incident_number	cf_recent_call	src_incident_id
nrt_ecf_account_number	cd_acct	src_acct_id
nrt_ecf_total_priority	cf_recent_call	priority_ind
ecf_called_time (Date)	cd_date	cal_dt
ecf_called_time (Time)	cd_time	src_time
nrt_user_name	cd_user	user_cd

NRT_EVENT_DETAILS

The data in the NRT_EVENT_DETAILS table is available in two different BI tables, one dimension and one fact: CD_EVENT and CF_RECENT_JOB. The following table shows which fields go into which table.

EVENT_DETAILS Field	BI Table Name	BI Field Name
nrt_outage_number	cd_event	src_nbr
nrt_event_idx	cd_event	event_nbr
nrt_ops_exclude_reason	cd_event	exclude_reason
nrt_operator_comment	cd_event	operator_comment
nrt_valid_state_key	cd_event	event_state_descr
nrt_event_status	cd_event	event_state_cd
nrt_street_address || ',' || nrt_city_state	cd_event	first_call_addr
nrt_trouble_code	cd_event	trouble_cd_list
X_coord	cd_event	X_coordinate
Y_coord	cd_event	Y_coordinate
nrt_outage_number	cf_recent_job	src_job_nbr
nrt_status + 1	cf_recent_job	event_status_key
nrt_begin_time	cf_recent_job	begin_dttm
nrt_completion_time	cf_recent_job	rst_dttm
nrt_est_restore_time (est_rst_date_key)	cd_date	cal_dt
nrt_est_restore_time (est_rst_time_key)	cd_time	src_time
nrt_ops_exclude_flag	cf_recent_job	oms_exclude_ind
nrt_cancel_flag	cf_recent_job	cancelled_ind

EVENT_DETAILS Field	BI Table Name	BI Field Name
re_key	cf_recent_job	ctrl_zone_key
dv_key	cf_recent_job	device_key
nrt_ops_cust	cf_recent_job	udm1

NRT Customer Outage

Customer Outage information is stored in three key NRT tables in Performance Mart: NRT_SERVICE_POINT_SUPPLY_NODES, NRT_EVENT_SUPPLY_NODES and NRT_EVENT_DETAILS. Data from each of these tables as well as Customer Keys in the CUSTOMER_SERVICE_POINTS table will be available in the CF_CUST_RECENT_OUTG table in BI.

NRT Fields	BI Table Name	BI Field Name
customer_service_points.cu_serv_loc_key	cf_cust_nrt_outg	acct_key, prem_key, addr_key
customer_service_points.cu_cust_key	cf_cust_nrt_outg	per_key
customer_service_points.cu_meter_key	cf_cust_nrt_outg	meter_key
nt_event_supply_nodes.nrt_outage_time	cf_cust_recent_outg	begin_dttm
nrt_eventsupply_nodes.when_restored_time	cf_cust_recent_outg	rst_dttm
nrt_event_supply_nodes.re_key	cf_cust_recent_outg	ctrl_zone_key
nrt_event_details.re_key	cf_cust_recent_outg	cause_ctrl_zone_key
nrt_event_supply_nodes.nrt_esn_duration	cf_cust_recent_outg	outg_duration
nrt_event_supply_nodes.dv_key	cf_cust_recent_outg	aff_device_key
nrt_event_details.dv_key	cf_cust_recent_outg	cause_device_key
nrt_event_supply_nodes.level1_name	cd_ctrl_zone	udf1_descr
nrt_event_supply_nodes.level2_name	cd_ctrl_zone	udf2_descr
nrt_event_supply_nodes.level3_name	cd_ctrl_zone	udf3_descr
nrt_event_supply_nodes.level4_name	cd_ctrl_zone	udf4_descr
nrt_event_supply_nodes.level5_name	cd_ctrl_zone	udf5_descr
nrt_event_supply_nodes.level6_name	cd_ctrl_zone	udf6_descr
nrt_event_supply_nodes.num_crit_c_cust_out	cd_prem	count(*) where udf6_cd = 1
nrt_event_supply_nodes.num_crit_d_cust_out	cd_prem	count(*) where udf7_cd = 1
nrt_event_supply_nodes.num_crit_k_cust_out	cd_prem	count(*) where udf8_cd = 1

NRT_EVENT_CREWS

The data in the NRT_EVENT_CREWS table is available in the CF_RECENT_CREW table.

EVENT_CREWS Field	BI Table Name	BI Field Name
nrt_ecr_crew_assn_time (assign_date_key)	cd_date	cal_dt
nrt_ecr_crew_assn_time (assign_time_key)	cd_time	src_time
nrt_ecr_crew_uassn_time (unassign_date_key)	cd_date	cal_dt
nrt_ecr_crew_uassn_time (unassign_time_key)	cd_time	src_time
nrt_ecr_crew_acpt_time (accept_date_key)	cd_date	cal_dt
nrt_ecr_crew_acpt_time (accept_time_key)	cd_time	src_time
nrt_ecr_crew_arrv_time (arrive_date_key)	cd_date	cal_dt
nrt_ecr_crew_arrv_time (arrive_time_key)	cd_time	src_time
nrt_ecr_crew_cmpl_time (cmpl_date_key)	cd_date	cal_dt
nrt_ecr_crew_cmpl_time (cmpl_time_key)	cd_time	src_time
nrt_ecr_crew_assn_user (assign_user_key)	cd_user	user_cd
nrt_ecr_crew_uassn_user (unassign_user_key)	cd_user	user_cd
nrt_ecr_crew_acpt_user (accept_user_key)	cd_user	user_cd
nrt_ecr_crew_arrv_user (arrive_user_key)	cd_user	user_cd
nrt_ecr_crew_cmpl_user (cmpl_user_key)	cd_user	user_cd
nrt_ecr_crew_work_dur	cf_recent_crew	WORK_ DURATION
nrt_ecr_crew_assn_dur	cf_recent_crew	ASSIGN_ DURATION
nrt_ecr_crew_disp_dur	cf_recent_crew	DISPATCH_ DURATION
nrt_ecr_crew_inroute_dur	cf_recent_crew	INROUTE_ DURATION

Migration Requirements

Before running the migration script, make sure that:

- The current Performance Mart and Oracle Utilities Network Management System databases must be accessible to the BI database using database links that will be created in the BI DWADM database account.
- The BI database must be installed following the installation instructions in the *Oracle Utilities Business Intelligence Installation Guide for Oracle.*
- The following Unix environment variables point to the Performance Mart and Oracle Utilities Network Management System database.

  CES_DM_USER - Oracle Username for the Performance Mart Database

CES_DM_PASSWD - Password for the CES_DM_USER user

CES_DM_INSTANCE - SQL*Net connection to the Performance Mart Database

RDBMS_USER - Oracle Username for the Oracle Utilities Network Management System Database

RDBMS_PASSWD - Password for the RDBMS_USER user

RDBMS_HOST - SQL*Net connection to the Oracle Utilities Network Management System Database

- The following two environment variables can be set if the default settings create errors when the migration script is run.
  - **CES_DM_DBLINK** - Name of the Database Link created in the BI Oracle account to point to the Performance Mart Database. If this is not set, then the value in the CES_DM_INSTANCE environment variable is used.
  - **CES_OPS_DBLINK** - Name of the Database Link created in the BI Oracle account to point to the Oracle Utilities Network Management System Database. If this is not set, then the value in the RDBMS_HOST environment variable is used.
- Verify that you have adequate storage. The storage requirements for the BI database will be similar to the current storage requirements for the Performance Mart database. So if the data in Performance Mart takes up 5 GB of space, then a good estimate for BI storage requirement will be 5 GB.
- The following additional Unix environment variables must be set:
  - **CES_BI_USER** - Oracle Username that owns the BI data tables. Normally this will be DWADM.
  - **CES_BI_PASSWD** - Password for the CES_BI_USER user.
  - **CES_BI_INSTANCE** - SQL*Net connection to the BI Database.
  - **CES_BI_DATA_SOURCE** - Data Source Indicator that will be used when storing the migrated records in the BI tables. This should match the value in the AP_MIN_VALUE field in the APPLICATION_PARAMS table where the AP_NAME = 'DATA_SOURCE_INDICATOR'. The default setting of this is 4.
  - **CES_SQL_FILES** - Directory name where the Oracle Utilities Network Management System SQL files are stored. Normally this will be $HOME/sql. This is used by the migration script to find the project sql files.

Running the Migration Script

The migration script, **migrate_business_intelligence,** will exist in the $HOME/bin directory of the Oracle Utilities Network Management System Unix account. It can be run from this directory, as long as the requirements mentioned in the preceding section are complete.

The migration script takes no parameters, and can be run from the bin directory using this command.

```
nohup ./migrate_business_intelligence > migrate_business_intelligence.out &
```

This will create two log files. The migrate_business_intelligence.out log file can be monitored while the script is running, and the migrate_business_intelligence.log file will be updated once the migration script is completed.

For project-specific migration issues, the following two files will be called from the migration script: project_migrate_bi_dim.sql and project_migrate_bi_fact.sql. The project_migrate_bi_dim.sql will be called after all of the dimension tables are populated by the product migration script, but before the fact tables are populated, so that records will exist in all of the dimension tables for foreign keys in the fact tables. Then the project_migrate_bi_fact.sql will

be called after the fact tables are populated, but before the BI Sequences are reset. If either of these two files don't exist in the sql directory, the following messages may appear in the output file:

```
SP2-0310: unable to open file "project_migrate_bi_dim.sql"
SP2-0310: unable to open file "project_migrate_bi_fact.sql"
```

If either of these two messages appear, and the corresponding project migration script has not been created, then these errors can be ignored.

Once the migration completes, there should be data in the following BI tables, matching the records that exist in Performance Mart.

- cd_acct
- cd_addr
- cd_call_info
- cd_city
- cd_crew
- cd_ctrl_zone
- cd_device
- cd_event
- cd_event_status
- cd_meter
- cd_per
- cd_prem
- cd_snl
- cd_user
- cf_cust_rst_outg
- cf_rst_job
- cf_rst_call
- cf_rst_crew

If data is migrated from Performance Mart to BI, then the datafiles generated by the initial extractor runs of all the extractors must not be loaded into BI. Otherwise, all of the active records already stored in BI will be marked inactive, and new records generated, causing a large increase in record counts in the BI tables with no benefit. For this reason, the Oracle Utilities Network Management System must be shutdown while the migration is run and the new BI extractors must be run once. Otherwise, the potential exists for losing data that changed after the migration was run but before the new BI extractors are initially run.

To work around this issue, the LAST_START_DATE and LAST_COMPLETE_DATE in the BI_EXTRACTOR_LOG table in the Oracle Utilities Network Management System database can be updated with this command once the last Performance Mart extract is run.

```
UPDATE bi_extractor_log

SET last_start_date = SYSDATE, last_complete_date = SYSDATE

WHERE extractor_name NOT LIKE 'NRT%';
```

Note that to do this update, the Oracle Utilities Network Management System database must have been migrated and the install_business_intelligence script run to create the BI extractor code.

Troubleshooting Migration Issues

The following sections describe some common troubleshooting scenarios and the resolution.

Cannot Delete from CD_USER table

If the BI Demo environment was installed, then existing records in the CC&B fact tables can point to existing records in the CD_USER table, which will keep the delete of the CD_USER records from running. The migration script deletes all of the OMS data, but does not modify any existing CC&B or EAM records. So if you need to delete the CC&B data in order to delete the demo records in the CD_USER table, the following deletes must be done in the BI database prior to running the migration script:

```
delete from CF_FT;
delete from CF_CASE;
delete from CF_CASE_LOG;
delete from CF_CC;
```

This will not delete all of the CC&B demo data, but will delete the records that refer to CD_USER records that the migration script needs to delete.

No Data in the CF_RECENT* tables

As mentioned in the NRT Table Mapping section above, the NRT data is not migrated during the migration run. This data will be populated by extracting the NRT data from the Oracle Utilities Network Management System database and loading it into the BI Database.

No Data in the CF_CTRL_ZONE_OUTG, CF_CITY_OUTG or CF_OUTG tables

The CF_CTRL_ZONE_OUTG and CF_CITY_OUTG tables are a replacement for the INDICE table in Performance Mart. However, the data in these tables can be calculated based on the records in the CF_CUST_RST_OUTG tables, so migration of this data was not done. If records are required for these tables in the BI database, then the SPL_OMS_SNAPSHOT_PKG.SPL_CTRL_ZONE_OUTG_SNAP_FNC or the SPL_OMS_SNAPSHOT_PKG.SPL_CITY_OUTG_SNAP_FNC can be run for the periods that data is required for.

The CF_OUTG table is a snapshot table, that must be refreshed every hour by running the SPL_OMS_SNAPSHOT_PKG.SPL_OUTG_SNAP_FNC function from OWB. As this data is not available in Performance Mart, no migration was possible. This data will need to be captured from the running BI database as it is used.

Snapshots

This section presents an example call to populate snapshot tables CF_CTRL_ZONE_OUTG and CF_CITY_OUTG for last month. This really only needs to be run once a month, sometime after the last changes are made to data in Oracle Utilities Network Management System for the previous month and extracted to BI.

Control Zone Outage Snapshot

```
declare temp NUMBER;
begin
    temp := SPL_OMS_SNAPSHOT_PKG.spl_ctrl_zone_outg_snap_fnc( FALSE,
'M', ADD_MONTHS( LAST_DAY( SYSDATE ), -1 ),
                        4, 1, NULL, 3, 5, 'NORM' );
    commit;
end;
/
```

City Outage Snapshot

```
declare temp NUMBER;
begin
    temp := SPL_OMS_SNAPSHOT_PKG.spl_city_outg_snap_fnc( FALSE, 'M',
ADD_MONTHS( LAST_DAY( SYSDATE ), -1 ),
                        4, 1, NULL, 3, 5, 'NORM' );
    commit;
end;
/
```

To create a Daily Indice record set, you would change the P_SNAP_TYPE_CD, which is now 'M' for Monthly, to 'D' for Daily, and also change ADD_MONTHS(LAST_DAY(SYSDATE), -1) to TRUNC(SYSDATE - 1) to create statistics for yesterday.

The CF_OUTG table is populated from a Workflow that you can schedule to run. It takes information from the CF*RECENT tables, and calculates an hourly snapshot, so this can be scheduled to run after the RECENT records have been loaded once an hour.

For more information on Snapshots and their parameters, please see the Oracle Utilities Business Intelligence Help. To display the online help, press the button () located in the Business Intelligence Action Bar at the top of any portal screen.

Chapter 12

LDAP Integration Configuration

This chapter describes how to configure integration with Lightweight Directory Access Protocol (LDAP). It includes the following topics:

- **Overview of LDAP**
- **LDAP Terminology**
- **LDAP Integration Architecture**
- **Configuration Options**
- **Configuration Files**

Overview of LDAP

The Lightweight Directory Access Protocol (LDAP) is a generally accepted protocol for accessing a centralized set of "directory services". From an Oracle Utilities Network Management System perspective, these directory services generally include the ability to securely manage users and user passwords (authentication).

From a practical perspective, the most common implementation of an LDAP accessible directory service is Microsoft's Active Directory (AD). For many Oracle Utilities Network Management System customers Microsoft's Active Directory will likely be the directory service of choice. For the purpose of Oracle Utilities Network Management System any LDAP accessible set of directory services should be sufficient. For the remainder of this discussion "directory services" will be generally referred to as AD – realizing it could be any set of LDAP accessible directory services.

The purpose of this section is to describe available Oracle Utilities Network Management System configuration options for utilizing an LDAP compliant set of directory services (such as Microsoft Active Directory). For the purists out there Microsoft AD may not be 100% LDAP compliant. However, by using Microsoft Windows Server 2003 R2 or via Microsoft supplied augmentation briefly described later in this document for older Microsoft Operating Systems – Active Directory can generally be made sufficiently LDAP compliant for Oracle Utilities Network Management System purposes.

At a very high level Oracle Utilities Network Management System supports three different user validation options:

- NONE – legacy configuration where the ces_user table is used (exclusively) to manage Oracle Utilities Network Management System usernames/passwords. This is essentially the same way Oracle Utilities Network Management System user validation worked prior to Oracle Utilities Network Management System 1.8 with no improvement in password management. This legacy option will not be significantly covered in this document.

- LDAP_HYBRID – must provide valid LDAP/AD username/password for Oracle Utilities Network Management System access. In addition the user must also have a valid (active) user entry in the ces_user table.
- MINIMAL – must provide a valid (active) ces_user table username for Oracle Utilities Network Management System access – no password. Assumes user was authenticated when they logged into their PC.

LDAP Terminology

The following table describes terms used in this chapter.

Term	Definition
LDAP	Lightweight Directory Access Protocol – an open standard for a user security repository (Directory Services) and authentication mechanism.
LDAPS	Secure LDAP employs SSL (Secure Socket Layer) encrypted connections rather than the unencrypted connections for standard LDAP.
Directory Services	Generally accessible mechanism for managing centralized information that changes relatively infrequently (less often than say an RDBMS table for example). Oracle Utilities Network Management System is generally only concerned about identify management (user names and passwords). Microsoft Active Directory is a common implementation of Directory Services which is generally LDAP accessible.
Active Directory	Microsoft's implementation of an LDAP accessible set of Directory Services. Only available for Microsoft operation systems. Microsoft Windows Server 2003 R2 generally supports a sufficiently LDAP compliant version of Active Directory – earlier Microsoft OS releases may require an Active Directory update in order to integrate to your Unix server.
RFC 2307	Describes identify attributes (schema) necessary to support a POSIX compliant OS.

LDAP Integration Architecture

Below is a high level overview of the Oracle Utilities Network Management System LDAP integration architecture. A given Oracle Utilities Network Management System installation must make some fundamental choices about how it chooses to deploy both Motif and Java Oracle Utilities Network Management System applications.

Motif Applications

If you are deploying Oracle Utilities Network Management System Motif applications you must decide where on the Unix Application Server you intend to support the execution of those (X11) applications. You essentially have two choices–lDAP and Minimal–and you should consult with your local system administrator to determine which security model is most appropriate for your installation.

LDAP

LDAP requires that every potential Oracle Utilities Network Management System user has a corresponding Unix username on the Unix Application Server under which his/her Oracle Utilities Network Management System X11 (OPS) environment applications can be executed.

- The advantages:
  - Generally considered the most secure as it requires the user to always enter a password to login to the Unix Application Server and gain access the Oracle Utilities Network Management System applications.
- The disadvantages:
  - Requires the system administrator to manage an independent user account on the Unix Application Server for every potential Oracle Utilities Network Management System X11 (Ops) user.

Note that even though this is considered the "LDAP" configuration option it does not actually require LDAP integration. It only requires that each potential Oracle Utilities Network Management System user have a Unix user account (on the supporting Unix application server) that matches his/her Windows PC login account – where Oracle Utilities Network Management System access is initiated.

LDAP (Active Directory) integration is generally the preferred mechanism to achieve this type of configuration but there are at least two options available:

- **Preferred option**: Configure your Unix Oracle Utilities Network Management System Application Servers to access your local LDAP/Active Directory for user authentication. Consult your local Unix system administrator for Unix platform specific LDAP/Active Directory integration options. Every Oracle Utilities Network Management System supported Unix platform has at least one OS vendor supplied LDAP (Active Directory) integration option. Use your favorite search engine to find available LDAP/Active Directory integration options for your platform. Identification and appropriate configuration of the Unix LDAP integration option/gateway most appropriate for your site is beyond the scope of this document.
- **Alternate option**: Locally configure and maintain a matching set of Unix usernames on each Unix Oracle Utilities Network Management System Application Server for each potential Oracle Utilities Network Management System user. This could be done locally on each Unix box or via Network Information Services (NIS). Appropriate management of this configuration option is project specific and beyond the scope of this document.

  Note: Note if you intend to authenticate Unix usernames against "Active Directory" Microsoft Windows Server 2003 R2 is the preferred starting point for Unix integration. POSIX (Unix) accounts have some attributes (such as Unix user id, login shell and home directory) that are NOT used (by default) by earlier versions Microsoft Active Directory. These attributes are defined in RFC 2307.

  If you are using Active Directory on a Microsoft OS prior to Windows Server 2003 R2 (or a configuration which for whatever reason does not presently comply with the POSIX schema defined in RFC 2307) you may need to extend your Active Directory schema to make it sufficiently compliant to support Unix integration. For OS releases prior to Windows Server 2003 R2 Microsoft has an add-on package called "Services For Unix (SFU) 3.5" which extends the Active Directory schema to support the integration of desired POSIX attributes.

Minimal

Support the execution of your Oracle Utilities Network Management System X11 (Ops) applications on a single (dedicated) Unix username on your Unix Application Server(s). Every potential Oracle Utilities Network Management System user must have access to this dedicated

Unix username. This is a deployment strategy that is may be more useful for testing but is also in line with several existing Oracle Utilities Network Management System installations.

- The advantages::
    - Relatively easy to set up and manage Oracle Utilities Network Management System X11 (Ops) user access. Minimizes need to routinely “cleanse” stale logs from inactive Oracle Utilities Network Management System Unix user accounts on the Unix Oracle Utilities Network Management System Application Server. All logs are managed under a single Unix username which can be managed by an Oracle Utilities Network Management System administrator.
- The disadvantages::
    - Generally less secure, as it does not require a secure password to access the Unix Oracle Utilities Network Management System application server. All Oracle Utilities Network Management System users use a shared Unix username account.

The minimal scheme essentially relies on two artifacts for authentication:

- The potential Oracle Utilities Network Management System user must have first logged into his/her PC on the corporate network – generally authenticating against an LDAP accessible set of Directory Services (Active Directory). Thus they are valid corporate users.
- The potential Oracle Utilities Network Management System user must have an active entry in the ces_user table with appropriate Oracle Utilities Network Management System environment access defined in the env_access table. The Oracle Utilities Network Management System Configuration Assistant can be used by a local Oracle Utilities Network Management System administrator to manage Oracle Utilities Network Management System access – via these Oracle Utilities Network Management System access tables.

Note that for either scheme (LDAP or MINIMAL) that the Oracle Utilities Network Management System architecture and integration with the Oracle Utilities Network Management System ISIS bus will effectively prevent a given Oracle Utilities Network Management System X11 (Ops) user from ever logging in more than once concurrently. If a given user requires more than one environment (for the same Oracle Utilities Network Management System application) they must use an alternate Oracle Utilities Network Management System user access account for secondary access.

Web (Java) Applications

Oracle Utilities Network Management System Web (Java) applications can either be configured to execute:

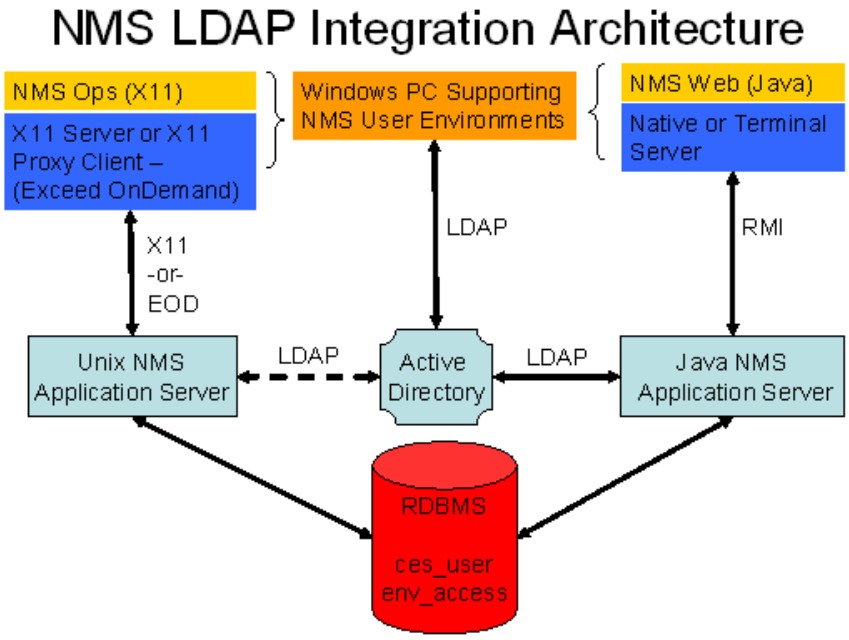
- Locally on the end-user’s PC.
- Remotely on something like a Microsoft Windows Terminal Server or Citrix Server – accessible locally via a Windows Terminal Server client or Citrix client application.

There are pros and cons to each configuration option but neither choice has a significant impact on the Oracle Utilities Network Management System security model.

There are pros and cons to each configuration option. Both the Microsoft Windows Terminal Server and Citrix Server options provide a secure mechanism for Oracle Utilities Network Management System user access over normal and low bandwidth network connections.

Running the Oracle Utilities Network Management System Web (Java) applications locally on the end-user's PC is not - by default - totally secure as the protocols between the Java applications and the Java Application Server are not entirely secure. Thus this mode of operation is generally best suited for “inside the operations firewall” installations and then only if a non-secure protocol is acceptable inside the operations firewall.

Neither choice has a significant impact on the Oracle Utilities Network Management System security model.



The dotted line between Active Directory and the Unix Oracle Utilities Network Management System Application Server in Figure 1 is for the "LDAP_HYBRID" integration option and indicates this LDAP link is optional. That is Unix usernames can be maintained locally on the Unix Oracle Utilities Network Management System Application Servers if so desired – no LDAP link required.

For the "MINIMAL" authentication configuration option neither link between Active Directory and Unix Oracle Utilities Network Management System Application Server and Java Oracle Utilities Network Management System Application Server are required – as authentication is granted based on an active entry in the ces_user table and valid application access in the env_access table. This is a less secure option that is similar to certain legacy configurations (NONE) that can be useful for performance/scalability testing.

Configuration Options

The RDBMS table ces_parameters contains three attribute/value pairs that determine the desired mode of Oracle Utilities Network Management System user authentication. Below is a table describing the LDAP relevant attributes and possible and/or example values.

ces_parameters.attrib	ces_parameters.value	Description
LDAP_DIRECTORY_SERVICES_LEVEL	NONE	Use ces_user table exclusively for authentication (legacy configuration).
LDAP_DIRECTORY_SERVICES_LEVEL	LDAP_HYBRID	Use LDAP accessible Directory Services (Active Directory) and ces_user user_name and active columns for authentication – ces_user.password not used.

ces_parameters.attrib	ces_parameters.value	Description
LDAP_DIRECTORY_SERVICES_LEVEL	LDAP_FULL	Use LDAP accessible Directory Services (Active Directory) for all authentication and authorization. Does not use ces_user table at all. Not supported for web (Java) applications.
LDAP_DIRECTORY_SERVICES_LEVEL	MINIMAL	Use ces_user user_name and active columns for validation – ces_user.password not used.
LDAP_DOMAIN	nms.local	Local LDAP (Active Directory) domain name – example. Must be set if using LDAP_HYBRID or LDAP_FULL.
LDAP_URL	ldap://10.143.76.44 or ldap://10.143.76.44:389 For Secure LDAP use: ldaps://address or ldaps://address:636	Protocol (default or secure), address and (optional) port number for Oracle Utilities Network Management System Java (JBoss or WebLogic) Application Server access to LDAP (Active Directory).

Secure LDAP (LDAPS)

If you want/need to secure LDAP communication access between the Java Application Server (JBoss or WebLogic) and the LDAP server (typically Active Directory) you can use the LDAPS protocol instead of normal LDAP for directory access. To enable LDAPS access you must:

1. Ensure your LDAP accessible directory (Active Directory) is configured to support secure (LDAPS) access. In general this means the Microsoft Certification Authority (CA) must be set up and properly configured on your Active Directory server(s). Configuration of the proper Certification Authority for Active Directory is beyond the scope of this document.
2. Copy the CA certificate from the LDAPS server and import it into the JRE used by your Java Application Server (JBoss or WebLogic). A command sequence similar to the example below should suffice. The example assumes 'msp-pdnt01' is your secure (LDAPS) server. The file 'c:/Downloaded/certnew.cer' is a default location/filename. The 'keytool' command is part of the JDK.
   - Open Internet Explorer and go to:

     ```
     http://msp-pdnt01.mydomain.com
     ```

   - Login with administrative username/password.
   - Click **Download CA Certificate**.
   - Leave encoding as "DER" and press **Download CA Certificate**.
   - Save the certificate file (defaults to c:/Downloaded/certnew.cer).
   - Move the certificate (certnew.cer) to the host running the JRE supporting your Java Application Server.
   - Import the certificate into the Java Keystore for the JRE supporting your Java Application Server:

     ```
     "# $JAVA_HOME/bin/keytool -import -alias msp-pdnt01 -file /
     path_to_certificate/certnew.cer -keystore $JAVA_HOME/jre/lib/
      security/cacerts
     ```

The alias (msp-pdnt01) can be almost any reasonable string - it is just how the certificate is referred to in the java keystore.

Configuration Files

This section identifies the configuration file that is used to specify and implement the desired LDAP integration configuration changes.

The $CES_HOME/sql/product_parameters.sql file contains the default LDAP configuration settings. If you want to change the value of the parameters in this file, you need to make a project version. If you have not already created a project version of this file to update other system parameters, do the following:

```
cp $CES_HOME/sql/product_parameters.sql $NMS_CONFIG/sql/
   ${NMS_PROJECT}_parameters.sql
```

where $NMS_PROJECT is equal to the value of the environment variable (e.g., BGE)

Modify the project version of this file (`$NMS_CONFIG/sql/${NMS_PROJECT}_parameters.sql`) as appropriate for your system configuration. Pay careful attention that the protocol, address and (optional) port specified via the LDAP_URL parameter properly match what is appropriate for your installation. Copy the updated file to the $NMS_HOME/sql runtime directory on your Oracle Utilities Network Management System server and apply the changes as follows:

```
cp $NMS_CONFIG/sql/${NMS_PROJECT}_parameters.sql $NMS_HOME/sql/
ISQL.ces $NMS_HOME/sql/${NMS_PROJECT}_parameters.sql
```

If the authentication mode is changed (from LDAP_HYBRID to MINIMAL for example) minimally DDService would need to be restarted. This is because DDService is the online "NMS authority" for what type of Oracle Utilities Network Management System application authentication/access strategy is active. The strategy is cached in DDService and cannot be fully changed on the fly without restarting DDService.

Chapter 13

Fault Location, Isolation, and Service Restoration Administration

This chapter describes how to configure and administer Fault Location, Isolation, and Service Restoration (FLISR). It includes the following topics:

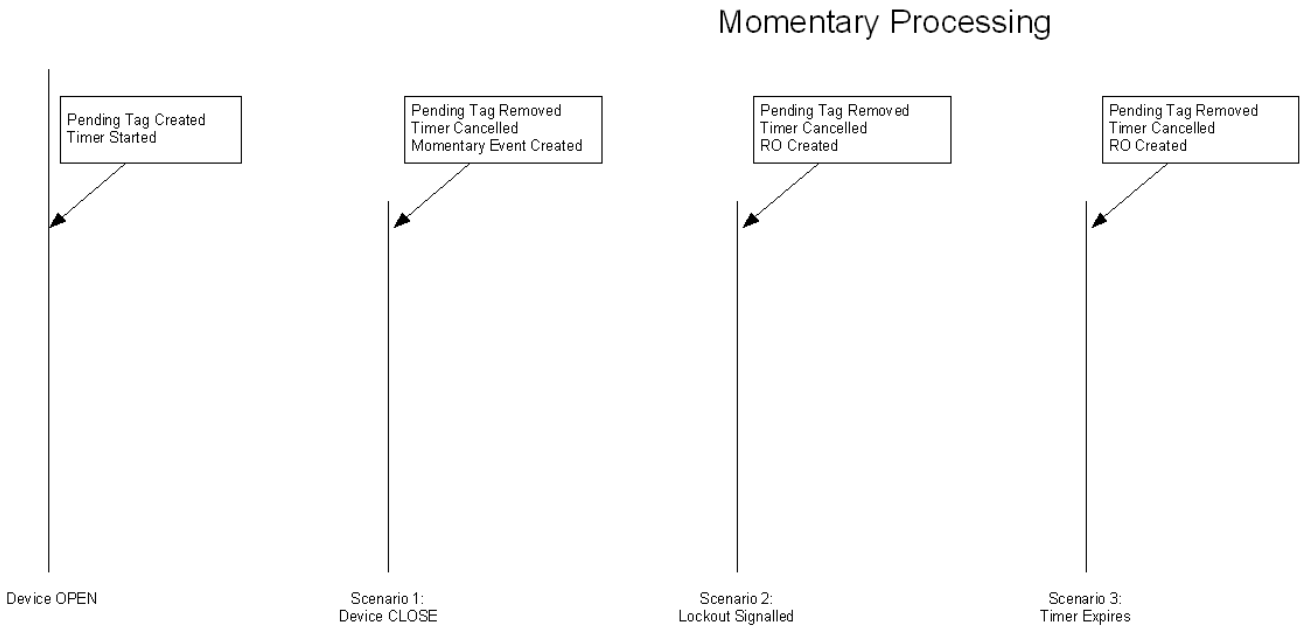
- **Introduction**
- **Fault Location, Isolation, and Service Restoration Timeline**
- **Software Architecture Overview**
- **Configuring Classes and Inheritance**
- **SRS Rules**
- **High Level Messages**
- **Troubleshooting**

Introduction

The intended audience for this document is the system administrators responsible for maintaining the Oracle Utilities Network Management System.

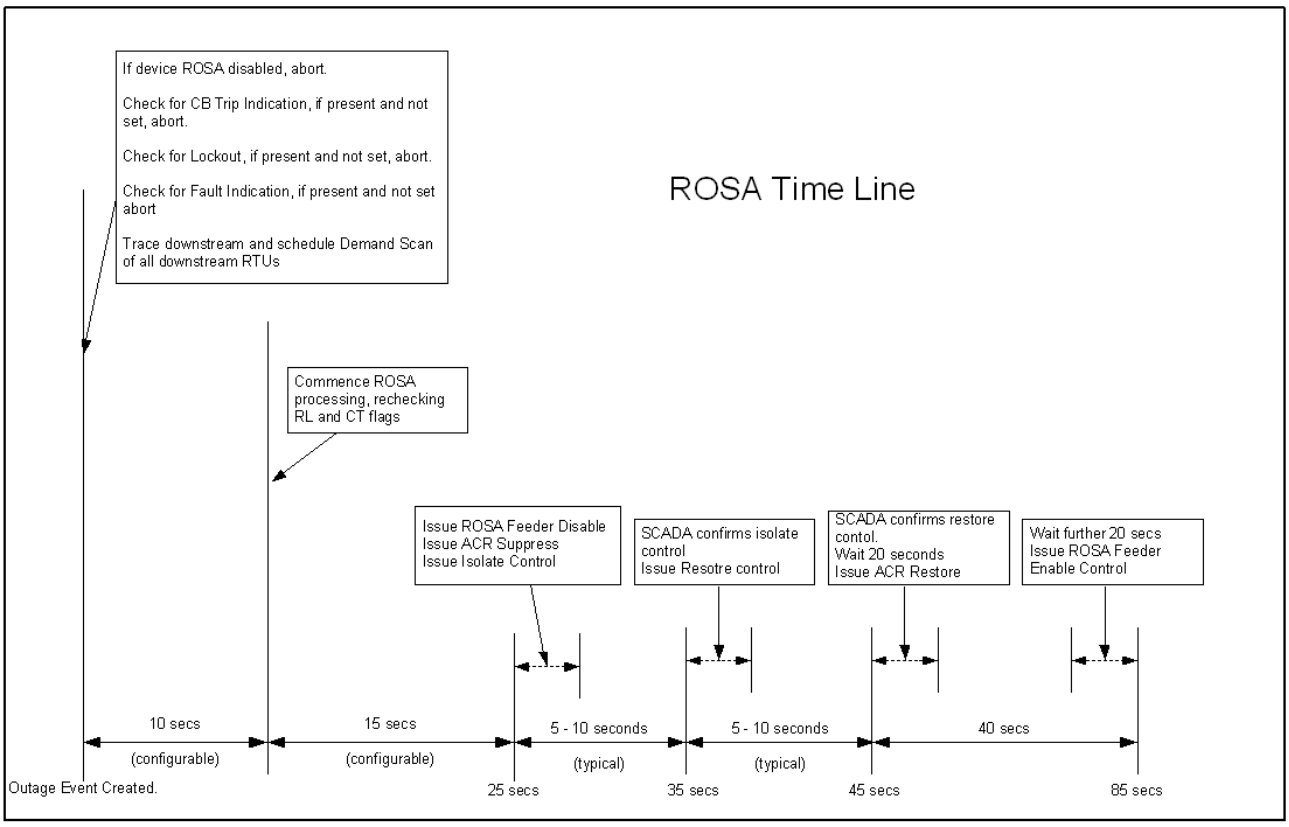
Fault Location, Isolation, and Service Restoration Timeline

These figures show the sequence of events in a Fault Location, Isolation, and Service Restoration scenario. The following figure shows the various scenarios in the momentary processing.



Note: RO is created only if customer supply nodes are de-energized as a result of the operation.

Once an RO is created, the Fault Location, Isolation, and Service Restoration processing sequence shown in the following figure is initiated.



The control sequence (starting at around 25 seconds) is only performed in automatic mode. In manual mode an operator must initiate the control sequence.

Timings in the above diagram are only indicative. Actual values will depend on the complexity of the solution required and the responsiveness of the isolate/restore controls sent to SCADA. The following timings are deterministic:

- The delay allowed for demand scans. This is configurable and defaults to 10 seconds
- The maximum time allowed for the solution in automatic mode. This is configurable and defaults to 15 seconds. If the solution takes longer to solve than this time, Fault Location, Isolation, and Service Restoration will not automatically execute the control sequence. The option for an operator to manually initiate the control sequence is preserved though.
- Maximum time allowed for automatic operations after the lockout is: Demand scan delay + 15 seconds (25 seconds in the default configuration).
- Wait times for Auto-Reclose operations. These are 20 seconds.

Software Architecture Overview

This section describes the role of various software components in implementing the Fault Location, Isolation, and Service Restoration functionality:

Component	Description
DDService	Tracks SCADA measurements, device operations and Conditions. DDService is the starting point for Fault Location, Isolation, and Service Restoration events. When a device trips, a pending operation is created. When the lockout occurs a completed device operation is sent to MTService. If the breaker is able to reclose – only a momentary event is created. DDService is also responsible for executing Fault Location, Isolation, and Service Restoration switch plans, both in manual and automatic mode. In manual mode the request to execute the switch plan can be initiated by the operator from the Switch Sheet Editor tool. In automatic mode the Fault Location, Isolation, and Service Restoration sub-system requests the switch sheet execution by DDService
MTService	The core of Fault Location, Isolation, and Service Restoration functionality. It contains most of the Fault Location, Isolation, and Service Restoration sub-system. Its initial task is to process device operations from DDService and determine the extent of energisation changes in the model. These changes are propagated to JMService for outage processing. If the device operation is a trip, the Fault Location, Isolation, and Service Restoration sub-system will perform an initial trace to initiate a demand scan of affected RTUs. The bulk of Fault Location, Isolation, and Service Restoration processing is triggered by JMService deciding that event has de-energised customers. In this scenario JMService instructs MTService to initiate Fault Location, Isolation, and Service Restoration processing. MTService then calculates the various isolate and restore scenarios and populates the database tables with the solutions.
JMService	Receives notifications from MTService about changes in energization on the network. JMService will determine if these changes de-energises customers and if so creates an outage event and informs MTService that Fault Location, Isolation, and Service Restoration processing of that event is required.

Component	Description
WorkAgenda	Monitors notifications from JMService about the creation, update and completion of events. WorkAgenda is configured to highlight Fault Location, Isolation, and Service Restoration events in various ways: • Events detected as potential Fault Location, Isolation, and Service Restoration events are highlighted with a yellow background. The background stays yellow until a Fault Location, Isolation, and Service Restoration solution is found or a further determination indicates that the event cannot be considered an FLISR event (e.g., all restoring switches or feeders are Fault Location, Isolation, and Service Restoration disabled) • Events for which a viable Fault Location, Isolation, and Service Restoration solution is found are highlighted with a pink background. • Events for which a Fault Location, Isolation, and Service Restoration solution is found, but the solution includes overloads on restoring feeders, are highlighted with a light blue background.
rosatool	Provides a summary of the Fault Location, Isolation, and Service Restoration solution for an event. If an event is found to have a Fault Location, Isolation, and Service Restoration solution, the operator can examine the details of that solution by using this tool. This tool primarily reads the database tables to determine the solution information calculated by MTService. The operator can also manually write, append and/or overwrite the generated switch plan.
Switching	Once a solution is found for the Fault Location, Isolation, and Service Restoration event, a switch plan can be created to execute the solution. The switch plan can be created (and executed) automatically, or it can be created manually. In either scenario the switch plan can be viewed from the Switch Sheet Editor. In manual mode the operator can request that DDService execute the plan. In both manual and automatic mode the operator can watch the results of DDService performing a switch plan execution.
Eman	The eman panel has a few Fault Location, Isolation, and Service Restoration features: • Status – the current Fault Location, Isolation, and Service Restoration status is displayed. Indicating if FLISR is disabled, in manual mode or automatic mode. Toolbox – the Fault Location, Isolation, and Service Restoration toolbox contains buttons to change the FLISR status and bring up various summaries.

Configuring Classes and Inheritance

Fault Location, Isolation, and Service Restoration utilizes standard class names to determine various features in the model. Devices in a model can be configured to the Fault Location, Isolation, and Service Restoration classes using class inheritance.

The following table lists the classes supported by Fault Location, Isolation, and Service Restoration:

Class Name	Purpose
flisr_cb	Set of SCADA devices that are protective. These are the SCADA devices that can trip when a fault is detected.
flisr_sectionalizer	Set of devices that are SCADA controllable, but are not protective. These devices: Might have fault indicators on them in order to give better indication of fault locations on the feeder Will be considered for isolate and restore devices
flisr_fuse	Set of non-SCADA protective devices. These are considered when determining loads and limiting devices
flisr_load	Set of devices that are loads on the network – typically distribution transformers.
flisr_cogen	Set of devices on the network that provide additional supply.
conductor	Set of conductor classes on the network. These are considered when determining limiting devices.
block_flisr	Condition classes. These define tags and conditions that automatically prohibit Fault Location, Isolation, and Service Restoration operations on a device.

Database Views

In order to determine loads and limiting devices Fault Location, Isolation, and Service Restoration needs to know basic load profile information about all devices. The following database VIEWS are required:

FLISR_TRANSFORMER

h_cls	INTEGER	Class number of device
h_idx	INTEGER	Index number of device
kva_rating	FLOAT	Transformer rating in kVA
partition	INTEGER	Model partition for device

FLISR_CONDUCTOR

h_cls	INTEGER	Class number of device
h_idx	INTEGER	Index number of device
amp_rating	FLOAT	Device's rating in amps
voltage	FLOAT	Device's nominal voltage in kV
partition	INTEGER	Model partition for device

FLISR_SWITCH

h_cls	INTEGER	Class number of device
h_idx	INTEGER	Index number of device
amp_rating	FLOAT	Device's rating in amps
voltage	FLOAT	Device's nominal voltage in kV
partition	INTEGER	Model partition for device

SRS Rules

The following SRS Rules configure Fault Location, Isolation, and Service Restoration functionality and options:

Rule Name	Description
allowFlisrAutoMode	Allow the operators to put Fault Location, Isolation, and Service Restoration into auto-mode
autoRecloseMeasurementName	SCADA attribute used to indicate recloser suppression
earthLeakageMeasurementName	SCADA attribute for earth leakage
failedQualityBitmask	The bitmask to apply to quality codes to determine if quality is bad.
faultIndicatorMeasurementName	SCADA attribute for Fault Indicators
flisrDemandScanThreshold	Time to wait for demand scans
flisrDisableMeasurementName	SCADA attribute that indicates Fault Location, Isolation, and Service Restoration should be disabled
flisrKVATolerance	KVA Tolerance when comparing loads against ratings
flisrMode	Start up mode for Fault Location, Isolation, and Service Restoration

Rule Name	Description
flisrSwitchPlanType	Type of switch plans to use for Fault Location, Isolation, and Service Restoration
flisrTemplateArEnable	Template containing Fault Location, Isolation, and Service Restoration Reclose Enable actions
flisrTemplateArSuppress	Template containing FLISR Reclose Suppress actions
flisrTemplateBase	Template for FLISR switch plans
flisrTemplateDisable	Template containing FLISR Disable actions
flisrTemplateEnable	Template containing FLISR Enable actions
flisrTemplateIsolate	Template containing FLISR Isolate actions
flisrTemplateRestore	Template containing FLISR Restore actions
flisrTemplateWait	Template containing FLISR Reclose Wait actions
manualOperationMeasurementName	SCADA attribute that indicates manual operation of a device
maxFlisrSolutionTime	How long we allow for solutions in automatic mode
mvarMeasurementName	SCADA attribute for current MVAR
mwMeasurementName	SCADA attribute for current MW
preTripMvarMeasurementName	SCADA attribute for pre-trip MVAR
preTripMwMeasurementName	SCADA attribute for pre-trip MW
recloseLockoutMeasurementName	SCADA attribute used to show recloser lockouts

High Level Messages

MTService accepts the following High Level messages:

```
Action any.MTService <command> <arguments>
```

Where:

Command	Arguments	Description
debug FLISR	<N>	Sets the debug level: 0 = off 1 = demand scan & timing info 2 = Trace 3 = Detailed Information regarding solution 4 = Full debug
flisr kva_tolerance	<N>	Sets the capacity tolerance to allow. Where <N> is the new tolerance in kVA
flisr base_flows		Outputs the base conductor flow information

Command	Arguments	Description
flisr ties		Outputs the ties (open) point summary
flisr alarms		Forces a check for the Fault Location, Isolation, and Service Restoration disabled device alarms
flisr check	ON/OFF	Toggle Fault Location, Isolation, and Service Restoration check mode on/off
flisr reload		Reload measurement configuration
flisr dump		Write internal data structures into log

Troubleshooting

The following high-level messages can be used to turn timing and demand scan information on/off. This is useful in determining that Fault Location Isolation Service Restoration is scanning the correct RTUs and that timing goals are being achieved.

To turn on the messages:

```
Action any.MTService debug FLISR 1
```

To turn off the messages:

```
Action any.MTService debug FLISR 0
```

Chapter 14

Distribution Management Application Configuration

This chapter provides an overview of the configuration and maintenance of Oracle Utilities Distribution Management System applications. It includes the following topics:

- **Environment Settings**
- **Configuring Oracle Utilities Network Management Services**
- **PFS_RULES Settings**

For DMS installation instructions, see the Oracle Utilities Network Management System Installation Guide.

Environment Settings

This section describes how the Oracle Utilities Network Management System Distribution Management services are configured. These settings should be configured for the applications listed below. Some specific settings are needed for the Java GUI based applications.

- X/Motif GUI Applications
  - Power Flow Extensions
  - Optimal Power Flow
  - Suggested Switching
  - Fault Location Isolation & Service Restoration
  - Volt/Var Optimization
- Java GUI Applications
  - Feeder Load Management
  - Fault Location Analysis
  - Web Switching

Configuring Oracle Utilities Network Management Services

PFService – Power Flow Service

The main application that runs the majority of the Oracle Utilities Network Management System Distribution Management business logic is the Power Flow service. If your environment will be running any of the applications listed above (except Web Switching), you must add the Power Flow Service as a system service by updating the $CES_HOME/etc/system.dat file. There are 3 main sections where this service needs to be defined: the service, program and instance sections. See the $CES_HOME/templates/system.dat.template file for examples of how to configure the Powerflow Service. Search for PFService in the file and copy those lines to $CES_HOME/etc/system.dat file. Make sure all lines are uncommented so that they are active. You must restart the system services in order for the Powerflow Service to be properly monitored by SMService.

The command line options for PFService are:

- **LPUpdate**: PFService should be run with this option to activate the periodic update of the historical load profile
- **hourlyProfiles**: PFService should be run with this option to activate the load interval data functionality

MTService – Model Topology Service

MTService should be run with the –trace_after_operate command line option if the Power Flow Service (PFService) is enabled to run in your environment. With this option, for device operations that energize a section of the network, MTService performs an extra trace to pass PFService information it needs about the topological island's new extents.

NMAdapter – Network Management Adapter

If your environment will be running any of the Java GUI applications (Feeder Load Management, Fault Location Analysis or Web Switching), the NMAdapter must be configured as a system service. The NMAdapter is the service that allows the Java GUI applications to communicate to specific X/Motif applications. Add the NMAdapter service as a system service by updating the $CES_HOME/etc/system.dat file. There are 3 main sections where this service needs to be defined: the service, program and instance sections. See the $CES_HOME/templates/system.dat.template file for examples of how to configure the NMAdapter Service. Search for NMAdapter in the file and copy those lines to $CES_HOME/etc/system.dat file. Make sure all lines are uncommented so that they are active. You must restart the system services in order for the NMAdapter to be properly monitored by SMService. See the following section for the setting of a specific environment variable that must also be set for the NMAdapter to work properly.

Configuring CES_HAS_NMA Environment Variable for Java GUI Applications

If you will be running any of the Java GUI Oracle Utilities Network Management System Distribution Management applications (Feeder Load Management, Fault Location Analysis or Web Switching), ensure that CES_HAS_NMA environment variable is set and properly updated in the $CES_HOME/.nmsrc file in the UNIX environment where services and X/Motif client applications will be started. This allows the user to run the applications in “hybrid” mode meaning that the Distribution Management Java GUI applications can communicate with other X/Motif applications such as the Work Agenda and Viewer

In order to make sure this variable is properly set, complete the following steps:

1. Alter the ~/.nmsrc file and add/uncomment the following line:

   ```
   export CES_HAS_NMA=1
   ```

2. Stop all services and log out of all X/Motif client applications
3. Log out and back into the environment
4. Validate that CES_HAS_NMA is set to 1 by running the command "env | grep CES_HAS_NMA". It should return you a value. If nothing is returned back to you, then the variable is not set.
5. Restart all services
6. Start the Motif Operator's workspace with 'ceslogin -user <user ID>', for example 'ceslogin -user oms2', and use that same user ID in the Operator's Workspace login window, and in the Java GUI login window.
7. Log in to the Java GUI environment using the "Web DMS" URL for Feeder Load Management or Fault Location Analysis. Use the "Web Switching" URL for a Web Switching user. Log in with the same user ID as the previous step.

Web DMS User Type Configuration

This section describes configuration settings for the Java GUI based Oracle Utilities Network Management System Distribution Management applications, including Feeder Load Management and Fault Location Analysis. The Web DMS tool uses the standard Oracle Utilities Network Management System login window with "Full Operations," "Administration" and "View Only" user types available. A user may be assigned either one of the user types, or any combination of them.

By default there are five Web DMS users with all three types of access privileges (user types) – oms1, oms2, oms3, oms4 and nms1. All user types are defined in the env_code table, and the access privileges for the user are defined in the env_access table. Please refer to the Configuration Assistant chapter of the Oracle Utilities Network Management System User Guide for details on adding or changing these user definitions.

> **NOTE:** When running any of the Java GUI applications (Feeder Load Management, Fault Location Analysis or Web Switching), you must also log into the corresponding X/Motif environment as the SAME USER that you used to log in to the Java GUI environment. This enables the two application environments to interact with each other. Failure to log in using the same user account will lead the two environments to not work properly together.

Power Flow User Type Configuration

This section describes configuration settings for Power Flow user types which are used for the X/Motif Oracle Utilities Network Management System Distribution Management tools.

There are three possible Power Flow user types: powerflow_admin, powerflow_operator and powerflow_viewer. Each user type has configurable access privileges to various Oracle Utilities Network Management System Distribution Management tools and functionalities. Refer to the PFS_RULES configurations section for more details on access privilege configurations for each of these user types.

PFS_RULES Settings

This section lists PFS_RULES parameters, their description and typical configuration values/ranges. Oracle Utilities Network Management System Distribution Management applications use the pfs_rules parameters to configure what kind of data sets are used and how the application results are computed and displayed. In order to make customer specific changes to these parameters, please do the following:

The `$CES_HOME/sql/product_pfs_rules.sql` file contains the default Power Flow configuration settings. If you want to change the value of the parameters in this file, you need to

make a customer version. If you have not already created a customer version of this file to update other Power Flow parameters, do the following:

```
cp $CES_HOME/sql/product_pfs_rules.sql $NMS_CONFIG/sql
   /${NMS_PROJECT}_pfs_rules.sql
```

where `$NMS_PROJECT` is equal to the value of the environment variable (e.g. BGE)

Modify the customer version of this file (`$NMS_CONFIG/sql/${NMS_PROJECT}_pfs_rules.sql`) as appropriate for your system configuration.

Copy the updated file to the `$NMS_HOME/sql` runtime directory on your Oracle Utilities Network Management System server and apply the changes as follows:

```
cp $NMS_CONFIG/sql/${NMS_PROJECT}_pfs_rules.sql $NMS_HOME/sql/

ISQL.ces $NMS_HOME/sql/${NMS_PROJECT}_pfs_rules.sql
```

Note: After applying these changes, you must restart the PFService application.

Parameters	Description
CYCLE_TIME	Cycle time for periodic powerflow solution (in secs). Typical val: 100000
TIME_THRESHOLD	Minimum time window to be enforced between analog/digital network status changes triggered powerflow solutions (in secs). Typical val: 30
LOAD_PERIODS	Historical load profile processing - Number of load periods in a day. Typical val: 24
NOM_TEMP	Nominal temperature for load profile updates. Typical val: 65
SMOOTHING	Smoothing constant for load profile updates. Typical val: 1.5
SUMMER_DATE	Summer start date for season-dependent load types. Typical val: 0501
WINTER_DATE	Winter start date for season-dependent load types. Typical val: 0501
DAY_TYPES	Number of configured day types. Typical val: 10
TYPE_1_DAYS	Day type 1. Typically weekdays of season 1. Typical val: DAYNAME_SEASON_1, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
TYPE_2_DAYS	Day type 2. Typically weekends of season 1. Typical val: DAYNAME_SEASON_1, SUNDAY, SATURDAY
TYPE_3_DAYS	Day type 3. Typical val: DAYNAME_SEASON_2, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
TYPE_4_DAYS	Day type 4. Typical val: DAYNAME_SEASON_2, SUNDAY, SATURDAY
TYPE_5_DAYS	Day type 5. Typical val: DEFAULT
TYPE_6_DAYS	Day type 6. Typical val: DATE, 5-17, 7-4, 12-24

Parameters	Description
TYPE_7_DAYS	Day type 7. Typical val: DAYNAME_SEASON_3, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
TYPE_8_DAYS	Day type 8. Typical val: DAYNAME_SEASON_3, SUNDAY, SATURDAY
TYPE_9_DAYS	Day type 9. Typical val: DAYNAME_SEASON_4, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY
TYPE_10_DAYS	Day type 10. Typical val: DAYNAME_SEASON_4, SUNDAY, SATURDAY
ALARM_PHASES	Flag that decides whether alarm is sent for only 1 phase or for all 3. Typical value: 3
LOADTYPE,0,0	Voltage computation parameter for load model of loadtype 0. Typical val: 1
LOADTYPE,0,1	Refer above. Typical val: 1
LOADTYPE,0,2	Refer above. Typical val: 0
LOADTYPE,0,3	Refer above. Typical val: 0
LOADTYPE,0,4	Refer above. Typical val: 1
LOADTYPE,0,5	Refer above. Typical val: 0
LOADTYPE,1,0	Refer above. Typical val: 0
LOADTYPE,1,1	Refer above. Typical val: 0.6
LOADTYPE,1,2	Refer above. Typical val: 0
LOADTYPE,1,3	Refer above. Typical val: 0.4
LOADTYPE,1,4	Refer above. Typical val: 0.6
LOADTYPE,1,5	Refer above. Typical val: 0
LOADTYPE,2,0	Refer above. Typical val: 0.4
LOADTYPE,2,1	Refer above. Typical val: 0.5
LOADTYPE,2,2	Refer above. Typical val: 0
LOADTYPE,2,3	Refer above. Typical val: 0.5
LOADTYPE,2,4	Refer above. Typical val: 0.5
LOADTYPE,2,5	Refer above. Typical val: 0
LOADTYPE,3,0	Refer above. Typical val: 0.5
LOADTYPE,3,1	Refer above. Typical val: 0.4
LOADTYPE,3,2	Refer above. Typical val: 0
LOADTYPE,3,3	Refer above. Typical val: 0.6
LOADTYPE,3,4	Refer above. Typical val: 0.4

Parameters	Description
LOADTYPE,3,5	Refer above. Typical val: 0
LOADTYPE,4,0	Refer above. Typical val: 0.6
LOADTYPE,4,1	Refer above. Typical val: 0
LOADTYPE,4,2	Refer above. Typical val: 0
LOADTYPE,4,3	Refer above. Typical val: 1
LOADTYPE,4,4	Refer above. Typical val: 0
LOADTYPE,4,5	Refer above. Typical val: 0
SINGLE_PH_FAULT_IMPEDANCE	Single phase fault imp. Typical val: 5.0
TWO_PH_FAULT_IMPEDANCE	Two phase fault imp. Typical val: 10.0
THREE_PH_FAULT_IMPEDANCE	Three phase fault imp. Typical val: 5.0
TAPSW_ENABLE	Flag that determines whether transformer taps are enabled as controls. Typical val: 1 (for Enable)
CAPSW_ENABLE	Flag that determines whether sw. capacitors are enabled as controls. Typical val: 1 (for Enable)
CAPTIMEVAR_ENABLE	Flag that determines whether time-switched capacitors are enabled as controls. Typical val: 1 (for Enable)
VHILIMIT	Percent high voltage limit value. Typical val: 1.08
VLOLIMIT	Percent low voltage limit value. Typical val: 1.08
PEAK_LOAD_STRATEGY	0 for Daily Peak, 1 for Seasonal Peak, or 2 for Both. Typical val: 1
PEAK_SCENARIOS	0 for Derivation: derive Load from Historical Load Profile, 1 Scale: Scale Load from User Entered Scaling factor, 2 None. Typical val: 0
PRI_VOLTAGE_LEVEL	Primary voltage level (in kV). Typical val: 12.47
SUBX_VOLTAGE_LEVEL	Primary voltage level (in kV). Typical val: 69.0
SEC_VOLTAGE_LEVEL	Secondary voltage level (in kV). Typical val: 0.12
DISPLAY_VOLTAGE_TYPE	Powerflow results voltage display base. 0=secondary voltage level, 1=primary voltage level. Typical val: 0
SEC_VOLTAGE_BASE	Secondary voltage base (in V). Typical val: 120
TRACEVOLT_DATA_SOURCE	The parameter decides what data source will be used while setting nominal volts downstream of xfmrs. 0 = xfmr secondary base voltage (from pf_xfmrs) will be used, 1 = line voltage (from network_components) will be used
PF_TRIGGERS_ANALOG	1=Powerflow triggered on analog measurement updates, 0=not triggered. Typical val: 1
PF_TRIGGERS_DIGITAL	1=Powerflow triggered on digital status updates, 0=not triggered. Typical val: 1

Parameters	Description
PF_TRIGGERS_PERIODIC	1=Powerflow solution run periodically, 0=not run periodically. Typical val: 0
PF_TRIGGERS_ONDEMAND	1=Powerflow triggered on demand, 0=not triggered. Typical val: 1
ANALOG_PRECISION	Analog precision percentage. Typical val: 5
LOAD_SCALING_TYPE	0=Dynamic Load Set Scaling, 1=All Downstream Load Scaling, 2=Feeder Load Set Scaling. Typical Val: 1
DATASOURCE_DEFAULT	Display string for default data source (solved powerflow values). Typical val: 'Default'
DATASOURCE_SCADA	Display string for scada data source (analog measurement). Typical val: 'SCADA'
DATASOURCE_MANUAL_ENTRY	Display string for manual data source (user-entered). Typical val: 'Manual Entry'
DATAQUALITY_PRIORITY_VALID	Defines priority level for data with VALID quality code. Typical val: 5
DATAQUALITY_STRING_VALID	Display string for data with VALID quality code. Typical val: 'Valid'
DATAQUALITY_COLOR_VALID	Display color for data with VALID quality code. Typical val: 'Green'
DATAQUALITY_ICCPCODES_VALID	Defines associated ICCPGateway quality code for data with VALID quality code. Typical val: '0'
DATAQUALITY_PRIORITY_HELD	Defines priority level for data with HELD quality code. Typical val: 7
DATAQUALITY_STRING_HELD	Display string for data with HELD quality code. Typical val: 'Held'
DATAQUALITY_COLOR_HELD	Display color for data with HELD quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_HELD	Defines associated ICCPGateway quality code for data with HELD quality code. Typical val: '16,8192'
DATAQUALITY_PRIORITY_ESTIMATED	Defines priority level for data with ESTIMATED quality code. Typical val: 6
DATAQUALITY_STRING_ESTIMATED	Display string for data with ESTIMATED quality code. Typical val: 'Estimated'
DATAQUALITY_COLOR_ESTIMATED	Display color for data with ESTIMATED quality code. Typical val: 'Green'
DATAQUALITY_ICCPCODES_ESTIMATED	Defines associated ICCPGateway quality code for data with ESTIMATED quality code. Typical val: '32,32768,65536,131072'
DATAQUALITY_PRIORITY_ALARM	Defines priority level for data with ALARM quality code. Typical val: 8

Parameters	Description
DATAQUALITY_STRING_ALARM	Display string for data with ALARM quality code. Typical val: 'Alarm'
DATAQUALITY_COLOR_ALARM	Display color for data with ALARM quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_ALARM	Defines associated ICCPGateway quality code for data with ESTIMATED quality code. Typical val: '256,512,1024'
DATAQUALITY_PRIORITY_ABNORMAL	Defines priority level for data with ABNORMAL quality code. Typical val: 4
DATAQUALITY_STRING_ABNORMAL	Display string for data with ABNORMAL quality code. Typical val: 'Abnormal'
DATAQUALITY_COLOR_ABNORMAL	Display color for data with ABNORMAL quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_ABNORMAL	Defines associated ICCPGateway quality code for data with ESTIMATED quality code. Typical val: '64,128,16384'
DATAQUALITY_PRIORITY_SUSPECT	Defines priority level for data with SUSPECT quality code. Typical val: 3
DATAQUALITY_STRING_SUSPECT	Display string for data with SUSPECT quality code. Typical val: 'Suspect'
DATAQUALITY_COLOR_SUSPECT	Display color for data with SUSPECT quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_SUSPECT	Defines associated ICCPGateway quality code for data with SUSPECT quality code. Typical val: '4096'
DATAQUALITY_PRIORITY_OVERLIMIT	Defines priority level for data with OVERLIMIT quality code. Typical val: 2
DATAQUALITY_STRING_OVERLIMIT	Display string for data with OVERLIMIT quality code. Typical val: 'Over Limit'
DATAQUALITY_COLOR_OVERLIMIT	Display color for data with OVERLIMIT quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_OVERLIMIT	Defines associated ICCPGateway quality code for data with OVERLIMIT quality code. Typical val: '2'
DATAQUALITY_PRIORITY_UNDERLIMIT	Defines priority level for data with UNDERLIMIT quality code. Typical val: 2
DATAQUALITY_STRING_UNDERLIMIT	Display string for data with UNDERLIMIT quality code. Typical val: 'Under Limit'
DATAQUALITY_COLOR_UNDERLIMIT	Display color for data with UNDERLIMIT quality code. Typical val: 'Yellow'
DATAQUALITY_ICCPCODES_UNDERLIMIT	Defines associated ICCPGateway quality code for data with UNDERLIMIT quality code. Typical val: '1'
DATAQUALITY_PRIORITY_INVALID	Defines priority level for data with INVALID quality code. Typical val: 1

Parameters	Description
DATAQUALITY_STRING_INVALID	Display string for data with INVALID quality code. Typical val: 'Not Valid'
DATAQUALITY_COLOR_INVALID	Display color for data with INVALID quality code. Typical val: 'Red'
DATAQUALITY_ICCPCODES_INVALID	Defines associated ICCPGateway quality code for data with INVALID quality code. Typical val: '2048'
DATACHECK_SRC_VALID_MIN_VOL	Data check rule. Minimum voltage for which data quality may be defined as VALID. Typical val: 1.0
DATACHECK_SRC_VALID_MAX_VOL	Maximum voltage for which data quality may be defined as VALID. Typical val: 1.1
DATACHECK_SRC_SUSPECT_MIN_VOL	Minimum voltage for which data quality may be defined as SUSPECT. Typical val: 0.95
DATACHECK_SRC_SUSPECT_MAX_VOL	Maximum voltage for which data quality may be defined as SUSPECT. Typical val: 1.15
DATACHECK_SRC_UNDERLIMIT_MIN_VOL	Minimum voltage for which data quality may be defined as OVERLIMIT. Typical val: 0.9
DATACHECK_SRC_OVERLIMIT_MAX_VOL	Maximum voltage for which data quality may be defined as OVERLIMIT. Typical val: 1.2
DATACHECK_FDR_SUSPECT_AMP_MISMATCH	Data check rule. Maximum tolerance margin for which feeder amp data quality may be defined as SUSPECT. Typical val: 0.2
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_CONTROLS	Defines the access privilege of PF user of type 'Viewer' for the controls settings of PF Options tool under real time mode, or study mode. 'X'=cannot open tool, or all options on tool are disabled, 'RW'=user has full view and modify authority for specified tool, 'RO'=user has only viewing authority for specified tool. Typical val for this setting: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_LIMITS	Defines the access privilege of PF user of type 'Viewer' for the limits settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_LOAD_DIST	Defines the access privilege of PF user of type 'Viewer' for the load distribution settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_MULT_LOAD	Defines the access privilege of PF user of type 'Viewer' for the multiple load scenario settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_FAULT_IMPEDANCE	Defines the access privilege of PF user of type 'Viewer' for the fault impedance settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_PSS_FILE	Defines the access privilege of PF user of type 'Viewer' for the PSS file settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'

Parameters	Description
ACCESS_PRIVILEGE_PFVIEWER_FOR_OPTIONS_TRIGGERS	Defines the access privilege of PF user of type 'Viewer' for the triggers settings of PF Options tool under real time mode, or study mode. Typical val: 'X,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_FEED_LOAD	Defines the access privilege of PF user of type 'Viewer' for the feeder load settings of PF Options tool under real time mode, or study mode. Typical val: 'RO,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_SUB_LOAD	Defines the access privilege of PF user of type 'Viewer' for the substation load settings of PF Options tool under real time mode, or study mode. Typical val: 'RO,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_SHUNTS	Defines the access privilege of PF user of type 'Viewer' for the shunt settings of PF Options tool under real time mode, or study mode. Typical val: 'RO,RW'
ACCESS_PRIVILEGE_PFVIEWER_FOR_XFMRS	Defines the access privilege of PF user of type 'Viewer' for the transformer settings of PF Options tool under real time mode, or study mode. Typical val: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_CONTROLS	Defines the access privilege of PF user of type 'Operator' for the controls settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_LIMITS	Defines the access privilege of PF user of type 'Operator' for the limits settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_LOAD_DIST	Defines the access privilege of PF user of type 'Operator' for the load distribution settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_MULT_LOAD	Defines the access privilege of PF user of type 'Operator' for the multiple load scenario settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_FAULT_IMPEDANCE	Defines the access privilege of PF user of type 'Operator' for the fault impedance settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_PSS_FILE	Defines the access privilege of PF user of type 'Operator' for the PSS file settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_OPTIONS_TRIGGERS	Defines the access privilege of PF user of type 'Operator' for the triggers settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RO,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_FEED_LOAD	Defines the access privilege of PF user of type 'Operator' for the feeder load settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_SUB_LOAD	Defines the access privilege of PF user of type 'Operator' for the substation load settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'

Parameters	Description
ACCESS_PRIVILEGE_PFOPERATOR_FOR_SHUNTS	Defines the access privilege of PF user of type 'Operator' for the shunt settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFOPERATOR_FOR_XFMRS	Defines the access privilege of PF user of type 'Operator' for the transformer settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ CONTROLS	Defines the access privilege of PF user of type 'Admin' for the controls settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ LIMITS	Defines the access privilege of PF user of type 'Admin' for the limits settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ LOAD_DIST	Defines the access privilege of PF user of type 'Admin' for the load distribution settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ MULT_LOAD	Defines the access privilege of PF user of type 'Admin' for the multiple load scenario settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ FAULT_ IMPEDANCE	Defines the access privilege of PF user of type 'Admin' for the fault impedance settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ PSS_FILE	Defines the access privilege of PF user of type 'Admin' for the PSS file settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_OPTIONS_ TRIGGERS	Defines the access privilege of PF user of type 'Admin' for the triggers settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_FEED_LOAD	Defines the access privilege of PF user of type 'Admin' for the feeder load settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_SUB_LOAD	Defines the access privilege of PF user of type 'Admin' for the substation load settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_SHUNTS	Defines the access privilege of PF user of type 'Admin' for the shunt settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
ACCESS_PRIVILEGE_PFADMIN_FOR_XFMRS	Defines the access privilege of PF user of type 'Admin' for the transformer settings of PF Options tool under real time mode, or study mode. Typical val for this setting: 'RW,RW'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_SCENARIO	Defines the access privilege of PF user of type 'Viewer' for the scenario settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_CONTROLS	Defines the access privilege of PF user of type 'Viewer' for the controls settings of OPF Options tool. Typical val: 'X'

Parameters	Description
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_CONSTRAINTS	Defines the access privilege of PF user of type 'Viewer' for the constraints settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_PARAMETERS	Defines the access privilege of PF user of type 'Viewer' for the parameters settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_WEIGHTS	Defines the access privilege of PF user of type 'Viewer' for the weights settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_CONDITIONS	Defines the access privilege of PF user of type 'Viewer' for the conditions settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFVIEWER_FOR_ OPTIONS_ARCHIVE	Defines the access privilege of PF user of type 'Viewer' for the archive settings of OPF Options tool. Typical val: 'X'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_SCENARIO	Defines the access privilege of PF user of type 'Operator' for the scenario settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_CONTROLS	Defines the access privilege of PF user of type 'Operator' for the controls settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_CONSTRAINTS	Defines the access privilege of PF user of type 'Operator' for the constraints settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_PARAMETERS	Defines the access privilege of PF user of type 'Operator' for the parameters settings of OPF Options tool. Typical val: 'RO'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_WEIGHTS	Defines the access privilege of PF user of type 'Operator' for the weights settings of OPF Options tool. Typical val: 'RO'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_CONDITIONS	Defines the access privilege of PF user of type 'Operator' for the conditions settings of OPF Options tool. Typical val: 'RO'
OPF_ACCESS_PRIVILEGE_PFOPERATOR_FOR_ OPTIONS_ARCHIVE	Defines the access privilege of PF user of type 'Operator' for the archive settings of OPF Options tool. Typical val: 'RO'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_SCENARIO	Defines the access privilege of PF user of type 'Admin' for the scenario settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_CONTROLS	Defines the access privilege of PF user of type 'Admin' for the controls settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_CONSTRAINTS	Defines the access privilege of PF user of type 'Admin' for the constraints settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_PARAMETERS	Defines the access privilege of PF user of type 'Admin' for the parameters settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_WEIGHTS	Defines the access privilege of PF user of type 'Admin' for the weights settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_CONDITIONS	Defines the access privilege of PF user of type 'Admin' for the conditions settings of OPF Options tool. Typical val: 'RW'
OPF_ACCESS_PRIVILEGE_PFADMIN_FOR_ OPTIONS_ARCHIVE	Defines the access privilege of PF user of type 'Admin' for the archive settings of OPF Options tool. Typical val: 'RW'
SS_ACT_KEY_ISOLATE_TAG	Defines the action key for the isolate tag operation. Typical val: 100

Parameters	Description
SS_ACT_KEY_ISOLATE_OPEN	Defines the action key for the isolate open operation. Typical val: 580
SS_ACT_KEY_RESTORE_OPEN	Defines the action key for the restore open operation. Typical val: 580
SS_ACT_KEY_RESTORE_CLOSE	Defines the action key for the restore close operation. Typical val: 260
SUB_SS_SAFETY_MARGIN	Tolerance margin for tie capacity calculation in substation suggested switching functionality. Typical val: 0.15
PROFILE_CACHING	0=Load interval data profiles are not cached in local memory at initialization and are instead built after every interval duration (hour), 1=profiles are cached only once at initialization and not at every interval. Typical val: 1
INTDATA_PARSER_1	Command name for program that extracts, transforms and loads load interval data from source to runtime tables. Typical val: 'prod_model_pf_hourly_load_profileloader_fromsql.ces'
INTDATA_PFPOST	PFPostProcessor procedure invocation for load interval data processing. Typical val: 'BEGIN pfpostprocessor.processIntervalData; END;'
INTDATA_NUMDAYTYPES	Defines number of day types for load interval data processing. Typical val: 2
INTDATA_NUMLOADPERIODS	Defines number of intervals in one day for load interval data processing. Typical val: 24
INTDATA_SEASON_REQD	Defines whether load interval data is season-dependent. 0=not dependent, 1=dependent. Typical val: 0
INTDATA_MONTH_REQD	Defines whether load interval data is month-dependent. 0=not dependent, 1=dependent. Typical val: 1
INTDATA_DAYTYPE_REQD	Defines whether load interval data is day-type-dependent. 0=not dependent, 1=dependent. Typical val: 1
INTDATA_KW_REQD	Defines whether load interval data is available for active power (kW). 0=not available, 1=available. Typical val: 1
INTDATA_KVAR_REQD	Defines whether load interval data is available for reactive power (kVAr). 0=not available, 1=available. Typical val: 0
INTDATA_KVA_REQD	Defines whether load interval data is available for apparent power (kVA). 0=not available, 1=available. Typical val: 0
INTDATA_PF_REQD	Defines whether load interval data is available for power factor. 0=not available, 1=available. Typical val: 0
_PF_MEAS_AMPS_SUM	Attribute number for current summation measurement of 3-ph circuit. Typical val: 1096
_PF_MEAS_AMPS	Attribute number for current measurement of 3-ph circuit. Typical val: 1012
_PF_MEAS_AMPS_A	Attribute number for current measurement of A-ph circuit. Typical val: 1013

Parameters	Description
_PF_MEAS_AMPS_B	Attribute number for current measurement of B-ph circuit. Typical val: 1014
_PF_MEAS_AMPS_C	Attribute number for current measurement of C-ph circuit. Typical val: 1015
_PF_MEAS_VA	Attribute number for apparent power of 3-ph circuit. Typical val: 1016
_PF_MEAS_VA_A	Attribute number for apparent power of A-ph circuit. Typical val: 1017
_PF_MEAS_VA_B	Attribute number for apparent power of B-ph circuit. Typical val: 1018
_PF_MEAS_VA_C	Attribute number for apparent power of C-ph circuit. Typical val: 1019
_PF_MEAS_KVA	Attribute number for apparent power (kVA) of 3-ph circuit. Typical val: 1020
_PF_MEAS_KVA_A	Attribute number for apparent power (kVA) of A-ph circuit. Typical val: 1021
_PF_MEAS_KVA_B	Attribute number for apparent power (kVA) of B-ph circuit. Typical val: 1022
_PF_MEAS_KVA_C	Attribute number for apparent power (kVA) of C-ph circuit. Typical val: 1023
_PF_MEAS_MVA	Attribute number for apparent power (MVA) of 3-ph circuit. Typical val: 1024
_PF_MEAS_MVA_A	Attribute number for apparent power (MVA) of A-ph circuit. Typical val: 1025
_PF_MEAS_MVA_B	Attribute number for apparent power (MVA) of B-ph circuit. Typical val: 1026
_PF_MEAS_MVA_C	Attribute number for apparent power (MVA) of C-ph circuit. Typical val: 1027
_PF_MEAS_VAR	Attribute number for reactive power of 3-ph circuit. Typical val: 1028
_PF_MEAS_VAR_A	Attribute number for reactive power of A-ph circuit. Typical val: 1029
_PF_MEAS_VAR_B	Attribute number for reactive power of B-ph circuit. Typical val: 1030
_PF_MEAS_VAR_C	Attribute number for reactive power of C-ph circuit. Typical val: 1031
_PF_MEAS_KVAR	Attribute number for reactive power (kVAr) of 3-ph circuit. Typical val: 1032
_PF_MEAS_KVAR_A	Attribute number for reactive power (kVAr) of A-ph circuit. Typical val: 1033

Parameters	Description
_PF_MEAS_KVAR_B	Attribute number for reactive power (kVAr) of B-ph circuit. Typical val: 1034
_PF_MEAS_KVAR_C	Attribute number for reactive power (kVAr) of C-ph circuit. Typical val: 1035
_PF_MEAS_MVAR	Attribute number for reactive power (MVAr) of 3-ph circuit. Typical val: 1036
_PF_MEAS_MVAR_A	Attribute number for reactive power (MVAr) of A-ph circuit. Typical val: 1037
_PF_MEAS_MVAR_B	Attribute number for reactive power (MVAr) of B-ph circuit. Typical val: 1038
_PF_MEAS_MVAR_C	Attribute number for reactive power (MVAr) of C-ph circuit. Typical val: 1039
_PF_MEAS_WATTS	Attribute number for active power of 3-ph circuit. Typical val: 1040
_PF_MEAS_WATTS_A	Attribute number for active power of A-ph circuit. Typical val: 1041
_PF_MEAS_WATTS_B	Attribute number for active power of B-ph circuit. Typical val: 1042
_PF_MEAS_WATTS_C	Attribute number for active power of C-ph circuit. Typical val: 1043
_PF_MEAS_KW	Attribute number for active power (kW) of 3-ph circuit. Typical val: 1044
_PF_MEAS_KW_A	Attribute number for active power (kW) of A-ph circuit. Typical val: 1045
_PF_MEAS_KW_B	Attribute number for active power (kW) of B-ph circuit. Typical val: 1046
_PF_MEAS_KW_C	Attribute number for active power (kW) of C-ph circuit. Typical val: 1047
_PF_MEAS_MW	Attribute number for active power (MW) of 3-ph circuit. Typical val: 1048
_PF_MEAS_MW_A	Attribute number for active power (MW) of A-ph circuit. Typical val: 1049
_PF_MEAS_MW_B	Attribute number for active power (MW) of B-ph circuit. Typical val: 1050
_PF_MEAS_MW_C	Attribute number for active power (MW) of C-ph circuit. Typical val: 1051
_PF_MEAS_VAH	Attribute number for apparent energy of 3-ph circuit. Typical val: 1052
_PF_MEAS_VAH_A	Attribute number for apparent energy of A-ph circuit. Typical val: 1053

Parameters	Description
_PF_MEAS_VAH_B	Attribute number for apparent energy of B-ph circuit. Typical val: 1054
_PF_MEAS_VAH_C	Attribute number for apparent energy of C-ph circuit. Typical val: 1055
_PF_MEAS_KVAH	Attribute number for apparent energy (kVAh) of 3-ph circuit. Typical val: 1056
_PF_MEAS_KVAH_A	Attribute number for apparent energy (kVAh) of A-ph circuit. Typical val: 1057
_PF_MEAS_KVAH_B	Attribute number for apparent energy (kVAh) of B-ph circuit. Typical val: 1058
_PF_MEAS_KVAH_C	Attribute number for apparent energy (kVAh) of C-ph circuit. Typical val: 1059
_PF_MEAS_MVAH	Attribute number for apparent energy (MVAh) of 3-ph circuit. Typical val: 1060
_PF_MEAS_MVAH_A	Attribute number for apparent energy (MVAh) of A-ph circuit. Typical val: 1061
_PF_MEAS_MVAH_B	Attribute number for apparent energy (MVAh) of B-ph circuit. Typical val: 1062
_PF_MEAS_MVAH_C	Attribute number for apparent energy (MVAh) of C-ph circuit. Typical val: 1063
_PF_MEAS_VARH	Attribute number for reactive energy of 3-ph circuit. Typical val: 1064
_PF_MEAS_VARH_A	Attribute number for reactive energy of A-ph circuit. Typical val: 1065
_PF_MEAS_VARH_B	Attribute number for reactive energy of B-ph circuit. Typical val: 1066
_PF_MEAS_VARH_C	Attribute number for reactive energy of C-ph circuit. Typical val: 1067
_PF_MEAS_KVARH	Attribute number for reactive energy (kVARh) of 3-ph circuit. Typical val: 1068
_PF_MEAS_KVARH_A	Attribute number for reactive energy (kVARh) of A-ph circuit. Typical val: 1069
_PF_MEAS_KVARH_B	Attribute number for reactive energy (kVARh) of B-ph circuit. Typical val: 1070
_PF_MEAS_KVARH_C	Attribute number for reactive energy (kVARh) of C-ph circuit. Typical val: 1071
_PF_MEAS_MVARH	Attribute number for reactive energy (MVARh) of 3-ph circuit. Typical val: 1072
_PF_MEAS_MVARH_A	Attribute number for reactive energy (MVARh) of A-ph circuit. Typical val: 1073

Parameters	Description
_PF_MEAS_MVARH_B	Attribute number for reactive energy (MVARh) of B-ph circuit. Typical val: 1074
_PF_MEAS_MVARH_C	Attribute number for reactive energy (MVARh) of C-ph circuit. Typical val: 1075
_PF_MEAS_WH	Attribute number for active energy of 3-ph circuit. Typical val: 1076
_PF_MEAS_WH_A	Attribute number for active energy of A-ph circuit. Typical val: 1077
_PF_MEAS_WH_B	Attribute number for active energy of B-ph circuit. Typical val: 1078
_PF_MEAS_WH_C	Attribute number for active energy of C-ph circuit. Typical val: 1079
_PF_MEAS_KWH	Attribute number for active energy (kWh) of 3-ph circuit. Typical val: 1080
_PF_MEAS_KWH_A	Attribute number for active energy (kWh) of A-ph circuit. Typical val: 1081
_PF_MEAS_KWH_B	Attribute number for active energy (kWh) of B-ph circuit. Typical val: 1082
_PF_MEAS_KWH_C	Attribute number for active energy (kWh) of C-ph circuit. Typical val: 1083
_PF_MEAS_MWH	Attribute number for active energy (MWh) of 3-ph circuit. Typical val: 1084
_PF_MEAS_MWH_A	Attribute number for active energy (MWh) of A-ph circuit. Typical val: 1085
_PF_MEAS_MWH_B	Attribute number for active energy (MWh) of B-ph circuit. Typical val: 1086
_PF_MEAS_MWH_C	Attribute number for active energy (MWh) of C-ph circuit. Typical val: 1087
_PF_MEAS_PF	Attribute number for power factor. Typical val: 1088
_PF_MEAS_PF_A	Attribute number for power factor of A-ph circuit. Typical val: 1089
_PF_MEAS_PF_B	Attribute number for power factor of B-ph circuit. Typical val: 1090
_PF_MEAS_PF_C	Attribute number for power factor of C-ph circuit. Typical val: 1091
_PF_MEAS_PHASE_ANGLE	Attribute number for phase angle of 3-ph circuit. Typical val: 1092
_PF_MEAS_PHASE_ANGLE_A	Attribute number for phase angle of A-ph circuit. Typical val: 1093
_PF_MEAS_PHASE_ANGLE_B	Attribute number for phase angle of B-ph circuit. Typical val: 1094

Parameters	Description
_PF_MEAS_PHASE_ANGLE_C	Attribute number for phase angle of C-ph circuit. Typical val: 1095
_PF_MEAS_PHASE_VOLTS	Attribute number for phase voltage of 3-ph circuit. Typical val: 1100
_PF_MEAS_PHASE_VOLTS_A	Attribute number for phase voltage of A-ph circuit. Typical val: 1101
_PF_MEAS_PHASE_VOLTS_B	Attribute number for phase voltage of B-ph circuit. Typical val: 1102
_PF_MEAS_PHASE_VOLTS_C	Attribute number for phase voltage of C-ph circuit. Typical val: 1103
_PF_MEAS_PHASE_KV	Attribute number for phase voltage (kV) of 3-ph circuit. Typical val: 1104
_PF_MEAS_PHASE_KV_A	Attribute number for phase voltage (kV) of A-ph circuit. Typical val: 1105
_PF_MEAS_PHASE_KV_B	Attribute number for phase voltage (kV) of B-ph circuit. Typical val: 1106
_PF_MEAS_PHASE_KV_C	Attribute number for phase voltage (kV) of C-ph circuit. Typical val: 1107
_PF_MEAS_PHASE_MV	Attribute number for phase voltage (MV) of 3-ph circuit. Typical val: 1108
_PF_MEAS_PHASE_MV_A	Attribute number for phase voltage (MV) of A-ph circuit. Typical val: 1109
_PF_MEAS_PHASE_MV_B	Attribute number for phase voltage (MV) of B-ph circuit. Typical val: 1110
_PF_MEAS_PHASE_MV_C	Attribute number for phase voltage (MV) of C-ph circuit. Typical val: 1111
_PF_MEAS_LINE_VOLTS	Attribute number for line voltage of 3-ph circuit. Typical val: 1200
_PF_MEAS_LINE_VOLTS_A	Attribute number for line voltage of A-ph circuit. Typical val: 1201
_PF_MEAS_LINE_VOLTS_B	Attribute number for line voltage of B-ph circuit. Typical val: 1202
_PF_MEAS_LINE_VOLTS_C	Attribute number for line voltage of C-ph circuit. Typical val: 1203
_PF_MEAS_LINE_KV	Attribute number for line voltage (kV) of 3-ph circuit. Typical val: 1204
_PF_MEAS_LINE_KV_A	Attribute number for line voltage (kV) of A-ph circuit. Typical val: 1205
_PF_MEAS_LINE_KV_B	Attribute number for line voltage (kV) of B-ph circuit. Typical val: 1206

Parameters	Description
_PF_MEAS_LINE_KV_C	Attribute number for line voltage (kV) of C-ph circuit. Typical val: 1207
_PF_MEAS_LINE_MV	Attribute number for line voltage (MV) of 3-ph circuit. Typical val: 1208
_PF_MEAS_LINE_MV_A	Attribute number for line voltage (MV) of A-ph circuit. Typical val: 1209
_PF_MEAS_LINE_MV_B	Attribute number for line voltage (MV) of B-ph circuit. Typical val: 1210
_PF_MEAS_LINE_MV_C	Attribute number for line voltage (MV) of C-ph circuit. Typical val: 1211
_PF_MEAS_CAP_POSITION	Attribute number for capacitor position of 3-ph circuit. Typical val: 1800
_PF_MEAS_CAP_POSITION_A	Attribute number for capacitor position of A-ph circuit. Typical val: 1801
_PF_MEAS_CAP_POSITION_B	Attribute number for capacitor position of B-ph circuit. Typical val: 1802
_PF_MEAS_CAP_POSITION_C	Attribute number for capacitor position of C-ph circuit. Typical val: 1803
_PF_MEAS_TAP_POSITION	Attribute number for tap position of 3-ph circuit. Typical val: 1807
_PF_MEAS_TAP_POSITION_A	Attribute number for tap position of A-ph circuit. Typical val: 1804
_PF_MEAS_TAP_POSITION_B	Attribute number for tap position of B-ph circuit. Typical val: 1805
_PF_MEAS_TAP_POSITION_C	Attribute number for tap position of C-ph circuit. Typical val: 1806
_PF_MEAS_CAP_REG	Attribute number for capacitor regulation status. Typical val: 1851
_PF_MEAS_TAP_REG	Attribute number for tap regulation status. Typical val: 1852
_PF_MEAS_TAP_PRI	Attribute number for tap position of primary. Typical val: 1853
_PF_MEAS_TAP_SEC	Attribute number for tap position of secondary. Typical val: 1854
_PF_MEAS_TAP_TER	Attribute number for tap position of tertiary. Typical val: 1855
_PF_MEAS_NUM_SEQ_CAP	Attribute number for num sequential capacitors. Typical val: 1856
_PF_MEAS_TEMPERATURE	Attribute number for temperature. Typical val: 1900
USED_MEASUREMENT_SET	Defines measurement sets to be used in the load allocation function. 0=feeder head only, 1=all measurement points. Typical val: 1

Parameters	Description
SCALING_APPLIED_AREA	Defines the area for chosen load scaling type and used measurement set settings. 0=selected feeder, 1=selected island, 2=system-wide. Typical val: 1
METERING_PERIOD_HOURS	Defines length of metering period in hours to convert energy measurements into other units. Typical val: 1.0
LOAD_ALLOCATION_TOLERANCE_PERCENTAGE	Maximum mismatch tolerance between solved and measured values. Typical val: 1.0
LOAD_ALLOCATION_MAX_ITERATION	Maximum number of iterations within which the load allocation function attempts to match up solved and measured values within the set tolerance. Typical val: 5
LOAD_ALLOCATION_TRIGGER	0=Load allocation triggered for every new powerflow solution request, 1=load allocation triggered for an island that has received measurement updates. Typical val: 1
PFOPTION_BUTTON_ON_COLOR	Display color for the PFOption tool button when its status is on. Typical val: Green
PFOPTION_BUTTON_OFF_COLOR	Display color for the PFOption tool button when its status is off. Typical val: Grey
PFOPTION_BUTTON_TRIGGER_ON_TEXT	Display string for the PFOption tool Triggers button when its status is on. Typical val: On
PFOPTION_BUTTON_TRIGGER_OFF_TEXT	Display string for the PFOption tool Triggers button when its status is off. Typical val: Off
PFOPTION_BUTTON_CONTROLS_ON_TEXT	Display string for the PFOption tool Controls button when its status is on. Typical val: On
PFOPTION_BUTTON_CONTROLS_OFF_TEXT	Display string for the PFOption tool Controls button when its status is off. Typical val: Off
PFOPTION_BUTTON_LIMITS_ON_TEXT	Display string for the PFOption tool Limits button when its status is on. Typical val: Enabled
PFOPTION_BUTTON_LIMITS_OFF_TEXT	Display string for the PFOption tool Limits button when its status is off. Typical val: Disabled
PFOPTION_BUTTON_MULTLOAD_ON_TEXT	Display string for the PFOption tool MultLoad button when its status is on. Typical val: Enabled
PFOPTION_BUTTON_MULTLOAD_OFF_TEXT	Display string for the PFOption tool MultLoad button when its status is off. Typical val: Disabled
PFOPTION_BUTTON_LOADDIST_ON_TEXT	Display string for the PFOption tool LoadDist button when its status is on. Typical val: Enabled
PFOPTION_BUTTON_LOADDIST_OFF_TEXT	Display string for the PFOption tool LoadDist button when its status is off. Typical val: Disabled
PFOPTION_BUTTON_VOLTDISP_ON_TEXT	Display string for the PFOption tool VolDisp button when its status is on. Typical val: Enabled
PFOPTION_BUTTON_VOLTDISP_OFF_TEXT	Display string for the PFOption tool VolDisp button when its status is off. Typical val: Disabled

Parameters	Description
SS_TABLE_SW_ON_TEXT	Display string for the Switch Status button of Load Restoration tool when its status is on. Typical val: Enabled
SS_TABLE_SW_OFF_TEXT	Display string for the Switch Status button of Load Restoration tool when its status is off. Typical val: Disabled
SS_TABLE_SW_ON_COLOR	Display color for the Switch Status button of Load Restoration tool when its status is on. Typical val: Green
SS_TABLE_SW_OFF_COLOR	Display color for the Switch Status button of Load Restoration tool when its status is off. Typical val: Grey
PF_TOOL_EDITABLE_COLUMN_COLOR	Display color for the editable column of power flow tools. Typical Val: Pink
PF_TOOL_AUTO_ON_COLOR	Display color for the AUTO control button of power flow tools when its status is on. Typical Val: Green
PF_TOOL_AUTO_OFF_COLOR	Display color for the AUTO control button of power flow tools when its status is off. Typical Val:Grey
PF_TOOL_AUTO_ON_TEXT	Display string for the AUTO control button of power flow tools when its status is on. Typical Val:Auto On
PF_TOOL_AUTO_OFF_TEXT	Display string for the AUTO control button of power flow tools when its status is off. Typical Val:Auto Off
MAX_IGNORE_AMP	Maximum ignored current value. The current is displayed as zero, if it is less than the value. Typical val: 0.01
MAX_IGNORE_VOL	Maximum ignored voltage value. The voltage is displayed as zero, if it is less than the value. Typical val: 0.01
CHECK_ENERGIZED_AMP	The current tolerance of energized device. Typical Val: 1.0
DISABLE_FLOW_DIRECTION	0-enable the flow direction check, 1-disable the flow direction check
MIN_LINE_LENGTH	Minimum line length. If the length is less that the value, set its length to minimum value. Typical val: 10.0
MAX_REALTIME_PFOBJECT	The maximum number of real-time power flow objects. Typical val: 10
MAX_STUDYMODE_PFOBJECT	The maximum number of study mode power flow objects. Typical val: 10
SOLUTION_STATUS_COMPLETED_SOLUTION	Display string for power flow solution status if power flow is solved. Typical val: Power Flow Solved
SOLUTION_STATUS_PENDING_SOLUTION	Display string for power flow solution status if power flow is not solved. Typical val: Power Flow Not Solved
SOLUTION_STATUS_INACCURATE_SOLUTION	Display string for power flow solution status if power flow solved is solved without sufficient accuracy. Typical val: Inaccurate Power Flow Solution
SOLUTION_STATUS_BAD_GEN_MODEL	Display string for power flow solution status if the gen models are bad. Typical val: Bad Gen Topology Model

Parameters	Description
SOLUTION_STATUS_BAD_LOAD_MODEL	Display string for power flow solution status if the load models are bad. Typical val: Bad Load Topology Model
SOLUTION_STATUS_BAD_LINE_MODEL	Display string for power flow solution status if the line models are bad. Typical val: Bad Line Topology Model
SOLUTION_STATUS_BAD_SHUNT_MODEL	Display string for power flow solution status if the shunt models are bad. Typical val: Bad Shunt Topology Model
SOLUTION_STATUS_BAD_XFMR_MODEL	Display string for power flow solution status if the transformer models are bad. Typical val: Bad Xfmr Topology Model
SOLUTION_STATUS_BAD_BUS_MODEL	Display string for power flow solution status if the bus models are bad. Typical val: Bad Bus Topology Model
SOLUTION_STATUS_BAD_LINK_MODEL	Display string for power flow solution status if the link models are bad. Typical val: Bad Link Topology Model
SOLUTION_STATUS_BAD_SYSTEM_BASE	Display string for power flow solution status if the system base is bad. Typical val: Bad System Base
SOLUTION_STATUS_BAD_PHASE_BUSIO	Display string for power flow solution status if failed to create BusIO object. Typical val: Bad Bus Topology Model
SOLUTION_STATUS_BAD_XFMR_TAPIO	Display string for power flow solution status if failed to create Xfmr TapIO object. Typical val: Bad Xfmr Topology Model
SOLUTION_STATUS_INIT_BUS_ERROR	Display string for power flow solution status if failed to initialize the buses. Typical val: Bad Bus Topology Model
SOLUTION_STATUS_INIT_BRANCH_ERROR	Display string for power flow solution status if failed to initialize the branches. Typical val: Bad Line/Link Topology Model
SOLUTION_STATUS_ISLAND_COUPLING	Display string for power flow solution status if there are device couplings between two islands. Typical val: Assign Two Islands to One Device
ISLAND_COUPLING_CHECK	0-Don't do island coupling check, 1-do the island coupling check
FLM_DAILY_PEAK_HOUR	The daily peak hour used by feeder load management. Typical val: 1
FLM_FEEDER_BREAKER_VIOLATION_WEIGHT	Weight factor for feeder breaker violation. Typical val: 0.2
FLM_LOAD_IMBALANCE_VIOLATION_WEIGHT	Weight factor for load imbalance violation. Typical val: 0.0
FLM_CONDUCTOR_VIOLATION_WEIGHT	Weight factor for conductor violation. Typical val: 0.2
FLM_POWER_TRANSFORMER_VIOLATION_ WEIGHT	Weight factor for power transformer violation. Typical val: 0.2
FLM_SWITCHES_VIOLATION_WEIGHT	Weight factor for switch violation. Typical val: 0.1
FLM_UNDER_VOLT_VIOLATION_WEIGHT	Weight factor for voltage underlimit violation. Typical val:0.1
FLM_OVER_VOLT_VIOLATION_WEIGHT	Weight factor for voltage overlimit violation. Typical val:0.1
FLM_DISTIRBUTION_TRANSFORMER_VIOLATION _WEIGHT	Weight factor for distribution transformer violation. Typical val: 0.1

Parameters	Description
FLM_BACKFEED_VIOLATION_WEIGHT	Weight factor for backfeed violation. Typical val: 0.0
FLM_WARNING_THRESHOLD	Warning Threshold for feeder load management. Typical Val: 0.8
FLM_WARNING_WEIGHT	Weight factor for warning in feeder load management. Typical val: 0.05
FLA_PICKUP_SCALE_FACTOR	Scale factor of pickup current used in fault location analysis. Typical val: 1.2
UPDATE_PF_SOLUTION	0-don't automatically update the selected device's power flow solution for device status changes. 1-automatically update the selected device's power flow solution for any status change if the power flow tool opens.

Chapter 15

Java Application Configuration

The intended audience for this chapter is the system administrators responsible for making customer specific configuration changes to Oracle Utilities Network Management System java applications. This chapter includes the following topics:

- **Overview**
- **Making Changes to Java Application Configuration**
- **Deploying Configuration Changes**

Overview

The Oracle Utilities Network Management System java applications are configured by using the standard product configuration with overrides that are specific to a customer. This chapter describes where the java application configuration files reside as well as how to update and deploy changes to these files to an Oracle Utilities Network Management System Web Gateway.

Making Changes to Java Application Configuration

After executing the installation procedures outlined in the Oracle Utilities Network Management System Installation Guide, the product configuration files for all java applications will be stored in ${NMS_HOME}/java/product. To make a change to any java configuration file, you will need to copy the file that needs to be changed to ${NMS_CONFIG}/jconfig, using the same directory structure as it exists in the product directory. For example, to change the MWMInterface.properties file, copy that file from ${NMS_HOME}/java/product/server to ${NMS_CONFIG}/jconfig/server. Make the customer specific changes on the copied version. Do not change the product version.

If a java property file is changed, it is only necessary for the customer version to include the changed lines. Any line in a property file that is not overridden by a customer-specific line will use the product value. Other file types, such as xml documents, need to be complete replacements of the product versions.

Deploying Configuration Changes

These steps are required after changes have been made to a customer's java application configuration after the initial installation of the Oracle Utilities Network Management System.

The ${NMS_CONFIG}/jconfig/build.properties file contains various properties that control the configuration build process. The following is a list of the commonly modified values:

project.name	The name of the project/customer. This is displayed in the Help About dialog of any Java GUI applications to identify the application as being configured for this particular customer.
project.tag	This is a CVS tag or other identifier used to identify a particular build of the customer-specific configuration. This is also displayed on the Help About dialog of any Java GUI applications to identify a customer-specific configuration deployment.
dir.localization	If the configuration is based off of a localized (non-English language) version, enter the directory of the localization configuration. Otherwise leave this commented out.
dir.config.deploy	This is the directory where runtime configuration jar files will be created. The default is a staging area ($NMS_HOME/java/deploy), but it is also possible to configure these runtime files to be deployed directly to the application server. Uncomment and update the JBoss or Weblogic sections if this is desired.

After making customer specific changes to the java application configuration files and also setting up the build.properties file for your environment, create the runtime configuration jar files by running the following command:

```
nms-install-config --java
```

This will create the nms_config.jar (client configuration) and nms_server_config.jar (server configuration) runtime files and store them in the directory defined in "dir.config.deploy" variable from the build.properties file described above. If these runtime files are configured to be deployed to a staging area, then they will need to be copied to the appropriate directory for the java application server (JBoss or WebLogic) to deploy them. See instructions below for deploying the changed configuration to your specific Java Application server.

Deploying to JBoss Application Server

1. Log in as the user account that will run the JBoss Application Server.
2. Set the JBOSS_HOME environment variable. For example:

   ```
   $ export JBOSS_HOME=/opt/jboss/jboss-4.2.3.GA
   ```

3. Change directory to the $JBOSS_HOME/bin directory:

   ```
   $ cd $JBOSS_HOME/bin
   ```

4. Stop JBoss Application Server (if running). For example:

   ```
   $ ./shutdown.sh -S
   ```

5. If you have saved the runtime files in a staging area, copy the nms_server_config.jar and the nms_config.jar to the lib directory as follows:

   ```
   $ cp {dir.config.deploy}/*config.jar $JBOSS_HOME/server/default/lib
   where "dir.config.deploy" is the value found in build.properties
   ```

6. Start the JBoss Application Server:

   ```
   $ ./run.sh --host hostname
   ```

7. Open a browser and navigate to `http://hostname:8080/nms`.

   Here *hostname* represents the DNS name or IP address of the JBoss Application Server.

Deploying to WebLogic Application Server

1. Login in as the user account that will run the WebLogic Application Server
2. If you have saved the runtime files in a staging area, copy the nms_server_config.jar and the nms_config.jar to the lib directory as follows:

   cp *dir.config.deploy*/*config.jar *BEA_HOME*/user_projects/domains/*DOMAIN_NAME*/lib

   where "*dir.config.deploy*" is the value found in build.properties.

   "*BEA_HOME*" represents the directory where WebLogic is installed.

   "*DOMAIN_NAME*" represents the WebLogic domain root directory where Oracle Utilities Network Management System will run.

3. A restart of the WebLogic-managed server that will be running Oracle Utilities Network Management System is required for these changes to take effect.

Chapter 16

Building Custom Applications

The intended audience for this chapter are software programmers responsible for building interfaces and applications that interact with the Oracle Utilities Network Management System. This chapter includes the following topics:

- **Overview**
- **Prerequisites**
- **Compiling using the Software Development Kit**

Overview

This chapter describes how to build C++ applications that interact with the Oracle Utilities Network Management System using the Oracle Utilities Network Management System Software Development Kit (SDK).

Most Oracle Utilities Network Management System implementations will require at least one custom built application, a model interface, while other implementations may have addition interfaces and other programs that interact with the Oracle Utilities Network Management System. To support the implementation of these interfaces and programs, the Oracle Utilities Network Management System has provided a Software Development Kit. The Software Development Kit is installed into the $CES_HOME/build directory and is pointed to using the .nmsrc environment variable $NMS_BUILD.

There are two subcomponents to the Software Development Kit:

$NMS_BUILD/make	The make rules to support the architecture and platform configuration.
$NMS_BUILD/include	The C++ header files required to interact with the Oracle Utilities Network Management System.

Note the following regarding usage of the Oracle Utilities Network Management System Software Development Kit:

- The SDK interfaces are not documented and are for use as-is.
- The SDK interfaces may change from release to release with no guarantees of forward or backward compatibility.
- The use of the SDK can impact the running Oracle Utilities Network Management System based on what is programmed with the SDK. Impacts may include performance

issues, system lock ups, system instability, data loss, and changes to system functionality. It is recommend that you heavily test any interfaces or programs you create and judge the impact on the Oracle Utilities Network Management System and understand these interfaces and programs should be considered "use at your own risk".

- The SDK may not be used to reverse engineer the features and functionality of the Oracle Utilities Network Management System.

Prerequisites

In addition to the prerequisites required to run the Oracle Utilities Network Management System, the following are required to use the Oracle Utilities Network Management System Software Development Kit:

GNU Make	Download the latest version via the http://www.gnu.org website.	
Platform-specific C++ compiler	**Platform**	**C++ Compiler Version**
	Oracle Enterprise Linux 5 Red Hat Enterprise Linux 5	gcc 4.1.1
	SUSE Linux	gcc 4.1.2
	SUNOS 5	Sun Studio 11
	AIX 6.1	Visual Age 10.1
	AIX 5.3	Visual Age 8.0
	HP Itanum 2	ANSI C++ A.06.15
	HP PA-RISC 2	ANSI C++ A.03.73

Verify that your .nmsrc was generated using the template from $CES_HOME/templates/nmsrc.template and that the environment variable $NMS_BUILD is set to $CES_HOME/build.

Compiling using the Software Development Kit

Place the C++ source code to build the custom interface or program in a subdirectory of the $NMS_CONFIG directory, typically $NMS_CONFIG/apps. The executables resulting from the compile will be generated into the $NMS_CONFIG/bin directory via the Makefile so the nms-install-config process can copy them to the runtime directory, $NMS_HOME/bin. If you create custom shared libraries, these need to be copied into $NMS_CONFIG/lib so they also are available for nms-install-config to copy them to the runtime directory, $NMS_HOME/lib.

The following is an example Makefile for the $NMS_CONFIG/apps directory:

```
#########################################################################
#
# Example $NMS_CONFIG/apps directory Makefile
#
#########################################################################
# Include compiler and architecture dependent Makefile parameters.
HAS_GUI = YES
include $(NMS_BUILD)/make/make.rules
LOCALLIBS = $(PP_LIB) $(MV_LIB) $(SUPPORT_LIBS) $(MB_LIB) $(GRWINDOW_LI

# Source for all run-time applications
SOURCES = \
        CustomInterface.C
OBJECTS = $(SOURCES:.C=.$(OBJ_EXT))
PROGRAM = CustomInterface$(EXE_EXT)

#########################################################################
# Targets

include $(SIMPLE_PROGRAM_MAKE)

all:: $(PROGRAM)
        @ if [ ! -d "$(NMS_CONFIG)/bin" ]; then \
            mkdir $(NMS_CONFIG)/bin; \
          fi
        cp $(PROGRAM) $(NMS_CONFIG)/bin;
```

The target executable file in this example is `CustomInterface` and the C++ source code to compile is `CustomInterface.C`.

From the command prompt within the $NMS_CONFIG/apps directory, build the custom program with "make clean" to remove old compiled binaries and "make" to compile and install the binaries into the $NMS_CONFIG/bin directory. Below is an example of what the output from the make system will look like as a result or running these two commands.

```
test-system:testuser> make clean
rm -f *.o *~ core .pure* gmon.out  so_locations *.sl *.so *.a
rm -f \#* 3log *.third *.third.*
rm -rf ptrepository cxx_repository Templates.DB SunWS_cache tempinc
rm -f OracleLite_preprocessor
test-system:testuser> make

OracleLite_preprocessor.o
--
motif Building OracleLite_preprocessor:
/users/testuser/nms/product/1.9.0.1/build/make/aix_shr xlC_r -q64 -g  ·
qcheck=nobounds:div:null -bbigtoc -qrtti=all -qnamemangling=v5 -qeh=v6
-DDIFFUSION_NOTIFIES  -DAIXRS -D_ALL_SOURCE -D_REENTRANT -DP_THREADS  ·
DHAS_XT -DFAULT_RESTORATION -DGSOAP_VERSION=0x0207090f -I/users/
testuser/nms/product/1.9.0.1/build/include   -I/users/testuser/isis/
include -I/usr/include/Motif2.1/Xm -I/usr/include/X11 -DXTFUNCPROTO= -
I/opt/local-9.1/boost  -I/opt/oms-9.1/include -brtl  -L/users/testuser,
blib  -o OracleLite_preprocessor OracleLite_preprocessor.o  -lPp  -lMv
-lMv  -lApp -L/opt/oms-9.1/lib -lxrttable -lxrttablestub -lpdsutil -
lXpm  -lCrew  -lPp  -lService  -lMB  -lGrWindow   -lintersys_xt  -
lWrapper  -lBase  -lfoss -L/users/testuser/blib -L/users/testuser/isis,
lib -lisisX -lmds -lisis -lisis_task_native  -lCmdLine  -L/usr/lpp/X11,
lib/R6/Motif2.1 -lMrm -lXm  -lXt -lX11 -lXext    -L/opt/oms-9.1/lib  -
lgsoap++  -lgsoap -L/opt/oms-9.1/lib  -lcurl
echo " "

cp OracleLite_preprocessor /users/testuser/prod_model/bin;
test-system:testuser>
```

After you have successfully compiled the custom application, run `nms-install-config` to pick up the executables from the $NMS_CONFIG/bin and install them into $NMS_HOME/bin.

Index

D

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W