Installation & Migration Guide

Release 5.5.1

Model N

THE LEADER IN REVENUE MANAGEMENT

Model N® Revenue Management Application, Release 5.5.1

Trademarks

Document Revised: March 17, 2011

Contact Model N
Model N
1800 Bridge Parkway
Redwood City, CA 94065

Phone: (650) 610 - 4600
FAX: (650) 610 - 4699

www.modeln.com

Table of Contents

List of Figures

List of Tables

List of Code Examples

1 About This Guide

1.1 Overview

This administrator's guide provides information on the following topics:

- what to install to get the Model N application up and running
- how to deploy the Model N application
- how to configure the additional software needed to run Model N
- how to monitor the Model N application once it is deployed
- how to configure the application to meet your business needs

1.2 Audience

This guide is intended for the following audiences:

- System administrators
- Model N Professional Services representatives who are planning the installation of your Model N system

1.3 Using This Guide

This guide is organized based on the typical order in which you would install Model N and the software needed to run it as well as monitor, and configure the Model N application.

Caution: Never directly modify the ACF component .xml files. Add an ACF component overlay that extends any particular component and perform your modifications there.

1.4 Conventions

The following table describes the conventions used in this guide.

Table 1-1: Conventions

Conventions	Descriptions
File names and code objects	Names of files and code objects are in `Courier New` font style.
System outputs	System outputs, such as confirmation messages our alerts, are first letter capitalized. For example: The system displays the following message: Do you want to save your changes before moving on?
Referenced topics	Referenced topic headings are in blue and underlined. For example: See Deployment for more information.
Referenced guides or documents	Referenced guide or document titles are in *italics*. For example: See *Medicaid Developer's Guide* for more information.
Lifecycle statuses	The lifecycle statuses are in `Courier New` font style. For example: Users cannot override the `Ignore` status or the `Fatal` severity level.
Notes	Notes are contained within two horizontal lines. For example: **Note:** This is a note.
Blank pages	There will be a blank page with header and footer at the end of the chapters ending on odd page numbers. This format is designed to accommodate printing multiple chapters on duplex paper and keeping the collation even.

1.5 Related Documents/Resources

The following related documents are available:

- *Reporting Guide*
- *Supported Configurations Matrix*
- *Release Notes*

1.6 Feedback

Model N welcomes your comments and suggestions on the quality and usefulness of this guide. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most?

If you find any errors or have any other suggestions for improvement, please indicate the chapter, section, and page number (if available). You can send comments to us in the following ways:

- E-mail:techdocs@modeln.com
- FAX: (650) 610 - 4699. Attn: Product Documentation
- Postal service:

  Model N

  Product Documentation

  1800 Bridge Parkway,

  Redwood City, CA 94065

At Model N, we are dedicated to improving the technical documentation. We are not able to reply to your comments individually, but we use your advice to improve our services.

2 Introduction

This chapter covers commonly referenced deployment values.

For information on Model N supported configurations, see the *Supported Configurations Matrix*.

2.1 Deployment Values

Throughout this guide, references are made to values that may vary over deployments. These deployment-specific values are indicated in the text and examples by variable names enclosed in angled brackets (< and >).

The following table provides a list of the most commonly referenced values. You should replace deployment-specific values indicated by variable names with the value chosen for the current deployment.

Table 2-1: Deployment Values

Name	Description	Example
<AppHost>	The host name or network address of the server on which the application server is installed. This setting is required when configuring the Model N system to connect to resources provided by the application server.	app01-mnprd.company.com
<AppName>	An identifier that is chosen to represent the application. This identifier is often associated with the company name or project. For example, if the project goes by the name Model N and the version being deployed is 5.3, you could choose an application name such as ModelN or MN53. **Note:** This must match the key value in the EAR file. The default value that Model N ships for the 5.4 pharma release is `mn54ph` and for the 5.4 med dev release is `mn54md`. If you've changed this value, make sure that the `<AppName>` value matches it.	mn54ph or mn54md
<BaseProps>	The base property file is used to configure the web application on start-up. When the application initializes, the default application settings are loaded into memory from this property file or from any property files that are included by the base property file. Any default configuration properties can be overridden by placing the property overrides later in the file than the default property or its included property file.	app01_modeln

Table 2-1: Deployment Values (Continued)

Name	Description	Example
<CognosInstall>	The Cognos installation directory which is by default `/opt/cognos.crn`. Indicate this value if you use another directory to install Cognos.	/opt/cognos.crn
<ContextRoot>	The application context root refers to the servlet context that is used by an application server to identify the application. For example, a context root located at `http://www.mycorp.com/modeln` would be used to route all requests for `/modeln` to the appropriate web application on the server. The context path allows multiple web applications to reside on the same server.	modeln
<DBHost>	The host name or network address of the database server on which the Model N database resides. This value is used when configuring the application or creating connection pools within the application server.	db01-mnprd.company.com
<DBInstance>	The name of the database, within the database server, where the Model N data resides.	MNPRDSID
<DBPass>	The application database password used when authenticating against the database using the `<DBUser>` account.	secret
<DBSysPass>	The database system password used when authenticating against the database using the system account.	ssecret
<DBUser>	The account that the Model N system uses to connect to the database. This account is different than the database administration account which is required to create the user.	modeln

Table 2-1: Deployment Values (Continued)

Name	Description	Example
<ExtDir>	The external directory for log files, data flows, property files and other host-specific data. See Application Server Support Files for more information.	/opt/mn/app01
<FTPHost>	The hostname or the network address of the FTP server that the Direct Loader uses to transfer data files between the application server and the database server.	ftp01-mnprd.company.com
<FTPPass>	The password for `<FTPUser>`.	secret
<FTPPort>	The port on which the `<FTPHost>` is listening for incoming FTP requests. The default value for the port is 21.	21
<FTPUser>	The user account that is used to connect to the FTP server.	mnprd
<HttpServerHost>	The hostname or network address of the web server that is used to access static application content and to forward application requests to the application server.	web01-mnprd.company.com
<JBossHome>	The directory where JBoss Enterprise Application Platform is installed.	C:\Program Files\EnterprisePlatform-4.3.0.GA_CP08 on Windows; <homedir>/EnterprisePlatform-4.3.0.GA_CP08 on Linux
<JBossPort>	The port on which the JBoss server will listen for incoming connections. This is the JBoss port specified when creating the JBoss server configuration using ConfigureJBoss ant target. The port that is specified in the `local_jboss.properties` file must match.	7001

Table 2-1: Deployment Values (Continued)

Name	Description	Example
<TNSPort>	The port on which the `<DBHost>` is listening for incoming connections.	1521
<WebLogicPort>	The port on which the WebLogic server is listening for incoming connections. This is the "Listen Port" or the "SSL Listen Port" if SSL is enabled, for the WebLogic server instance running the Model N application. The port that is specified in the specified in the `local_weblogic.properties` file must match this value.	7001

3

Deployment Overview: Steps, Tools, and Files

The Model N application is deployed as a multi-tiered Java enterprise application that is composed of an HTTP server, a Java application server, a reporting server, and a database server. The deployment package contains several components, each corresponding to a tier of the application environment. A typical deployment consists of the following files:

```
<AppName>.ear
Exp_<FromDBUser>.zip
<AppName>-docroot.jar
<AppName>-external.zip
tools-ant.jar
tools.jar
tools-platform.jar
oracle-platform.jar
oracle-pricemaster.jar
oracle-oqd.jar
cognos-docroot.zip
cognos-webapp.zip
```

These deployment environment and distribution components are described in the following sections.

3.1 Enterprise Application Archive

The enterprise application archive (EAR) file is a compressed archive containing the Model N web application and its supporting files. Contained in the EAR file is a web application archive (WAR) file, which is deployed within the Java application server. For detailed instructions on how to install the EAR or WAR file within a specific application server, please consult:

- Installation of the EAR File for WebLogic
- Install the new enterprise application for WebSphere

Note: If you are installing Revenue Planning and Intelligence, you will need to un-archive that WAR file prior to deployment.

3.2 Database Export

If you do not wish to populate the Model N database with the initial application data, you can use the database export. The database export contains all of the tables and database schema that compose the Model N database. In addition to the structural definition of the database, the export file contains base application data that is necessary for the application to function correctly. This data includes access control data, application resources, and behavioral configurations.

If a database export file has been provided, it is generally provided as a compressed archive. The filename of the archive adheres to the following naming conventions:

```
Exp_<FromDBUser>.zip
```

This archive must be extracted using the appropriate compression utility before it can be imported into the database using the appropriate import tools.

The `<FromDBUser>` portion of the filename indicates the database user under which the export was created. This information is typically required when re-importing the database export. For example, you export an Oracle export by specifying the `<FromDBUser>` as the `fromuser` of the import file. For additional details regarding the import syntax, refer to Importing Data.

3.3 Static HTML Content

The Model N application requires that all static content be placed on an HTTP server. All application requests are then directed first to the HTTP server, which either proxies the request to the application server or serves any static files that can be found on the HTTP server. Model N uses an HTTP server due to the large number of images and static files that it references during every request. Using the HTTP server improves performance.

The **<AppName>**`-docroot.jar` file contains all of the static HTML content that must be placed on the HTTP server. Extract this file onto the HTTP server and place it into either the server docroot or the docroot of a virtual host created for the Model N application. Instructions for this process can be found in the WebLogic section, How to Configure iPlanet/Sun Java System Web Server, and in the WebSphere section, How to Configure the IBM HTTP Server (IHS).

3.4 Application Server Support Files

The application server support files are any files that are not distributed as part of the web application archive directly, but which must be available on the application server for use by the Model N application. These files include any server-specific Java libraries, application property files, and sample configuration files or scripts.

Any external support files will be placed in a `pharma-external.zip` or `meddev-external.zip` archive that can then be extracted into a permanent location on the application server. The contents of the `pharma-external.zip` or `meddev-external.zip` file should look similar to the following:

```
/content
/entities
/jdbc
  /ojdbc6.jar
/logs
/props
  /local_common.properties
  /local_weblogic.properties
  /local_websphere.properties
  /local_ops.properties
/scripts
   /database
      /directloader_proc.sql
      /directloader_exec.sql
   /monitoring
      /modeln_heartbeat
```

Following is a brief description of the file contents:

`/content`

Contains any data files that may be needed to populate the database using the Model N application. Any content that you upload to the application is also stored in this location. The application property files must reference this location for the application to locate the content files. (See the `com.modeln._default.contentRoot` and `com.modeln._default.featureBuildModule` application properties.)

`/entities`

The entity DTDs are used by the application XML parsers to validate .xml files and JSP files. Use these entities if the application server does not have access to the internet. The location of the entities directory is specified in the application property files. (See `com.modeln.entitiesRoot`)

`/jdbc`

You can find any additional Java libraries required by the Model N application but not included in the web application archive in the corresponding directory within the support archive. Model N recommends that you install these libraries on the application server somewhere in the classpath. For example, you should install the recommended Oracle JDBC drivers on the application server to ensure that there are no unexpected errors while running the application.

`/logs`

Model N recommends putting application logs in the logs directory. You can specify the log directory in the application property file. (See the property file example below.)

/props

The property directory contains the application property files that are used to initialize the Model N application. These property files are provided outside of the Web Application archive so that they can be customized by system administrators without need to re-deploy the application. Re-start the Model N application to effect changes to the property files. Include the property directory in the application classpath for the class loader to locate the property files.

/scripts

The scripts directory contains the scripts used to configure the direct loader as described in Direct Loader Channel Configuration.

The following code section shows an application property file for WebLogic and WebSphere. The bolded numbers one through seven that appear in parentheses are explained below the code section.

Code 3-1: Sample Application Property File for WebLogic and WebSphere

```
#====================================================================
# Model N Application Runtime Property
# ====================================================================

com.modeln.packaging.vertical=pharma (1)
com.modeln.packaging.configuration=FullSuitePackage
include_200=version
# Local variables used only within this property file
baseDir=/home/mn/release/mnmn
logDir=${baseDir}/logs
com.modeln._default.timeZone=America/Los_Angeles(2)
# Broadcast service allows for broadcasting messages from one Model N
server to another when all the Model N servers in the cluster access the
same Model N database schema.
com.modeln.BroadcastSvc.clusterName=mnmn (3)
com.modeln.env.nodeName=<uniqueNameForEachServer> (4)
com.modeln.BroadcastSvc.mcastAddress=<valid multi-cast address> (5)

# Log file name, size, and log level configuration
com.modeln.Log.MULTI_FILE_BASE_DIR=${baseDir}/content
com.modeln.Log.initFile=xml<ProdLogMgrInit.xml> (6)
com.modeln.Log.MAX_FILE_SIZE=1024
com.modeln.Log.MONITOR_BUFFSIZE=100000

# JMX configuration for application monitoring
com.modeln.jmxEnabled=true
com.modeln.jmxHttpPort=6025
com.modeln.jmxHttpHost=localhost
```

Code 3-1: Sample Application Property File for WebLogic and WebSphere (Continued)

```
# Cached App Server XML and DTD files
com.modeln.entitiesRoot=${baseDir}/entities (7)

# DataFlow Import/Export configuration
com.modeln._default.contentRoot=${baseDir}/content

# Application UI and authentication configuration
com.modeln.App.logoutRedirect=/modeln/app/login.html
com.modeln.App.staticClientFiles=true
com.modeln.App.applicationComponentName=\
   com.modeln.ac.application.CMnAppRootComp

# Cognos integration (8)
com.modeln.AppSwitch.disableCognosReporting=true
com.modeln.ReportingSvc.authenticationRequired=false
com.modeln.ReportingSvc.reportServerURL=\
   http://localhost:9999/p2pd/servlet/dispatch

# Direct Loader configuration (9)
com.modeln.db.allowSysScripts=false
com.modeln.DirectLoaderChannel.externalDirectory=C:\\DirectLoade
r
com.modeln.DirectLoaderChannel.requiresFTP=true
```

Code Section Notes:

(1) The packaging configuration statements define what application configuration to use. For more information, see Packaging.

(2) The `time zone` setting affects the use of dates within the application. If set incorrectly, the dates may appear incorrectly in the log file and may result in abnormal application behavior. Please ensure that the time and locale settings are correct at the operating system level as well, and that the application server and database server clocks are synchronized.

(3) The cluster name is used to establish a command cluster relationship between any Model N application instances that share the same cluster name. When application instances share a common cluster name, the Model N application will attempt to distribute commands across the cluster. To prevent unrelated application instances from communicating, each application instance must be configured with a unique cluster name. Set this property to configure broadcast service.

(4) Set this property to configure broadcast service.

(5) Set this property to configure broadcast service. Specify a valid multicast address (recommended multicast addresses are between 228.1.0.0 and 228.1.0.255) as the value of this property and ensure that this multicast address is not used for any other purpose on the local network. The Model N servers running as part of this cluster will communicate with each other using this multicast address. The multicast address specified for each of the Model N servers must be the same.

(6) The log manager configuration file can be used to configure the logging verbosity and output channels used by the application logger. Logs will be written to the location specified by the `FILE_NAME` and `MULTI_FILE_BASE_DIR` properties.

(7) The entities root can be used to perform DTD validation against local files rather than the files provided by the application server.

(8) For information on Cognos see Installing the Reporting Engine.

(9) For information on configuring the Direct Loader see Configuring Direct Loader.

3.5 Setting the Content Information

Set the following properties in the personal property file to determine where data files are exported to:

- `com.modeln._default.contentRoot=`<*location of the delta content root*>
- `com.modeln._default.contentSets=`<*content file sets*>
- `com.modeln._default.featureBuildModule=`<*content location of the specific feature*>
- `com.modeln._default.basePathSuffix=`<>

For example:

```
com.modeln._default.contentRoot=/home/jdoe/dev/54/projectName
com.modeln._default.contentSets=base,product,customer
com.modeln._default.featureBuildModule=company
com.modeln._default.basePathSuffix=content
```

The directory the data files will be exported to is:

`/contentRoot/featureBuildModule/basePathSuffix/contentSets`

- OR -

`/home/jdoe/dev/54/projectname/company/content/base/`

3.6 How to Re-deploy an Application

To re-deploy an application:

1. Stop the Model N application.
2. Stop Cognos in order to drop and import the Model N database.
3. Drop the Model N database schema.
4. Re-import the database dump.
5. Deploy the Model N EAR file.
6. Start Cognos.

7. If you have re-imported the database dump, make sure that you repeat the steps described in:
    a. the Java Stored Procedure Installation section
    b. the Configuring Direct Loader chapter
8. Start the Model N application.
9. Deploy the new report templates if there are any.

3.7 Operations Tools

The operations tools described in this section have been developed as ant tasks using ant version 1.6.5. The following ant-based tools are available in the Model N application:

- DBPreExport
- DBPostImport
- RunDataflow
- RunPopulate
- RunFullPopulate
- ReportDoAll
- ReportSchemaResolver
- ReportImporter
- ReportDeployer
- ReportModeller
- ActivateConfigurationSet
- DeactivateConfigurationSet
- StopEditingCustomConfigurationSet
- MigratePropsToConf
- ExportConfigurationSet
- ImportConfigurationSet
- Bootstrap

3.7.1 How to Configure the Environment

Important: All steps of the Ops tool deployment (configuring the environment, configuring the ant tools, and setting up the ant tools) must be repeated when deploying a new build.

To configure your environment to run these ant utilities:

1. Obtain the Model N binary distribution from either the Model N portal or the FTP site.
2. From your Model N binary distribution, copy the `tools.jar` file and `tools-ant.jar` file to a temporary directory.
3. Extract the tools files.
    a. Extract the `tools.jar` file (contains scripts files) to a tools directory as in the following examples for Windows and Linux:

```
c:\modeln\curBuild\tools
```

```
/opt/mn/release/tools
```

   b. Extract the `tools-ant.jar` file (contains ant binaries) to an ant directory as in the following examples for Windows and Linux:

      `c:\modeln\curBuild\tools\`

      `/usr/local/ant` or `/opt/mn/release/tools`

      Make sure that you grant executable permission to ant if you are using Linux and that you set the environment variable, `ANT_HOME`, to the directory that contains the `bin` directory of the ant executable. For example:

      `C:\modeln\curBuild\tools`

      If the `bin` directory is under `/opt/mn/release/tools`, the `ANT_HOME` would be `/opt/mn/release/tools`.

4. Make sure that the `props` directory and `prop` file are set up correctly.

   a. If you are setting up the ant tools before the application is set up, you need to create the props directory as follows:

      – On Windows `mkdir c:\modeln\props`

      – On Linux `mkdir /opt/mn/release/props`

   b. Get the `pharma-external.zip` file and unzip it to the Model N installation directory.

   c. Update the `local_ops.properties` file to match your system (for example, your database setting).

Note: The `local_ops.properties` file must set the `com.modeln.Log.FILE_NAME` property to a value different from your WebSphere or WebLogic property file in order to ensure that the operations tool does not overwrite the application log. Model N recommends setting the value to `com.modeln.Log.FILE_NAME=${logDir}/ops.log`.

3.7.1.1 How to Configure the Ant Tools

To configure the ant tools:

1. Go to the ant script directory:

   - On Windows `c:\modeln\curBuild\tools\modeln\build\ant\`
   - On Linux `/opt/mn/release/tools/modeln/build/ant`

2. Run the following configure command:

```
ant -f ops.xml configure
```

3. Answer the following prompts. Press **Enter** to accept the default answer:

Table 3-1: Ant Configuration Prompts

Prompt	Response
Please specify the location of the external directory: `[c:\modeln]`	This is typically the top-level directory for your project installation.
Please specify a temporary working directory: `[c:\modeln\curBuild\tools\temp]`	You must create this directory. This directory must have several gigabytes of available disk space and can be anywhere in the system.
Please specify the EAR file location: `[c:\modeln\curBuild\pharma.ear]`	Point to your Model N distribution file.
Please specify the Model N Jar file directory: `[c:\modeln\curBuild\modeln]`	Point to your Model N distribution directory. **Note:** Make sure that the Jar file directory contains all of the `*-content.jar` files (for example, `app-content.jar` and `compliance-content.jar`) that are in the Model N application.
Please specify the name of the Model N Operations property file: `[local_ops]`	Enter the name of the props file. It is not necessary to enter a path. The system will search for it by using the `[external directory/props/(prop filename)]`.
Please specify the application server type: (weblogic, websphere, jboss)	Enter the server type and press Enter.
Please specify the WebLogic (WebSphere) installation directory: `[c:\ibm\weblogic92]`	Enter the server installation directory and press Enter.
Please specify the location of the application server Java executable: `[c:\ibm\jdk150_04\bin\java]`	The Windows JDK version may be different from the Linux JDK version. For example, for WebLogic, the Java installed with the Linux version of WebLogic is in a different directory than in Windows. Be sure to override this if your JDK directory is different from the default.

After you have followed the preceding configuration steps, the `ops.properties` file is generated based on your input. This file is used for set up and to run the ant tools.

Note: Check the `ops.properties` file to make sure that you have not made any mistakes in the configuration. You can modify the `ops.properties` file directly.

3.7.1.2 How to Set Up the Ant Tools

To set up the ant tools:

1. Go to the ant script directory:
    - On Windows `c:\modeln\curBuild\tools\modeln\build\ant\`
    - On Linux `/opt/mn/release/tools/modeln/build/ant`
2. Run the following setup command:

    ```
    ant -f ops.xml setup
    ```

    This command uses the information in the `local_ops` and the `ops.properties` files to extract contents from the EAR file and put the contents into the appropriate locations.

The ant script setup is now complete and you can run the tools that are described in Running the Tools.

3.7.2 Running the Tools

Following are descriptions of the tools and instructions on how to run them.

3.7.2.1 DBPreExport

DBPreExport drops all materialized views and materialized view logs from the database by calling a stored procedure in the `MN_GLOBAL_PKG`. Run DBPostImport after dropping the materialized views to rebuild the logs and views.

To run DBPreExport, enter:

```
ant -f ops.xml DBPreExport
```

3.7.2.2 DBPostImport

DBPostImport accounts for creating the database objects that are dropped when a database is exported. This includes materialized views and indices. Indices are typically not exported, so running DBPostImport regenerates the indices along with the materialized views and materialized view logs.

To run DBPostImport, enter:

```
ant -f ops.xml DBPostImport
```

Note: Before you run DBPostImport, you must add the following property to the `local_ops.properties` file in order for the database schema to be analyzed during the DBPostImport process:
`com.modeln.analyzeschema.dbuser=${myprop.dbuser}`

3.7.2.3 RunDataflow

Data flows are typically Inbound or Outbound. Inbound data flows receive data from external systems, process the data and publish the results of processing to the Model N system. Outbound data flows extract data from the Model N system, process it and then send it to an external system.

To run RunDataflow:

1. Enter:

    ```
    ant -f ops.xml RunDataflow
    ```

The following message is displayed:

```
Please specify dataflow parameters (Ex: -c
ConfigurationFromXml -o Root)
```

2. Specify the data flow parameters.

The following is a list of the options that you can use with RunDataflow:

Table 3-2: Data Flow Options

Flag	Argument	Required	Description
-am	<true|false>	false	Interactively queries the database for runs and requests aborts.
-b	<browser cmd>	false	System command to start browser (see -s).
-c	<conf name>	false	Loads all the data flows for a configuration.
-dm	<true|false>	false	Displays the debugging output.
-dp	<true|false>	false	Interactively defines the runtime properties.
-drm	<true|false>	false	Interactively queries the database for runs and deletes them.
-d	<true|false>	false	Describes the available configurations.
-dr	<true|false>	false	Begins the data flow but does not process any `DataSets`.
-e	<true|false>	false	Echoes the `DataflowDriver` configuration.
-eem	<true|false>	false	Interactively queries the database for runs and exports errors.
-f	<true|false>	false	Files to process.
-h		false	Shows usage.
-od	<directory>	true	Directory for message summaries.
-pf	<prop file>	true	Model N system property file.

Table 3-2: Data Flow Options (Continued)

Flag	Argument	Required	Description

-r	<realm>	true	The realm. The available values are:
			`GLOBAL-REALM`
			`DEFAULT-REALM`

-rb	<true|false>	false	Automatically rolls back a database transaction.
-rm	<true|false>	false	Recovers an existing run.
-rn	<run name>	false	Specifies the name of a run.
-rtp	<rtp1 rtp2 ...>	false	Specifies runtime properties in the following format: `name:value[:class]`
-sn	<true|false>	false	Allows notifications of various events to be sent.
-sr	<true|false>	true	Saves the summary to disk.
-s	<true|false>	false	Automatically shows the summary using browser setting (see -b).
-sm	<true|false>	false	Interactively queries the database for old runs and generates summaries.
-t	<transform name>	true	The name of `MsgLogTransform` to apply to Summary.
-wm	<true|false>	false	Waits for the user to shut down the environment.
-u	<user name>	true	Model N community member name.

3.7.2.4 RunPopulate

RunPopulate populates the content files for one Model N reader.

To run RunPopulate, enter:

```
ant -f ops.xml RunPopulate
```

You are prompted by the following text:

```
Please specify a Readers's name (Required):
```

If you do not provide a Reader's name, you receive the following error message:

```
BUILD FAILED
Please specify an Importers reader name and try again.
```

3.7.2.5 RunFullPopulate

RunFullPopulate populates all of the Model N readers and data flow content files.

To run RunFullPopulate, enter:

```
ant -f ops.xml RunFullPopulate
```

You are prompted by the following text:

```
Please specify the output log file [RunFullPopulate.log]:
```

If you do not provide a file name, the output log will redirect you to the default log file, `RunFullPopulate.log` which is in the current directory.

3.7.2.6 ReportSchemaResolver

The ReportSchemaResolver tool takes the Cognos archives and updates database schema references in them and optionally applies a unique identifier to the archive content, which consists of models and reports. For more information on this tool, see the Model and Report Deployment chapter in the Model N *Reporting Guide*.

You can use one of the following options with ReportSchemaResolver:

Table 3-3: ReportSchemaResolver Option

Flag	Argument	Required	Description
-add	add	yes	The add argument adds the identifier and updates the database schema name.
-remove	remove	yes	The remove argument removes the identifier.
-db	db	yes	The db argument updates the database schema name.

To run ReportSchemaResolver, enter:

```
ant -f ops.xml ReportSchemaResolver
```

3.7.2.7 ReportImporter

The ReportImporter tool imports the application-specific Cognos archives into the Cognos Content Store. For more information on this tool, see the Model and Report Deployment chapter in the Model N *Reporting Guide*.

To run ReportImporter, enter:

```
ant -f ops.xml ReportImporter
```

3.7.2.8 ReportDeployer

The ReportDeployer tool deploys the application-specific report templates to the Model N environment. For more information on this tool, see the Model and Report Deployment chapter in the Model N *Reporting Guide*.

To run ReportDeployer, enter:

```
ant -f ops.xml ReportDeployer
```

Tip: Make sure that you let the tool run to completion because the tool copies the appropriate reports over. If you interrupt it, some reports may not be copied over and you will receive error messages for those reports that were not copied over.

3.7.2.9 ReportDoAll

Use the ReportDoAll tool to call the ReportSchemaResolver tool, the ReportImporter tool, and the ReportDeployer tool in succession. The purpose of the ReportDoAll tool is to make it easier to call these other tools by allowing you to run them all in succession rather than run them each separately.

Refer to the Schema Resolver Tool section of the Model and Report Deployment chapter in the *Reporting Guide* for which properties you must set to configure the ReportSchemaResolver tool.

Set `com.modeln.ReportingService archivefolderPath` to the directory containing your Cognos archives. Set `com.modeln.ReportingService dbSchema` to the schema/user for your Model N database. For instance, if you use the user `MNPH` to log into the database, specify `MNPH` here.

Important: To make reports available in the Model N application, you must run the ReportDoAll tool, or the three tools that it calls. Otherwise, you will not be able to view any reports in the Model N application.

You can use one of the following options with ReportDoAll:

Table 3-4: ReportDoAll Option

Flag	Argument	Required	Description
-import	import	yes	The import argument deploys reports from Cognos archives.
-export	export	yes	The export argument makes Cognos archives ready for deployment elsewhere. This is equivalent to running the ReportSchemaResolver tool using the -remove option.

To run ReportDoAll, enter:

```
ant -f ops.xml ReportDoAll
```

For more information on the ReportDoAll tool, see Importing and Deploying Models and Reports.

3.7.2.10 ReportModeller

The ReportModeller tool generates models based on Model N objects (FGOs) and the database schema. It translates the FGOs, their attributes and the relationships into Cognos query subject attributes and relationships. For more information on this tool, see the Model N Model Generation Tool chapter in the Model N *Reporting Guide*.

To run ReportModeller, enter:

```
ant -f ops.xml ReportModeller
```

3.7.2.11 ActivateConfigurationSet

The ActivateConfigurationSet tool activates a configuration set.

To run ReportModeller, enter:

```
ant -f ops.xml ActivateConfigurationSet
```

3.7.2.12 DeactivateConfigurationSet

The DeactivateConfigurationSet tool deactivates a configuration set.

To run DeactivateConfigurationSet enter:

```
ant -f ops.xml DeactivateConfigurationSet
```

3.7.2.13 StopEditingCustomConfigurationSet

The DeactivateConfigurationSet tool deactivates a configuration set.

To run DeactivateConfigurationSet enter:

```
ant -f ops.xml StopEditingCustomeConfigurationSet
```

3.7.2.14 MigratePropsToConf

The `MigratePropsToConf` tool converts the properties under customer features to the configuration set format and put under the `<feature>/conf/` directory structure. The FGO and CGO definition configurations are migrated to .xml files and other properties are categorized and shuffled around to different property files. The dynamic enumerations is migrated to content files under the `<feature>/content/base/Global/Enumerations/` directory. This is only needed for migrations form a pre-5.3 release to 5.3 or later.

How to Run MigratePropsToConf

1. Run `printprops.sh` to print out the list of properties and save it somewhere before you run the MigratePropsToConf tool. You can compare the properties before and after the database migration to make sure no properties are missing.
2. Set up the OPS tools ant environment.
3. Create the packaging `xxxPkgConfig.xml` file (FullSuitePackage + customer feature) if migrating from a 5.0.x or earlier release.

4. Create the configuration migration spec, `xxxConfMigrationSpec.xml`, to specify which customer features will be migrated and where to output the generated new configuration files.
5. For each customer feature or module, create a `<feature>_main_en_US.properties` file for the feature. This file must directly or indirectly include all other properties for the feature. Remove all the file includes of other `<modelnBaseProductFeature>_main` since feature dependencies are no longer defined in property files. They are defined in `xxxPkgConfig.xml` files.
6. Run MigratePropsToConf in the target ANT environment to migrate the property files to the new configuration formats. The output directory can be specified in `xxxConfMigrationSpec.xml` file.

Note: You need Java JDK 1.5 or higher.

The following inputs are required:

Table 3-5: MigratePropstoConf Prompts

Prompt	Response
Please enter the path to your custom property files.	Example: `/home/you/dev/company/nni/props`
Please enter the path for the custom package config xml file.	Example: `/home/you/dev/company/config` Optional if the package config xml file is already in the class path.
Please enter the configuration migration extension specification file.	Example: `/home/you/dev/company/migration/MyConfMigrationSpec.xml`

7. Copy the generated `<feature>/conf/` directory and `<feature>/content/` directory from the temp output directory back to the workspace, then prepare to run the database migration.
8. Run `printprops.sh` again after database migration. Compare the outputs with the original output to make sure no properties were lost after the database migration.

The `CustomConfMigrationSpec.xml` file is located in the scripts directory, which is described in the section on Application Server Support Files.

Following is an example of a `CustomConfMigrationSpec.xml` file:

Code 3-2: CustomConfMigrationSpec.xml File

```
<?xml version="1.0" encoding="UTF-8" ?>
<!-- OutputDirectory is the directory where the new conf files will be
written to.
  -->
<!-- RootDirectory is the parent directory of the features or modules.
  -->
```

Code 3-2: CustomConfMigrationSpec.xml File (Continued)

```
<!-- Please modify these values for your env
  -->
<conf_migration_extn_spec RootDirectory="/home/mjoshi/dev/53/modeln"
OutputDirectory="/home/mjoshi/dev/temp">
<!-- Specify the features that need to be migrated here
  -->
<migrate_features>
<!--
     If no feature is specified, all deployed features will be migrated,
     otherwise, only the specified features will be migrated



  -->
  </migrate_features>
<classifiers>
<!--
     One or more classifiers can be registered here.
     The attributes ClassName, PropertyPattern and DestinationFile are
all mandatory attributes.
     This classifier will be invoked whenever a property which contains
this property pattern is
     encountered. The destination file must be relative to the conf
directory for the feature
     or module that the property belongs to. An example would be
     <classifier ClassName="com.bsc.CBscPropertyClassifier"
PropertyPattern="com.bsc"
     DestinationFile="bscprops/${feature}_custom.properties" >
     ${feature} is a built-in construct that the framework will
translate into the feature or
     module name at runtime.



  -->
<!--
<classifier
ClassName="com.modeln.tools.migration.proptoconf.extension.test.CMnTest
ExtnConfClassifier"
    PropertyPattern="bsc.migration" DestinationFile="bscprop/
bsc_custom_en_US.properties"/>


  -->
  </classifiers>
<!--  Lists all the property files that need to be included when running
in development mode
  -->
<dev_prop_files Description="This lists the property files within a
feature that is to be used for development alone">
  <dev_prop_file Feature="platform" ResourceName="intg_dev" />
  </dev_prop_files>
<retain_prop_files Description="These are directories or files whose
contents must not be migrated. They should be moved to under the conf
directory as is">
```

Code 3-2: CustomConfMigrationSpec.xml File (Continued)

```
<!--
ResourceName: Please do not keep suffix like properties or
_en_US.properties
        Full file name can be used is it is not a property file


  -->
<!--  Default Target dir = "properties"
  -->
  <prop_file Feature="gp" ResourceName="plugins/gpplugins"
MoveSubDir="plugins" />
  <prop_file Feature="usregpharma" ResourceName="plugins/benchmark_v1"
/>
  <prop_file Feature="gp" ResourceName="gp_ptb" />
<!-- Move dtd files
  -->
  <prop_file Feature="app" ResourceName="reportDeployment.dtd"
TargetDir="dtd" />
  <prop_file Feature="app" ResourceName="reportModel.dtd"
TargetDir="dtd" />
  <prop_file Feature="app" ResourceName="reportModule.dtd"
TargetDir="dtd" />
<!--  Adhoc Model property files
  -->
  <prop_file Feature="gp" ResourceName="gp_adhoc_model" />
  </retain_prop_files>
<retain_logical_grouping>
  <prop_file Feature="ALL" ResourceNamePattern="_queries" />
  </retain_logical_grouping>
  </conf_migration_extn_spec>
```

Workspace Setup to Migrate Properties for Custom Features

For the `MigratePropsToConf` script to detect and move the SQL and .xml files under `/props` for custom features, the `/props` directory must be located as follows:

```
<AppRoot>/<featurename/props/
```

If your current workspace is not set up in this way, create a temp workspace and copy all the files under /`props` to the temp workspace.

In the external `ConfMigrationSpec.xml` file, configure the root directory and set it to `<AppRoot>`. Configure the output directory. It can be any existing directory. the generated `<featurename>/conf/`is written to the output directory. You can copy the root directory and the output directory to your workspace when you are done migrating the property files to configuration files.

3.7.2.15 ExportConfigurationSet

The ExportConfigurationSet tool migrates your property files to the configuration directory structure.

To run ExportConfigurationSet enter:

```
ant -f ops.xml ExportConfigurationSet
```

For more information on the ExportConfigurationSet tool, see the Configuration Console chapter in the *Operations Guide*.

3.7.2.16 ImportConfigurationSet

The ImportConfigurationSet tool migrates your property files to the configuration directory structure.

To run ImportConfigurationSet enter:

```
ant -f ops.xml ImportConfigurationSet
```

For more information on the ImportConfigurationSet tool, see the Configuration Console chapter in the *Operations Guide*.

3.7.2.17 Bootstrap

The Bootstrap tool uploads application configuration .xml files which include FGO and CGO definitions and creates database materialized views and indices.

To run Bootstrap enter:

```
ant -f ops.xml Bootstrap
```

When Bootstrap is run, you are prompted for a custom configuration set name:

```
Enter a custom config set name for update (optional):
```

If you are loading a saved configuration, type the name of the configuration. If not, press Enter to bypass that process.

4 Configuring the Oracle Database

This chapter describes the configuration and setup of the Oracle database. The description assumes that Oracle has been installed correctly and that a database administrator is available to create the users and tablespaces required by the Model N application.

The following are requirements for installation:

- Oracle 11gR2 Database Server
- Oracle JVM (J2SE 1.3 compliant)

Note: See the Supported Platform Matrix for the specifically supported server configurations.

Use of multiple schemas in the same database is not supported.

Some of the shell/cmd scripts in this chapter contain lines that are longer than can be displayed on the width of a page. Lines that have been broken for page formatting reasons are indicated with the pilcrow (¶) symbol. Do not separate these lines in the actual file or the script may fail to run.

4.1 Setup and Configuration

This section provides information on the requirements for setting up and configuring Model N.

- If you use Direct Loader to perform data flows, you will need an FTP server running on the database host. For more information, see Configuring Direct Loader.
- If you are installing Revenue Planning & Intelligence (RPI), you will need to create a user for RPI and grant that user certain privileges because RPI uses a different schema than the rest of the Model N application.

4.1.0.1 Oracle

You can install Oracle with default settings in addition to the settings recommended for `init.ora` in the section, Initialization File. However, you can reduce the size of the Oracle installation as follows:

1. Start DBCA.
2. Choose to create a new database.
3. Set the global database name.
4. Clear all of the options and choose YES to delete the associated tablespaces.
5. Click the Standard Database Features button.
6. Clear all options except the JVM option (if the JVM is required) and choose YES to delete the tablespaces.
7. Click OK to continue.

Note: The Oracle JVM (any J2SE 1.4.2 compliant version) is required when using Java stored procedures to implement the functionality of some Model N applications, including Price Master, Promotions (order-Quantity Discounts or OQD), and Configuring Direct Loader.

4.1.0.2 Character Set

To support multiple languages, the database server must be configured to store characters using a UTF-8 encoding.

Note: Oracle recommends using `AL32UTF8` as the database character set and `AL16UTF16` as the national character set. The `AL32UTF8` character set is a superset of all other Oracle `UTF8` character sets (including the basic `UTF8` character set), so using `AL32UTF8` would minimize the risk of characters not being represented in the database character set. Also, `AL32UTF8` is recommended as the database character set for use with Cognos in Model N Reporting, so using it further supports the option of using a single database instance for both Model N and Cognos when size is an issue.

4.1.0.3 JVM

If you use Model N Java stored procedures, you must configure Oracle to support the Oracle JVM. The following table lists the stored procedure packages and the features that they contain.

Table 4-1: Java Stored Procedure Packages

Feature	Stored Procedure Package
Direct Loader	`oracle-platform.jar`
Price Master	`oracle-pricemaster.jar`
Order Quantity Discount	`oracle-oqd.jar`

Ensure that the JVM is correctly installed before running the Model N application. The Oracle JVM is installed by default when the DBCA utility is used to create the database. If the JVM has not been installed, it must be installed manually. The following system requirements are necessary to install the Oracle JVM:

- `shared_pool_size` >= 100MB (minimum)
- `java_pool_size` >= 150MB
- `SYSTEM` tablespace > 120MB of free space or created with `AUTOEXTEND ON`
- `UNDO` tablespace > 250MB of free space

To install the JVM manually, run the following script from an account with `SYSDBA` privilege:

```
SQL> @?/javavm/install/initjvm.sql
```

For a complete set of instructions, go to https://metalink.oracle.com/ and refer to article 149393.1. If the JVM installation fails or appears to be functioning incorrectly, you need to completely remove and re-install the JVM.

To verify a JVM installation, connect to the Oracle database and perform the query that is listed below. Depending upon the platform and products installed, the results of this query should resemble the following results:

```
SQL> select owner,status,count(*) from dba_objects
2 where object_type like '%JAVA%'
3 group by owner,status;


OWNER                 STATUS            COUNT
ODM                   VALID             247
SYS                   VALID             261
WKSYS                 VALID             40
ORDSYS                VALID             903
ORDPLUGINS            VALID             1
```

The following system requirements must be met to install the Oracle JVM:

- The Pool must have at least 24MB of free space.
- The Java Pool must have at least 12MB of free space.
- The SYSTEM tablespace must have at least 70MB of free space.
- The SYSTEM RBS must have at least 100MB of free space.

The `initjvm.sql` script in 11g will check to see that these resources are available when it is run. If they are not available, the execution of the script will terminate with an error indicating which resource needs to be increased.

4.1.0.4 Listener Configuration

To avoid getting ORA-3136 in the `Alerts.log` file, the following parameter can be set in `$ORACLE_HOME\NETWORK\ADMIN\sqlnet.ora` file.

```
sqlnet.inbound_connect_timeout=0
```

4.1.1 Initialization File

The following table describes the recommended settings that differ from the default `init.ora` configuration. These settings are based on an assumption of a dual-CPU dedicated database server with 4GB RAM. A superscript after a value indicates that there is a corresponding footnote following the table.

Note: Because the Model N *Installation Guide* is not a tuning guide, the information in the following table describes reasonable initial settings for an Oracle database used by the Model N application.

Table 4-2: Initial Oracle Database Settings

Setting	Value	Description
_allow_level_ without_connect_by	TRUE[2]	Allows referencing columns named level in queries without specifying a connect by clause.
_b_tree_bitmap_ plans	FALSE[5]	Disables the use of bitmap plans for tables that only have B-tree indexes.
_fast_full_scan_enable d	FALSE[5]	Disables index fast full scan operations.
_gby_hash_ aggregation_ enabled	FALSE[4]	Disables group-by and aggregation using hash schemes. **Note:** Remove this setting if you have Oracle 11gr2.
_like_with_bind_as_e quality	TRUE[5]	Enables treatment of LIKE predicates with bind as an equality predicate.

Table 4-2: Initial Oracle Database Settings (Continued)

Setting	Value	Description
optimizer connect_by_cost_based	FALSE[2]	Disables cost-based transformation for connect by queries.
optimizer skip_scan_ enabled	FALSE	Disables the index skip scan.
_remove_aggr_ subquery	FALSE[2]	Disables removal of subsumed aggregated subqueries. **Note:** Remove this setting if you have Oracle 11gr2.
sort elimination_cost_ratio	5[5]	Defines cost ratio for sort elimination in first_rows mode.
compatible	X.X.X.X.X	This parameter specifies the release with which the Oracle server must maintain compatibility. It allows you to take advantage of the maintenance improvements of a new release immediately in a production system without testing the new functionality in a test environment. Some features of the current release may be restricted. To ensure availability of fixes deployed through Oracle patches, this parameter must be set to the latest patch version applied. For example, after installing Oracle patch 10.2.0.3.0 this parameter should be set to 10.2.0.3.0.
`db_block_size`	8192	The size of an Oracle data block.
`db_cache_size`	1073741824[1]	This parameter specifies the size of the `DEFAULT` buffer pool for buffers with the primary block size (the block size defined by the `DB_BLOCK_SIZE` parameter). Like the `shared_pool_size`, this parameter is application-dependent and in order to tune it correctly, the buffer cache hit ratio has to be monitored on an ongoing basis. It is also possible to set the `db_cache_advice` parameter to `on` to enable statistics gathering used for predicting behavior with different cache sizes through the `V$DB_CACHE_ADVICE` performance view.

Table 4-2: Initial Oracle Database Settings (Continued)

Setting	Value	Description
db_files	1024	This parameter specifies the maximum number of database files that can be opened for this database. The maximum valid value is the maximum number of files, subject to operating system constraint, that will ever be specified for the database, including files to be added by `ADD DATAFILE` statements.
`db_writer_processes`	4	The number of background database writer processes. Oracle may adjust this parameter based on the number of CPUs and processor groups. Oracle recommends setting this parameter to a value less than or equal to the number of available CPUs. However, setting this parameter to a higher value may also help speed up write operations on systems with lots of DML operations, even if only one CPU is available. In any case, since this parameter is CPU bound, `db_writer_processes` should be set to its default value of `1`, if the server experiences CPU contention.
fast_start_mttr_target	300	This parameter specifies the number of seconds the database takes to perform crash recovery of a single instance. **Note:** This is only for Oracle 11g Enterprise Edition. If you are using Oracle 11g Standard Edition, you can ignore this parameter.
filesystemio_options	setall	Enables direct I/O and asynchronous I/O operations for file system files. This is the recommended setting to ensure optimal performance for read/write operations on data files.
`java_pool_size`	167772160[1]	This parameter should be set to 167772160 (160MB) when using the JVM. Otherwise, it can be set to 0 to disable the Java pool. When disabling the Java pool, the 160MB of RAM that are freed up can be added to the `db_cache_size`, `shared_pool_size`, and/or `large_pool_size`.
job_queue_processes	10	This parameter specifies the maximum number of processes that can be created for the execution of jobs and is required when scheduling jobs through the `DBMS_JOB` package. For example, it is required for gathering database statistics automatically.

Table 4-2: Initial Oracle Database Settings (Continued)

Setting	Value	Description
nls_length_ semantics	CHAR[3]	Enables character-length semantics for CHAR and VARCHAR2 columns.
open_cursors	2000	This parameter specifies the maximum number of cursors allowed in a single session.
optimizer_ features_enable	X.X.X.X	This parameter acts as an umbrella parameter for enabling a series of optimizer features based on an Oracle release number. It should be set to the latest value for a given release or patch. That value can be found in the Oracle Database Reference for base releases or the Oracle Patch Documentation when applying patches.
optimizer_secure_view_merging	FALSE[5]	Disables view merging or predicate move-around globally for the database.
`pga_aggregate_ target`	1073741824	This parameter specifies the target aggregate PGA memory available to all server processes attached to the instance. This parameter must be set to enable the automatic sizing of SQL working areas used by memory-intensive SQL operators such as `sort`, `group-by`, `hash-join`, `bitmap merge`, and `bitmap create`. Oracle uses this parameter as a target for PGA memory. The `pga_aggregate_target` parameter indicates the maximum amount of memory that the server can allocate to the PGA (Program Global Area), thus determining the optimal size of each work area allocated in auto mode, for example, when `workarea_size_policy` is set to auto. Oracle recommends this be set to 16% of the physical memory for OLTP systems and 40% of physical memory for DSS systems.
`processes`	300	This parameter specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value must allow for all background processes such as locks, job queue processes, and parallel execution processes. This parameter may have to be set to a higher value on systems with lots of concurrent sessions.

Table 4-2: Initial Oracle Database Settings (Continued)

Setting	Value	Description
remote_login _passwordfile		Specifies that the Oracle password file can be used by more than one database and the password file can contain names other than SYS.
session_cached_ cursors	500[5]	Specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed. This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are re-executed by a user.
`shared_pool_size`	1073741824[1]	The pool usage is highly application-dependent, therefore it is necessary to examine each database application individually in order to project a recommended pool value. Like the buffer cache hit ratio, the pool hit ratio has to be monitored on an ongoing basis. It is a good idea to start with a higher value for this parameter and then monitor the database to see if this value can be increased or decreased.
undo_management	AUTO	This parameter starts the instance in automatic undo management mode, meaning that no manual management of rollback segments is required.
undo_retention	10800	The amount in seconds of committed undo information to be retained in the database. This parameter can be used to satisfy queries that require old undo information to rollback changes to produce older images of data blocks.
undo_tablespace	UNDOTBS1	Specifies the undo tablespace to be used when an instance starts up in automatic undo management mode.

Table 4-2: Initial Oracle Database Settings (Continued)

Setting	Value	Description
`workarea_size_ policy`	AUTO	Turns on the automatic SQL execution memory management, thus allowing Oracle to size work areas used by memory-intensive operators automatically, based on the PGA memory used by the system, the target PGA memory set in the `pga_aggregate_target` parameter and the requirement of each individual operator. The `workarea_size_policy` can only be set to auto if the `pga_aggregate_target` parameter has also been specified.
`_optim_peek_user _binds`	false	Enables peeking of user binds.

[1] Since automatic SGA tuning introduced significant overhead in the tests, Model N recommends that you use the old-fashioned memory management instead of Oracle 11g's automatic SGA tuning through `sga_target`.

[2] Required to overcome Oracle errors caused by optimizer changes or bugs. Particularly, this addresses Bug 5391575 in Oracle Metalink for Oracle 10.2.0.3.

[3] `nls_length_semantics` have been changed to CHAR for the 5.3 release to address UTF-8 handling issues.

[4] Oracle recommends disabling a new 10g feature called Hash Group by aggregation. If this feature causes a performance degradation in queries migrated from Oracle 9i.

[5] These settings have been taken from Oracle's recommended database initialization parameter settings for Oracle Applications Release 12 on Oracle 10g databases.

Consult the Model N database support staff before using any non-default values other than those specified in the preceding table.

The following table describes the recommended settings that differ from the default `init.ora` configuration. These settings are based on an assumption of a dual-CPU dedicated database server with 8GB RAM.

Note: Because the Model N *Installation Guide* is not a tuning guide, the information in the following table describes reasonable initial settings for an Oracle database used by the Model N application.

Table 4-3: Recommended Initial Settings for Oracle

Setting	Value	Description
_allow_level_ without_connect_by	TRUE[2]	Allows referencing columns named level in queries without specifying a connect by clause.

Table 4-3: Recommended Initial Settings for Oracle (Continued)

Setting	Value	Description
_b_tree_bitmap_ plans	FALSE[5]	Disables the use of bitmap plans for tables that only have B-tree indexes.
_fast_full_scan_enabled	FALSE[5]	Disables index fast full scan operations.
_gby_hash_ aggregation_ enabled	FALSE[5]	Disables group-by and aggregation using hash schemes.
`db_block_size`	8192	The size of an Oracle data block.
db_cache_size	4294967296[1]	The number of database block buffers in the Oracle SGA. Like the `shared_pool_size` this parameter is application-dependent and in order to tune is correctly the buffer cache hit ratio has to be monitored on an ongoing basis. It is also possible to set the `db_cache_advice` parameter to on to enable statistics gathering used for predicting behavior with different cache sizes through the `V$DB_CACHE_ADVICE` performance view.
`compatible`	X.X.X.X.X	This parameter specifies the release with which the Oracle server must maintain compatibility. It allows you to take advantage of the maintenance improvements of a new release immediately in a production system without testing the new functionality in a test environment. Some features of the current release may be restricted. To ensure availability of fixes deployed through Oracle patches, this parameter must always be set to the latest patch version applied. For example, after you install Oracle patch 10.2.0.3.0 you should set this parameter to 10.2.0.3.0.
db_files	1024	This parameter specifies the maximum number of database files that can be opened for this database. The maximum valid value is the maximum number of files, subject to operating system constraint, that will ever be specified for the database, including files to be added by ADD DATAFILE statements.

Table 4-3: Recommended Initial Settings for Oracle (Continued)

Setting	Value	Description
db_writer_processes	4	This parameter specifies the number of background database writer processes. Oracle may adjust this parameter based on the number of CPUs and processor groups. Oracle recommends setting this parameter to a value less than or equal to the number of available CPUs. However, setting this parameter to a higher value may also help speed up write operations on systems with lots of DML operations, even if only one CPU is available. In any case, since this parameter is CPU bound, `db_writer_processes` should be set to its default value of 1, if the server experiences CPU contention.
fast_start_mttr_target	300	This parameter specifies the number of seconds the database takes to perform crash recovery of a single instance.
filesystemio_options	setall	Enables direct I/O and asynchronous I/O operations for file system files. This is the recommended setting to ensure optimal performance for read/write operations on data files.
`java_pool_size`	167772160[1]	This parameter should be set to 167772160 (160MB) when using the JVM. Otherwise, it can be set to 0 to disable the Java pool. When disabling the Java pool, the 160MB of RAM that are freed up can be added to the `db_cache_size`, `shared_pool_size`, and/or `large_pool_size`.
job_queue_processes	10	This parameter specifies the maximum number of processes that can be created for the execution of jobs and is required when scheduling jobs through the `DBMS_JOB` package. For example, it is required for gathering database statistics automatically.
_like_with_bind_as_equality	TRUE[5]	Enables treatment of LIKE predicates with bind as an equality predicate.
nls_length_semantics	CHAR[3]	Enables character length semantics for CHAR and VARCHAR2 columns.
open_cursors	8000	This parameter specifies the maximum number of cursors allowed in a single session.
_optimizer_connect_by_cost_based	FALSE[2]	Disables cost-based transformation for connect by queries.

Table 4-3: Recommended Initial Settings for Oracle (Continued)

Setting	Value	Description
optimizer_ features_enable	X.X.X.X	This parameter acts as an umbrella parameter for enabling a series of optimizer features based on an Oracle release number. It must be set to the latest value for a given release or patch. That value can be found in the Oracle Database Reference for base releases or the Oracle Patch Documentation when applying patches.
optimizer_secure_view_ merging	FALSE[5]	Disables view merging or predicate move-around globally for the database.
optimizer skip_scan_ enabled	FALSE	Disables the index skip scan.
`pga_aggregate_ target`	2147483648	This parameter specifies the target aggregate PGA memory available to all server processes attached to the instance. This parameter must be set to enable the automatic sizing of SQL working areas used by memory-intensive SQL operators such as `sort`, `group-by`, `hash-join`, `bitmap merge`, and `bitmap create`. Oracle uses this parameter as a target for PGA memory. The `pga_aggregate_target` parameter indicates the maximum amount of memory that the server can allocate to the PGA (Program Global Area), thus determining the optimal size of each work area allocated in auto mode, for example, when `workarea_size_policy` is set to auto. Oracle recommends this be set to 16% of the physical memory for OLTP systems and 40% of physical memory for DSS systems.
processes	600	This parameter specifies the maximum number of operating system user processes that can simultaneously connect to Oracle. Its value must allow for all background processes such as locks, job queue processes, and parallel execution processes. This parameter may have to be set to a higher value on systems with lots of concurrent sessions.
remote_login_ passwordfile		Specifies that the Oracle password file can be used by more than one database and the password file can contain names other than SYS.

Table 4-3: Recommended Initial Settings for Oracle (Continued)

Setting	Value	Description
_remove_aggr_ subquery	FALSE[2]	Disables removal of subsumed aggregated subqueries.
sort elimination_cost_ratio	5[5]	Defines cost ratio for sort elimination in first_rows mode.
session_cached_ cursors	500[5]	Specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement cause the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently used algorithm to remove entries in the session cursor cache to make room for new entries when needed. This parameter also constrains the size of the PL/SQL cursor cache which PL/SQL uses to avoid having to reparse as statements are re-executed by a user.
`shared_pool_size`	1073741824[1]	The pool usage is highly application-dependent. Therefore, it is necessary to examine each database application individually in order to project a recommended pool value. Like the buffer cache hit ratio, the pool hit ratio has to be monitored on an ongoing basis. It is a good idea to start with a higher value for this parameter and then monitor the database to see if this value can be increased or decreased.
undo_management	AUTO	This parameter starts the instance in automatic undo management mode, meaning that no manual management of rollback segments is required.
undo_retention	10800	The amount in seconds of committed undo information to be retained in the database. This parameter can be used to satisfy queries that require old undo information to rollback changes to produce older images of data blocks.
undo_tablespace	UNDOTBS1	This parameter specifies the undo tablespace to be used when an instance starts up in automatic undo management mode.

Table 4-3: Recommended Initial Settings for Oracle (Continued)

Setting	Value	Description
`workarea_size_policy`	AUTO	Turns on the automatic SQL execution memory management, thus allowing Oracle to size work areas used by memory-intensive operators automatically, based on the PGA memory used by the system, the target PGA memory set in the `pga_aggregate_target` parameter and the requirement of each individual operator. The `workarea_size_policy` can only be set to auto if the `pga_aggregate_target` parameter has also been specified.
`_optim_peek_user_binds`	FALSE	Disables the bind variable peeking.
`sec_case_sensitive_logon`	FALSE	Disables password case sensitivity during database logins. This parameter is recommended to be set to FALSE to be compatible with 10g behavior, but IT should consider turning it on for better security.

[1] Since automatic SGA tuning introduced significant overhead in the tests, Model N recommends that you use the old-fashioned memory management instead of Oracle 11g's automatic SGA tuning through `sga_target`.

[2] Required to overcome Oracle errors caused by optimizer changes or bugs.

[3] `nls_length_semantics` have been changed to CHAR for the 5.3 release to address UTF-8 handling issues.

[4] Oracle recommends disabling a new 10g feature called Hash Group by aggregation. If this feature causes a performance degradation in queries migrated from Oracle 9i.

[5] These settings have been taken from Oracle's recommended database initialization parameter settings for Oracle Applications Release 12 on Oracle 10g databases.

Consult the Model N database support staff before using any non-default values other than those specified in the preceding table.

4.1.2 Documented Oracle Parameters Added for Oracle 11gR2

The following standard Oracle parameters have been added to the parameter file for Oracle 11.2.0.1:

filesystemio_options = setall

This parameter is used to avoid buffered I/O at the operating system level. The `SETALL` option sets all the options for a particular file system to enable direct I/O or asynchronous I/O. Setting the `FILSYSTEMIO_OPTIONS` to anything other than `SETALL` could reduce performance.

nls_length_semantics = CHAR

This parameter setting ensures that `CHAR` and `VARCHAR2` columns are always created using character length semantics. Existing columns are not affected. `NLS_LENGTH_SEMANTICS` does not apply to tables in the `SYS` and `SYSTEM` schemata. The data dictionary always uses byte semantics.

optimizer_secure_view_merging = FALSE

This parameter setting causes Oracle not to use view merging or predicate move-around when determining the cost of a query. If this parameter is set to `TRUE`, Oracle assesses the query, considering all transformations and chooses the method with the lowest cost. Setting this parameter to `FALSE` is recommended by Oracle for Oracle Applications Release 12 on Oracle 10g (see MetaLink white paper Doc ID 396009.1).

session_cached_cursors = 500

`SESSION_CACHED_CURSORS` specifies the number of session cursors to cache. Repeated parse calls of the same SQL statement causes the session cursor for that statement to be moved into the session cursor cache. Subsequent parse calls will find the cursor in the cache and do not need to reopen the cursor. Oracle uses a least recently-used algorithm to remove entries in the session cursor cache to make room for new entries when needed.

The default setting of 0 for this parameter causes cursors to be closed immediately, thus requiring them to be reopened for each subsequent execution.

Setting this parameter to 500 is recommended by Oracle for Oracle Applications Release 12 on Oracle 10g.

4.1.3 Documented Oracle Parameters Removed for Oracle 10gR2

The following standard Oracle parameters have been removed from the parameter file for Oracle 10.2.0.3 and are still not applicable in Oracle 11g:

db_file_multiblock_read_count = 16

This parameter has been removed and left at its default setting because its scope has been redefined between Oracle 9i and Oracle 10g release 2.

`DB_FILE_MULTIBLOCK_READ_COUNT` still specifies the maximum number of blocks read in one I/O operation during a sequential scan. However, as of Oracle Database 10g release 2, the default value of this parameter is a value that corresponds to the maximum I/O size that can be performed efficiently. This value is platform-dependent and is 1MB for most platforms.

Because the parameter is expressed in blocks, it will be set to a value that is equal to the maximum I/O size that can be performed efficiently divided by the standard block size. Note that if the number of sessions is extremely large the multiblock read count value is decreased to avoid having the buffer cache flooded with too many table scan buffers.

Even though the default value may be a large value, the optimizer does not favor large plans if you do not set this parameter. It would do so only if you explicitly set this parameter to a large value. The total number of I/Os needed to perform a full table scan depends on such factors as the size of the table, the multiblock read count, and whether parallel execution is being utilized for the operation.

Removing this parameter is recommended by Oracle for Oracle Applications Release 12 on Oracle 10g.

timed_statistics = TRUE

This parameter has been removed because `TIMED_STATISTICS` is enabled by default if the `STATISTICS_LEVEL` parameter is set to `TYPICAL` (the default) or `ALL`. Since the `STATISTICS_LEVEL` parameter is left at its default setting, `TIMED_STATISTICS` is always set to `TRUE`. As a result, there is no need to explicitly set this parameter.

4.1.4 Undocumented Oracle Parameters

The following undocumented or hidden Oracle parameters have been added to the parameter file for Oracle 10.2.0.3 and are applicable in Oracle 11g:

_allow_level_without_connect_by = TRUE

This parameter setting has been recommended by Oracle support to overcome `ORA-01788` errors in hierarchical queries and queries using the level pseudo column or any query referring to a column as level. You can find details on this workaround in Oracle Doc ID 271939.1 on MetaLink.

Following is a brief summary of the Oracle support document:

```
SELECT level, assembly_id, assembly_name, parent_assembly
FROM bill_of_materials
UNION
SELECT level, assembly_id, assembly_name, parent_assembly
FROM bill_of_materials
START WITH parent_assembly IS NULL
CONNECT BY parent_assembly = PRIOR assembly_id;
fails on 10g and 11g with
ORA-01788: CONNECT BY clause required in this query block
```

Cause

Not using the `CONNECT BY` syntax in each query block is restricted in Oracle 11g because the level pseudocolumn is meaningful only in hierarchial queries. If you are using level without `CONNECT BY`, it is highly likely that you are making a mistake that could lead to unexpected errors.

Solution

In the preceding query, you should include the `CONNECT BY` syntax in each query block. An underscore parameter is provided for backward compatibility in case someone uses level without `CONNECT BY` in code that worked with 9i and cannot fix the code right away. You must set:

```
_allow_level_without_connect_by = TRUE
```

You can also replace level with 0 (or with a different alias) in the query that does not have the `CONNECT BY` clause.

```
_b_tree_bitmap_plans = FALSE
```

This parameter disables the use of bitmap plans for tables that have only B-tree indexes. If this parameter is set to `TRUE` (default setting), the optimizer is allowed to produce bitmap plans for normal b*tree indexes even if no bitmap indexes exist.

Setting this parameter to `FALSE` is recommended by Oracle for Oracle Applications on Oracle 11g.

_fast_full_scan_enabled = FALSE

This parameter disables index fast full scans if only indexes are required to resolve the queries.

Setting this parameter to `FALSE` is recommended by Oracle for Oracle Applications on Oracle 11g.

_gby_hash_aggregation_enabled = FALSE

This parameter setting has been recommended by Oracle support to overcome query performance issues when moving SQL queries from Oracle 9i to Oracle 10g. You can find the details of the proposed workarounds for performance issues related to migrating from Oracle 9i to Oracle 10g in Oracle Doc ID 295819.1 on MetaLink.

Oracle 10g release 2 introduced a feature called Hash Group by aggregation which allows a hash algorithm to process group by statements. This functionality can be disabled by setting the `_GBY_HASH_AGGREGATION_ENABLED` database parameter to `FALSE`.

Oracle support also recommends disabling another 10g feature called costed query transformations in case of performance degradations. 10g adds costed subquery unnesting and view merging functionality. This functionality can be disabled setting the database parameter `_OPTIMIZER_COST_BASED_TRANSFORMATION` to `OFF`.

Disabling costed query transformations may result in consistent `ORA-00600` errors. Therefore, the `_OPTIMIZER_COST_BASED_TRANSFORMATION` parameter will not be used for Model N databases.

_like_with_bind_as_equality = TRUE

Setting this parameter to `TRUE` causes the cost-based optimizer to treat `LIKE` predicates with bind variables as an equality predicate for costing purposes.

This parameter forces the optimizer to treat expressions of the form `[indexed-column like :b1]` similar to `[index-column = :b1]`.

Normally, the CBO assigns internal default selectivity estimates for the `LIKE` operator but this is nowhere near the true selectivity and can cause an index access to be rejected.

Note: Default selectivity has changed from earlier Oracle releases.

Release	Selectivity
earlier than 9.2.X	25%
9.2.X and greater	5%

As of 8.1.7.2 and Oracle9i this parameter also enabled equality costing for expressions of the form:

```
function(column) LIKE function(:bind)
```

Setting this parameter to `TRUE` is recommended by Oracle for Oracle Applications on Oracle 11g.

_optimizer_connect_by_cost_based = FALSE

This parameter disables the use of cost-based transformations for connect by queries. For example, it effectively enforces the rule-based optimizer for such queries. This parameter setting has been recommended by Oracle support as a workaround for bug 5391575. The bug results in an ORA-00600 error for certain hierarchical queries with a connect by construct.

_optimizer_skip_scan_enabled = FALSE

This parameter setting has been recommended by Oracle support as a workaround for bug 5022061 and can be fixed by applying 6932786 Oracle 10.2.0.3 patch 21 bug for Windows Itanium 64bit.

_remove_aggr_subquery = FALSE

This parameter disables removal of subsumed aggregated subqueries. This parameter setting has been recommended by Oracle support as a workaround for bug 5513123. This bug results in an ORA-07445 error and a core dump for update statements on tables with a function based index.

_sort_elimination_cost_ratio = 5

This parameter defines the cost ratio for sort elimination in `first_rows` mode of the CBO. When using an index access plan for a query that has an `ORDER BY` clause, the final sorting can be avoided.

For example, if the value is set to 5, it means that a plan that avoids a sort may not be five times more expensive than a plan that does not avoid it. As a result, the optimizer compares the cost of all queries accordingly and picks the low cost execution plan. A value of 0 (the default setting) would mean that an execution plan with `ORDER BY` sort elimination be chosen even if it is more expensive than queries that do a final sorting.

Setting this parameter to 5 is recommended by Oracle for Oracle Applications on Oracle 11g.

4.1.5 Redo Logs

Model N requires that the "redo" logs be configured to have 250MB log files and the number of redo log groups be set to 6.

4.1.6 Tablespaces and Partitions

This section describes how to create the necessary tablespaces and files for a smaller deployment, for example, no table greater than 1 gigabyte (GB).

The following SQL commands should be executed using a database administration account.

Code 4-1: SQL Syntax

```
CREATE TABLESPACE MN_DATA1
  DATAFILE '/<mount_point>/oradata/<DBInstance>/mn_data101.dbf'
  SIZE 500M AUTOEXTEND ON NEXT 10M MAXSIZE 2048M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;

CREATE TABLESPACE MN_IDX1
  DATAFILE '/<mount_point>/oradata/<DBInstance>/mn_idx101.dbf'
  SIZE 500M AUTOEXTEND ON NEXT 10M MAXSIZE 2048M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;

CREATE TABLESPACE MN_DATA2
  DATAFILE '/<mount_point>/oradata/<DBInstance>/mn_data201.dbf'
  SIZE 200M AUTOEXTEND ON NEXT 10M MAXSIZE 2048M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 256k;

CREATE TABLESPACE MN_IDX2
  DATAFILE '/<mount_point>/oradata/<DBInstance>/mn_idx201.dbf'
  SIZE 200M AUTOEXTEND ON NEXT 10M MAXSIZE 2048M
  EXTENT MANAGEMENT LOCAL UNIFORM SIZE 256k;
```

4.1.7 Users and Roles

The following SQL commands should be executed using a database administrator account.

Create the `MN_RESOURCE` database role, if it does not exist:

Code 4-2: Creating an MN_RESOURCE Database Role

```
CREATE ROLE MN_RESOURCE;

GRANT
    ALTER SESSION
   ,CREATE CLUSTER
   ,CREATE DATABASE LINK
   ,CREATE INDEXTYPE
   ,CREATE MATERIALIZED VIEW
   ,CREATE OPERATOR
   ,CREATE PROCEDURE
   ,CREATE SEQUENCE
   ,CREATE SESSION
   ,CREATE SYNONYM
   ,CREATE TABLE
   ,CREATE TRIGGER
   ,CREATE TYPE
   ,CREATE VIEW
   ,SELECT ANY DICTIONARY
TO MN_RESOURCE;
```

Create a database user using SQL commands such as the following:

Code 4-3: SQL Syntax

```
CREATE USER <DBUser> IDENTIFIED BY <Password>
DEFAULT TABLESPACE <Default_Tablespace>
TEMPORARY TABLESPACE <Default_Temp_Tablespace>;

GRANT MN_RESOURCE, UNLIMITED TABLESPACE
TO <DBUser>;
```

4.1.8 Global Transactions (XA)

This section describes how to enable global transactions on the database server. The steps in this section are required only if the application is deployed on WebLogic and if the JMS server to be used by data flows are also configured on WebLogic.

If you do not perform these steps on the database server, normal XA database queries and updates may work fine. However, if you have not performed these steps, when the WebLogic Server Transaction Manager recovers on a re-boot after a crash, recover for the Oracle resource will fail with `XAER_RMERR`. Crash recovery is a standard operation for an XA resource.

To enable global transactions on the database server:

1. Log on to `sqlplus` as sys user. For example:

   ```
   sqlplus sys/<DBSysPass>@<DBInstance> as sysdba
   ```

2. Execute the following command:

   ```
   @xaview.sql
   ```

3. Execute the following script to grant permissions:

   ```
   PROMPT

   PROMPT Granting XA Permissions ...

   GRANT SELECT ON v$xatrans$ TO public;

   GRANT SELECT ON pending_trans$ TO public;

   GRANT SELECT ON dba_2pc_pending TO public;

   GRANT SELECT ON dba_pending_transactions TO public;

   EXIT;
   ```

4.1.9 Java Stored Procedure Installation

You must have already run your data import before you install the Java stored procedure. See Importing Data for the steps on how to import your data.

4.1.10 Oracle Real Application Clusters

For an Oracle Real Application Clusters (Oracle RAC) configuration, the following other items must be applied:

- When setting up Direct Loader, each node must have a Direct Loader directory mapped under the same name so it can be read from and written to from each node.

- For the web server, the JDBC URL that the application uses to connect to the RAC must be changed. Enter the following JDBC URL in SQL explorer to connect to the RAC.

Code 4-4: JDBC URL for Oracle RAC

```
jdbc:oracle:thin:${myprop.dbuser}/
${myprop.dbpass}@(DESCRIPTION=(ADDRESS_LIST=(ADDRESS=(PROTOCOL=TCP)(HOS
T=hdqpdbuild13.modeln.com)(PORT=1521))(ADDRESS=(PROTOCOL=TCP)(HOST=hdqp
dqadb1.modeln.com)(PORT=1521)))(CONNECT_DATA=(SERVICE_NAME=mndb11g)))
```

where `${myprop.dbuser}` is the user name and `${myprop.dbpass}` is the password.

4.2 Oracle Streams

Oracle Streams captures database changes at a source database, stages the changes, propagates the changes to one or more destination databases, and then applies the changes.

This section of the document provides information regarding the set up and configuration of Oracle Streams.

4.2.1 How to Set Up ARCHIVELOG Mode

The Oracle Streams capture process reads information from the redo log files of the database. To ensure that the information in the redo log files is available for capture, you must run the source database in ARCHIVELOG mode.

To enable database archiving:

1. Shutdown the database by executing:

```
$ sqlplus sys/manager@mndb11g as sysdba

SQL> shutdown immediate
```

2. Start the database in mount mode by executing:

```
SQL> startup mount
```

3. Enable the archive log by executing:

```
SQL> alter database archivelog;
```

4. Start the database in open mode by executing:

```
SQL> alter database open
```

4.2.2 How to Set Up an Oracle Streams Administrator Account

The Oracle Streams environment is managed by an administrative database user. To manage this environment, the administrator needs some specific privileges and must create tables to store Oracle Streams-related information.

For each database participating in the Oracle Streams data sharing environment, you must create a user account and designate it as the Oracle Streams administrator.

Important: Do not use this account for any other purpose, and do not use the SYS or the SYSTEM user as the Oracle Streams administrator.

To create an administrator for Oracle Streams:

1. Create a separate tablespace for storing the objects to be owned by the Oracle Streams administrator. Use the following command to create the tablespace:

   ```
   CREATE TABLESPACE streams_tbs DATAFILE'/home/oracle/oradata/
   mndb11g/streams_tbs.dbf' SIZE 1G;
   ```

2. Create the Oracle Streams administrator user in the database as follows:

   ```
   CREATE USER strmadmin

   IDENTIFIED BY strmadmin

   DEFAULT TABLESPACE streams_tbs

   TEMPORARY TABLESPACE temp

   QUOTA UNLIMITED ON streams_tbs;
   ```

3. Grant the DBA role to this administrator by executing:

   ```
   GRANT DBA TO strmadmin;
   ```

4. Use the GRANT_ADMIN_PRIVILEGE procedure in the DBMS_STREAMS_AUTH package to grant the required privileges to the Oracle Streams administrator:

   ```
   BEGIN

      DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE(

      grantee => 'strmadmin',

      grant_privileges => true);

   END;

   /
   ```

5. Repeat these steps to set up an Oracle Streams administrator on each of the databases participating in the Oracle Streams data sharing environment.

4.2.3 How to Set Initialization Parameters

Specific initialization parameters must be set on the databases participating in the Oracle Streams data sharing. The following table describes these initialization parameters and the values they must be set to.

Table 4-4: Oracle Streams Initialization Parameters

Parameter	Source Database	Destination Database
AQ_TM_PROCESSES	3	3

Table 4-4: Oracle Streams Initialization Parameters (Continued)

Parameter	Source Database	Destination Database
COMPATIBLE	11.2.0.0.0	11.2.0.0.0
JOB_QUEUE_PROCESSES	2 or higher	N/A
STREAMS_POOL_SIZE	200MB minimum	200MB minimum

For the Model N schema, set only the following initialization parameters, as the other parameters are already set during database creation using Model N database creation template.

```
SQL> alter system set AQ_TM_PROCESSES=3 SCOPE=spfile;

SQL> alter system set STREAMS_POOL_SIZE = 209715200 SCOPE=spfile;
```

To make the preceding values effective, restart the database as follows:

```
SQL> shutdown immediate

SQL> startup
```

Repeat the initialization of parameters on the secondary database.

4.2.4 How to Create a Database Link

You must create a database link from the source database to the destination database. In this example, a database link from MNDB11G (the source database) to MNDB11G2 (the destination database).Include the MNDB11G2 entry in the tnsnames.ora file on the source database.

```
CONNECT strmadmin/strmadmin@MNDB11G


CREATE DATABASE LINK MNDB11G2

CONNECT TO strmadmin

IDENTIFIED BY strmadmin

USING 'MNDB11G2';
```

4.2.5 How to Set Up an Oracle Streams Queue

Data moves from the source database to the destination database through queues. Use the SET_UP_QUEUE procedure in the DBMS_STREAMS_ADM package to set up the queues. By default, this procedure creates a queue table named `streams_queue_table` and a queue named `streams_queue`. You can override these names by specifying the `queue_table` and `queue_name` parameters of the procedure SET_UP_QUEUE.

```
CONNECT strmadmin/strmadmin@MNDB11G

EXEC DBMS_STREAMS_ADM.SET_UP_QUEUE();
```

```
CONNECT strmadmin/strmadmin@MNDB11G2

EXEC DBMS_STREAMS_ADM.SET_UP_QUEUE();
```

4.2.6 How to Set Up the Capture Process

The capture process initiates replication by capturing the changes in the source database. It then formats each change into a logical change record (LCR) and placed them into the queues.

A capture process must be created to extract the changes from the redo logs. You can configure a capture process to run on the source database (called a local capture) or remotely on another database (called a downstream capture).

While creating a capture process, you add rules to specify which changes to capture and which ones to discard. The rules are combined into rule sets. The capture process can have a positive rule set and a negative rule set. For a replication capture process to work, you must add a positive rule set that specifies that the data manipulation language (DML) changes as well as the data definition language (DDL) changes be captured. To do this, use the ADD_SCHEMA_RULES procedure of the DBMS_STREAMS_ADM package. In the following example, the procedure call creates the local capture process, named `capture_stream`.

Code 4-5: Local Capture Process with Positive Rule Set

```
Conn strmadmin/strmadmin@mndb11g
BEGIN
  DBMS_STREAMS_ADM.ADD_SCHEMA_RULES(
    schema_name     => 'MN schema name',
    streams_type   => 'capture',
    streams_name   => 'capture_stream',
    queue_name     =>
           'strmadmin.streams_queue',
    include_dml    => true,
    include_ddl    => true,
    inclusion_rule => true);
END;
/
```

where:

- `schema_name` indicates that these rules are associated with the specified schema and.*`MN schema name`* is the name of the Model N schema.
- `streams_type` indicates that this procedure will create a capture process.
- `inclusion_rule` with the value set to *true* means that this procedure will create a positive rule set for the capture process.
- `include_dml` set to *true* means that a rule will be created for data manipulation language (DML) changes.
- `include_ddl` set to *true* means that a rule will be created for data definition language (DDL) changes.

The queue, `streams_queue`, specified in this procedure call was created by the SET_UP_QUEUE procedure call.

The optional parameter `source_database` can be used to specify another database for a downstream capture. If you omit this parameter or set the parameter value to NULL, the procedure will create a local capture process.

For a replication capture process to ignore DML and DDL changes on particular tables and MVs, you need to add a negative rule set that specifies that the data manipulation language (DML) changes as well as the data definition language (DDL) changes not to be captured. To do this, use the ADD_TABLE_RULES procedure of the DBMS_STREAMS_ADM package.

Note: The Model N schema contains MVs that use `rowid` for refresh operations. As the `rowid` data type is not supported for replication, the respective MV is added to a negative rule set.

Code 4-6: Local Capture Process with Negative Rule Set

```
BEGIN
  DBMS_STREAMS_ADM.ADD_TABLE_RULES(
    table_name     => 'MN Schema Name.MN_CMT_MEM_MDL_MV',
    streams_type   => 'capture',
    streams_name   => 'capture_stream',
    queue_name     => 'strmadmin.streams_queue',
    include_dml    => true,
    include_ddl    => true,
    inclusion_rule => false);
END;
/
```

4.2.7 How to Set Up the Propagation Process

Once the changes are captured and placed into the queue, propagate the changes to the destination database. To do so, create a propagation process and associate the source queue with the destination queue. The source and destination queues (each named the default `streams_queue`) were created in the source and destination databases, respectively, in the pevious sections by calling the SET_UP_QUEUE procedure.

The following procedure creates a propagation process and adds rules to the positive rule set of that process.

1. As the Oracle Streams administrator, connect to the source database and execute the following block.

Code 4-7: Propagation Process Example

```
Conn strmadmin/strmadmin@mndb11g

BEGIN
  DBMS_STREAMS_ADM.ADD_SCHEMA_PROPAGATION_RULES(
    schema_name => 'MN schema name',
    streams_name => 'MNDB11G_to_MNDB11G2',
    source_queue_name => 'strmadmin.streams_queue',
    destination_queue_name =>
'strmadmin.streams_queue@MNDB11G2',
```

Code 4-7: Propagation Process Example (Continued)

```
    include_dml => true,
    include_ddl => true,
    source_database => 'MNDB11G',
    inclusion_rule => true);
END;
/
```

This procedure call creates a propagation process, named `MNDB11G_to_MNDB11G2`, for which the source queue is the `streams_queue` in the MNDB11G database and the destination queue is the streams_queue in the MNDB11G2 database. This procedure also adds DML and DDL rules to the positive rule set.

2. Create the destination schema. Before you can start replicating the DDL and DML changes on a source database to the destination database, the schema must exist in the destination database. If the schema doesn't already exist in the destination database, you can export and import the schema using the Oracle Data Pump.

4.2.8 How to Set Up the Apply Process

The apply process must be created on the destination database, associate the destination queue with it, and add rules for the apply process. To achieve this, call the ADD_SCHEMA_RULES procedure of the DBMS_STREAMS_ADM package.

1. As the Oracle Streams administrator, connect to the destination database and execute the following block.

Code 4-8: Apply Process Example

```
Conn strmadmin/strmadmin@mndb11g2

BEGIN
  DBMS_STREAMS_ADM.ADD_SCHEMA_RULES(
    schema_name      => 'MN schema name',
    streams_type     => 'apply',
    streams_name     => 'apply_stream',
    queue_name       =>
            'strmadmin.streams_queue',
    include_dml      => true,
    include_ddl      => true,
    source_database => 'MNDB11G',
    inclusion_rule   => true);
END;
/
```

This procedure creates an apply process (as indicated by the `streams_type` parameter) for the given schema. The apply process, named `apply_stream`, is associated with the `streams_queue`. The apply process also adds DML and DDL rules to the positive rule set (as indicated by the parameter inclusion_rule).

4.2.9 How to Start the Capture and Apply Processes

Now that all of the necessary objects and processes are configured, start the capture and apply processes.

To start the capture process, connect to the source database and execute the START_CAPTURE procedure of the DBMS_CAPTURE_ADM package:

Code 4-9: Starting the Capture Process

```
Conn strmadmin/strmadmin@mndb11g

BEGIN
  DBMS_CAPTURE_ADM.START_CAPTURE(
    capture_name  =>
                'capture_stream');
END;
/
```

Similarly, to start the apply process, connect to the destination database and execute the START_APPLY process of the DBMS_APPLY_ADM package.

Recommendation: Before starting the apply process, best practices recommend that you set the `disable_on_error` parameter of the apply process to *n*, so that the apply process will continue even if it encounters some errors.

Code 4-10: Starting the Apply Process

```
Conn strmadmin/strmadmin@mndb11g2
BEGIN
  DBMS_APPLY_ADM.SET_PARAMETER(
    apply_name  => 'apply_stream',
    parameter   => 'disable_on_error',
    value       => 'n');
END;
/

BEGIN
  DBMS_APPLY_ADM.START_APPLY(
    apply_name  => 'apply_stream');
END;
/
```

The Oracle Streams replication environment is ready, and DML and DDL changes in the source database schema will be replicated in the corresponding destination database schema.

4.3 Managing Data

This section describes how to import data and how to analyze data statistics.

4.3.1 Importing Data

To import data:

1. Drop and recreate the users using the script in Users and Roles.
2. An Oracle database dump can be imported using either the Oracle Data Pump utility (impdp) or the older Import utility (imp).

Note: The database tool used to perform the import must be the same as the tool used to export the database dump. For example, if the database dump you're attempting to import was created using the Data Pump, then you must use the "impdp" utility to import the database.

To import the database dump using the Data Pump utility:

a. Transfer the database dump to the database server. Make note of the directory on the Database Server where the database dump was placed. The next step will grant Oracle permission to read and write to that location.

b. Grant Oracle permission to write the dump file to the file system on the database server. The following SQL command creates an alias that points to a directory on the file system. The Data Pump then refers to that alias when exporting the file rather than referring to the actual directory location.

```
sqlplus system/password@<DBInstance>

DROP DIRECTORY mydiralias;
CREATE DIRECTORY mydiralias AS '/path/to/directory';
GRANT READ, WRITE ON DIRECTORY mydiralias TO dbuser;
```

c. Perform the Data Pump export (this can be run from a remote host).

```
impdp system/password@<DBInstance>
      schemas=<FromDBUser>
      directory=mydiralias
      dumpfile=Exp_<FromDBUser>.dmp
      skip_unusable_indexes=y
      table_exists_action=skip
      remap_schema=<FromDBUser>:<DBUser>
      transform=oid:n
```

To import the database dump using the Import utility:

```
$ imp userid=system/manager@<DBInstance>
      file=Exp_<FromDBUser>.dmp
      fromuser=<FromDBUser>
      touser=<DBUser>
      commit=N
      indexes=N
      buffer=1048576
```

3. If the dump file contained materialized views, the import generates warnings while creating the materialized views and logs. If errors occur due to materialized

views, to delete the materialized views once the database has been imported, execute the following stored procedure:

```
SQL> Execute mn_global_pkg.drop_mat_views;
```

For some Model N applications, you must implement functionality on the database server using Java stored procedures. These applications include Sales Submissions, Price Master, and Promotions (Order-Quantity Discounts or OQD). The Java classes used by Java stored procedures for these applications must be deployed to the database server.

This section describes the steps required to deploy Model N Java classes to an Oracle database server. The Oracle JVM must be installed on the database server before you can create Model N stored procedures. See JVM, for details regarding installation of the Oracle JVM.

After confirming that the Oracle JVM has been installed on the database server, install the Model N stored procedures using the following steps:

1. Run a a `DBPostImport` operation as described in Operations Tools to ensure that the database schema is up-to-date and has been synchronized with the current application deployment.
2. For applications using Java stored procedures, load the Oracle JAR files into the database using the Oracle `loadjava` command.

   These JAR files contain Model N Java classes and property files. The Java class files have been compiled using the Oracle JDK 1.3 compiler to ensure compatibility with the Oracle JVM. The command line syntax for loading (including the JAR file name) and the contents of the JAR file are described per application in the following section.

Important: You must load `oracle-platform.jar` first before you can load the other two JAR files.

---

d. For the Sales Submissions application, execute the following command:

```
loadjava -schema <DBUser> -user <DBUser>/
</DBPass>@<DBInstance>
-force -resolve -verbose oracle-platform.jar
```

The Sales Submissions data flows use a new integration channel called the Direct Loader to load large volume sales data with high throughput. For additional details on installing and configuring the Direct Loader, see Configuring Direct Loader. The Platform JAR contains the Java stored procedures used by the Direct Loader.

e. For the Price Master application, execute the following command:

```
loadjava -schema <DBUser> -user <DBUser>/
</DBPass>@<DBInstance>
-force -resolve -verbose oracle-pricemaster.jar
```

The Price Master JAR file contains the following classes and property files:

`com.modeln.bp.pm.resolution.*`

The Price Master JAR file contains the following property files:

`pm_*.properties`

`switches/pm_*.properties`

f. For the Promotions (Order-Quantity Discounts or OQD) application, execute the following command:

```
loadjava -schema <DBUser> -user <DBUser>/
</DBPass>@<DBInstance>
-force -resolve -verbose oracle-oqd.jar
```

The Promotions JAR file contains both promotion and price master classes and property files because the Promotions application introduces new stored procedures that compare prices across price master and promotions to return the lowest prices. The following list describes the contents of the Promotions JAR file.

Java Classes:

`com.modeln.bp.pm.resolution.*`

`com.modeln.bp.promotion.resolution.*`

`com.modeln.bp.prc.CMnOraclePricingServer`

Properties:

`pm_*.properties`

`promotion_*.properties`

`switches/pm_*.properties`

In addition to the standard JAR files provided for these Model N applications, you may need to create JAR files for custom application extensions to Model N and load them on the database server as in Java Stored Procedure Installation. Such JAR files may contain configuration properties (in property files) with overridden values or extended functionality (in Java) required by custom application extensions.

3. Grant access permission to the application user to execute the Java stored procedures.

   The permissions must be granted by an Oracle user with administrative access to the database permission tables. The following stored procedure can be used to grant these access permission rights:

```
SQL> call dbms_java.grant_permission(
        '<DBUser>',
        'SYS:java.lang.RuntimePermission',
        'getClassLoader',
        '');
```

The stored procedures are now installed and can be utilized by the application.

Note: This grant of runtime permission applies to Java stored procedures for all Model N applications.

---

4. Recompile all PL/SQL packages. You can do this using SQL Plus as follows:

```
$ sqlplus <DBUser>/</DBPass>@<DBInstance>
SQL> exec dbms_utility.compile_schema( '<DBUser>' );
```

4.3.2 Exporting Data

To export data:

1. Drop the materialized views using the following stored procedure:

```
$ sqlplus <DBUser>/<DBPass>@<DBInstance>
SQL> call MN_GLOBAL_PKG.DROP_MAT_VIEWS();
```

2. An Oracle database dump can be exported using either the Oracle Data Pump utility (expdp) or the older Export utility (exp).

   To export the database dump using the Data Pump utility:

   a. Grant Oracle permission to write the dump file to the file system on the database server. The following SQL command creates an alias that points to a directory on the file system. The Data Pump then refers to that alias when exporting the file rather than referring to the actual directory location.

```
sqlplus system/password@<DBInstance>

DROP DIRECTORY mydiralias;
CREATE DIRECTORY mydiralias AS '/path/to/directory';
GRANT READ, WRITE ON DIRECTORY mydiralias TO dbuser;
```

   b. Perform the Data Pump export (can be run from remote host)

```
expdp system/password@dbinstance schemas=dbuser
directory=mydiralias dumpfile=Exp_dbuser.dmp
exclude=statistics version=10.2
```

   The version attribute lets you preserve backward compatibility for the dump if you need to export an 11g dump and import it into a 10g instance.

   To export the database dump using the Export utility:

```
$ exp userid=system/manager@<DBInstance>
    file=Exp_<FromDBUser>.dmp
    owner=<DBUser>
    grants=N
    compress=N
    recordlength=8192
```

4.3.3 Gathering Schema Statistics

After you have imported the data, analyze the schema to provide statistics to the Cost-Based Optimizer. It is recommended that you run the `DBMS_STATS` package once every week and immediately following any significant data loads. The syntax to compute statistics is shown below:

```
SQL> Execute DBMS_STATS.GATHER_SCHEMA_STATS(
            ownname => '<DBUser>',
            estimate_percent => DBMS_STATS.AUTO_SAMPLE_SIZE,
            degree => DBMS_STATS.DEFAULT_DEGREE,
            cascade => TRUE);
```

Statistics must never be collected on global temporary tables. To find out if statistics exist for a global temporary table open a SQL*Plus session, connect to the Model N schema and run the following query:

```
select table_name, num_rows, last_analyzed
from tabs
where temporary = 'Y' ;
```

If the values returned in the `num_rows` or `last_analyzed` columns are not null for any table, you must delete that table's statistics. You can delete statistics with the following script:

```
DECLARE
  l_user VARCHAR2(30);
BEGIN

  SELECT user
    INTO l_user
    FROM dual;

  FOR c_rec in (SELECT table_name
                  FROM user_tables
                 WHERE temporary='Y'
                   AND last_analyzed IS NOT NULL) LOOP

   dbms_stats.delete_table_stats
     ( ownname=> l_user,
       tabname=> c_rec.table_name,
      cascade_indexes=>TRUE );
  END LOOP;
END;
/
```

4.3.4 Improving Performance

Caching small tables in the Oracle database has shown to improve performance. Within `tools.jar`, you can find the following scripts:

- default_cache.sql
- keep_cache.sql
- recycle_cache.sql

Perform the following steps:

1. Login as user sysdba.
2. ALTER SYSTEM SET sga_max_size = 6G SCOPE=SPFILE;
3. Restart the database.
4. Login as user sysdba.
5. ALTER SYSTEM SET DB_KEEP_CACHE_SIZE=1G SCOPE=both;
6. ALTER SYSTEM SET DB_RECYCLE_CACHE_SIZE=2G SCOPE=both;
7. ALTER SYSTEM SET DB_CACHE_SIZE=1G SCOPE=both;
8. Restart the database.
9. Login as user mn53ph.
10. Run `keep.sql`
11. Run `recycle.sql.`

The script `default_cache.sql` resets the buffer cache from KEEP to DEFAULT for tables when the tables grow larger than 1000 blocks.

5 Configuring Direct Loader

This chapter describes how to install the Direct Loader channel and includes a frequently asked questions section. The Direct Loader channel is an integration channel that loads large volumes of data into the system through flat files using Oracle's external table feature.

5.1 Direct Loader Channel Configuration

This section contains information on how to configure the Direct Loader channel.

5.1.1 How to Configure the Direct Loader Channel

To configure the Direct Loader channel:

1. Load the Java stored procedures.

   For information on how to load these procedures, refer to the instructions in Java Stored Procedure Installation.

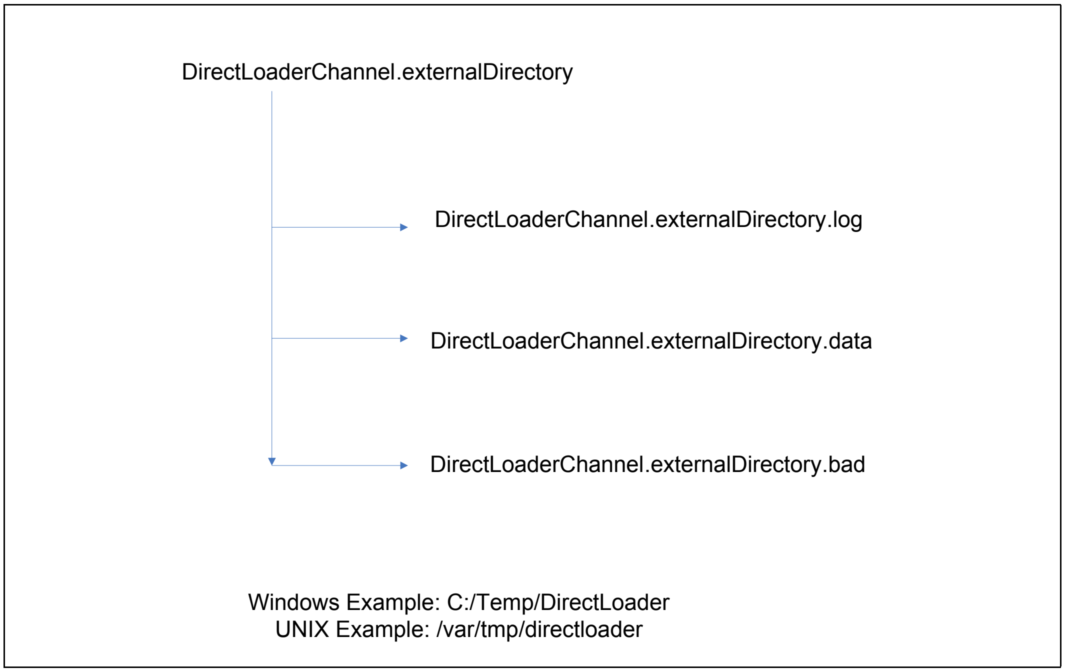
2. Create a directory on the database server to be used by the Direct Loader. If the database is deployed on a UNIX system, grant all file permissions for this directory to the Oracle process.

3. Set the absolute path to the directory you have just created as the value for property, `com.modeln.DirectLoaderChannel.externalDirectory`.

4. The data, the log, and the bad subdirectories default to `directLoaderData`, `directLoaderLog`, and `directLoaderBad` respectively. The Direct Loader creates these subdirectories automatically. Therefore, you do not need to create them manually. You can override the default values for these directories using the following properties if you are not using the default values for the sub directories:

```
com.modeln.DirectLoaderChannel.externalDirectory.data
com.modeln.DirectLoaderChannel.externalDirectory.log
com.modeln.DirectLoaderChannel.externalDirectory.bad
```

Figure 5-1: Directory Structure of the Direct Loader Directories on the Database Server


DirectLoaderChannel.externalDirectory
DirectLoaderChannel.externalDirectory.log
DirectLoaderChannel.externalDirectory.data
DirectLoaderChannel.externalDirectory.bad
Windows Example: C:/Temp/DirectLoader
UNIX Example: /var/tmp/directloader

5. Grant permissions to the database user to access the directories on the database server either:
   - by executing scripts on the database server

     OR
   - by using DBPostImport

How to Execute Scripts on the Database Server

The scripts are located in the scripts directory, which is described in the section on Application Server Support Files. A DBA must run the scripts to grant permissions to the database.

a. Login as SYSTEM user on the database.

   For example: `sqlplus system/<SysPassword>@<DBName>`

b. Execute `directloader_proc.sql`.

   For example: `SQL> @directloader_proc.sql`

c. Execute the stored procedure that you just created with appropriate arguments based on your configuration settings.

For a Windows Database:

```
SQL> EXECUTE mn_direct_loader_setup('<DBUser>',
'<FTPHost>','C:\modeln\directLoader','directLoaderData',
'directLoaderLog','directLoaderBad','\');
SQL> commit;
SQL> exit;
```

For a UNIX Database:

```
SQL> EXECUTE mn_direct_loader_setup('<DBUser>', '<FTPHost>','/
opt/modeln/directLoader','directLoaderData',
'directLoaderLog','directLoaderBad','/');
SQL> commit;
SQL> exit;
```

Argument description:

- Arg 1: Database user name
- Arg 2: FTP Server (com.modeln.DirectLoaderChannel.ftp.server)
- Arg 3: Existing directory on the database
- (com.modeln.DirectLoaderChannel.externalDirectory)
- Arg 4: Data sub-directory
- (com.modeln.DirectLoaderChannel.externalDirectory.data)
- Arg 5: Log sub-directory
- (com.modeln.DirectLoaderChannel.externalDirectory.log)
- Arg 6: Bad sub-directory
- (com.modeln.DirectLoaderChannel.externalDirectory.bad)
- Arg 7: File separator on the database server ('/' for a Unix system, '\' for a Windows system)

How to Use DBPostImport

You can choose to allow DBPostImport to automatically grant the required permissions to the database user by setting the following properties and then running DBPostimport operation as described in Operations Tools:

`a. com.modeln.db.allowSysScripts=true`

Set this property to `true` if scripts are allowed to be executed as SYSTEM user. If scripts are not allowed to be executed as SYSTEM user, set this property to `false` and do not set the following properties that are listed in this section. Go to After the database permissions have been granted, restart the application server.

`b. com.modeln.db.sysPassword=Manager`

This property determines the database SYSTEM user password.

`c. com.modeln.DirectLoaderChannel.databaseFileSeparator=/`

This property determines the file separator on the database server. Set the value to / if the database server is installed on a Unix system. Set the value to `\\` if the database server is installed on a Windows system.

6. After the database permissions have been granted, restart the application server.

   If the Model N application is deployed in a clustered environment, all of the application servers participating in the cluster need to be restarted.

5.1.2 How to Configure Direct Loader in FTP Mode

You can configure the Direct Loader in FTP mode in order to get a better performance while loading large content files through the Direct Loader. Note that configuring the Direct Loader in FTP mode is optional.

In FTP mode, the Direct Loader transfers the content file from the application server to the FTP server from where the database picks up the file (through FTP) and writes it to the database directory.

1. Setup and configure the FTP server.

   Install an FTP server that can be used by the Direct Loader to transfer content files to the database server.

   To run the Direct Loader in FTP mode, set the following properties:

   - To use the FTP server, set the following properties:

     `com.modeln.DirectLoaderChannel.requiresFTP=true`

     `com.modeln.FileChannel.streamContent=false`

   - For the FTP server, set `com.modeln.DirectLoaderChannel.ftp.server=<FTPHost>`.
   - For the FTP user name, set `com.modeln.DirectLoaderChannel.ftp.user=<FTPUser>`.
   - For the FTP user password, set `com.modeln.DirectLoaderChannel.ftp.password=<FTPPass>`.
   - For the FTP server port, specify a port if it is not the default FTP port which is 21. To specify a different port, set `com.modeln.DirectLoaderChannel.ftp.port=<FTPPort>`.
   - The FTP server type defaults to auto as follows: `com.modeln.DirectLoaderChannel.ftp.serverType=auto`

5.2 Out-of-the-Box Configuration Settings

This section contains sample out-of-the-box configuration settings for the Direct Loader on Windows and on UNIX.

5.2.1 Windows Database

The following sample shows the out-of-the-box configuration settings for the Direct Loader on Windows. The sample contains code section notes (bold and in parentheses) that are described following the sample:

```
# Directory setting on the database server
com.modeln.DirectLoaderChannel.externalDirectory=
C:\\modeln\\directLoader (1)
com.modeln.DirectLoaderChannel.databaseFileSeparator=\\ (2)

# Settings to allow DBPostImport to configure directloader
com.modeln.db.allowSysScripts=true (3)
com.modeln.db.sysPassword=<DBSysPass> (4)

com.modeln.DirectLoaderChannel.deleteExtFile=true
```

Code Section Notes:

(1) Make sure this directory is created on the database server. Note the double back slashes `\\` in the directory path. Also, be sure to use the Windows separator `\\` and not the UNIX file separator. Also note that the directory path must be an absolute path and not a relative path.

(2) Set the database file separator as `\\` for a Windows database server. Note that the value is a double back slash `\\` instead of a single slash.

(3) If you set this value to `false`, you need to execute the scripts provided in Direct Loader Channel Configuration by logging in to the database as system user.

(4) Set the database system user password if you set the previous flag to `true` allowing DBPostImport to run the SQL scripts for granting the required permissions to the application database user.

5.2.2 UNIX Database

The following sample shows the out-of-the-box configuration settings for the Direct Loader on Windows. The sample contains code section notes (bold and in parentheses) that are described following the sample:

```
# Directory setting on the database server
com.modeln.DirectLoaderChannel.externalDirectory=/opt/modeln/
directLoader (1)
com.modeln.DirectLoaderChannel.databaseFileSeparator=/ (2)



# Settings to allow DBPostImport to configure directloader
com.modeln.db.allowSysScripts=true (3)
com.modeln.db.sysPassword=<DBSysPass> (4)

com.modeln.DirectLoaderChannel.deleteExtFile=true
```

Code Section Notes:

(1) Make sure that this directory is created on the database server. Make sure that the Oracle user has read and write permissions on this directory. Note that the directory path is an absolute path and not a relative path.

(2) Set the database file separator as / for a UNIX database server.

(3) If you set this value to `false`, you need to execute the scripts provided inDirect Loader Channel Configuration by logging in to the database as system user.

(4) Set the database system user password if you set the previous flag to `true` allowing DBPostImport to run the SQL scripts for granting the required permissions to the application database user.

5.3 Verifying Setup

Once the Direct Loader setup is complete, you can use the installation verification process (see Installation Verification on page 187 for more information)to verify the setup. Ensure that all verification steps under the “Direct Loader Verification” group passed. Cross-check the setup steps if any verification step failed.

5.4 Frequently Asked Questions

This section contains a few of the typical questions that users may have about the Direct Loader.

How do I determine if there was an error while a data flow that uses the Direct Loader was running?

The Direct Loader is configured as an outbound channel in data flow configurations. Therefore, all errors originating from the Direct Loader are logged as Outbound Channel Errors in the data flow result.

Go to **DataFlow Result Detail > Dataflow Results Messages** to find the outbound channel error messages. If you see an outbound channel error message, you will need to drill down into the message to diagnose the problem.

What does it mean when the Outbound Channel Error says the following: "*Error writing to outbound channel for dataunit: Max retries for write exceeded.*" ?

Data flows are configured to re-try a certain number of times in case any exception occurs during channel processing. When the data flow reports this error, something has caused the data flow to re-try writing to the channel, but it failed repeatedly due to some error condition. You need to drill down into the message detail by clicking on the exception link in the Messages page. Scroll down on the Exception page and look for the original exception under the related `CMnException` section.

How do I resolve the issue when an exception similar to the following exception is reported in the related `CMnException` section described in the previous question?

```
execute(): SQL=|{? = call
mn_direct_loader_pkg.<stored_proc_name>(?,?)}| …
Detailed exception:
```

```
java.sql.SQLException: ORA-29532: Java call terminated by
uncaught Java exception:
java.security.AccessControlException: the Permission
(java.io.FilePermission /tmp/log/Sale-one15_csv.bad read) has
not been granted to MNPH. The PL/SQL to grant this is
dbms_java.grant_permission( 'MNPH', SYS:java.io.FilePermission',
'/tmp/log/Sale-one15_csv.bad', 'read' )
```

When you see an `AccessControlException`, you should verify the following to make sure that the Direct Loader is set up correctly with the required database permissions:

- Check that the directory to which the error message is referring exists on the database server. For instance, in the previous example, you should check that the directory `/tmp/log` exists on the database server.
- Verify that the `com.modeln.DirectLoaderChannel.externalDirectory` property is set up correctly and points to an existing directory on the database server.
- Verify that the required `FilePermissions` have been granted to the database user by either running DBPostImport (refer to Direct Loader Channel Configuration for setting it up for Direct Loader scripts) or by manually running the scripts (refer to Out-of-the-Box Configuration Settings).
- After the database permissions have been granted, restart the application server to ensure that the connection pool used by the application creates connections that have the updated permissions information. If the Model N application is deployed in a clustered environment, all of the application servers participating in the cluster need to be restarted.

How do I resolve the issue when an exception similar to the following exception is reported in the related `CMnException` section described in the second question?

```
SQL=|{call mn_sale_validation_pkg.validate_distributor_price(?,?,?)}|
Arguments=1:|[SQLType: 2,IN]2|, 2:|[SQLType: 2,IN]46981|, 3:|[SQLType:
2,IN]1|
SQLException : Message = ORA-29541: class
HCLAPP13_MNMD_TAMRC4.com/modeln/bp/pm/resolution/CMnOraclePriceMaster
could not
be resolved
ORA-06512: at "HCLAPP13_MNMD_TAMRC4.MN_PRICE_MASTER_PKG", line 0
ORA-06512: at "HCLAPP13_MNMD_TAMRC4.MN_SALE_VALIDATION_PKG", line 918
ORA-06512: at "HCLAPP13_MNMD_TAMRC4.MN_SALE_VALIDATION_PKG", line 1214
ORA-06512: at line 1
```

Make sure that you have followed the instructions described in Java Stored Procedure Installation.

Why does the data flow fail with the following error message: *Missing IN or OUT parameter at index*?

The application server is using the wrong Oracle JDBC Driver. Verify that the classpath for the server is set up correctly and includes the `ojdbc6.jar` provided by Model N along with application server support files.

To verify the JDBC Driver version that the server is using, look at the server logs.

6

Deploying with Oracle WebLogic Server

This chapter describes how to install the Model N enterprise application under WebLogic. Make sure that the Oracle set-up has been performed prior to this so that the Model N application database is available prior to starting the WebLogic server.

Note: Model N does not support WebLogic clustering.

6.1 Configuring the WebLogic Environment

This section explains how to create the WebLogic environment under which you will install the Model N application. For these instructions to work, you must have already installed WebLogic 11g (10.3.2) and configured the WebLogic Administration Server.

6.1.1 How to Configure the Environment

To configure the WebLogic environment:

1. Start the WebLogic management server.
2. Enter Edit mode by clicking the Lock & Edit button in the Change Center.

3. Click **<yourDomain> > Environment > Servers** in the left navigation pane.
4. Click New under Servers in the right navigation pane.
5. Enter the server name in the Server Name field.

   Use this name to refer to the Model N application instance from now on.
6. Leave the Server Listen Address field blank.
7. Enter the listen port in the Server Listen Port field.
8. Make sure that you indicate that the server is stand-alone and not in a cluster.
9. Click Next.
10. Click Finish.

    The Summary of Servers page opens.

    To allow WebLogic Server to connect to the Node Manager, you must create a machine.
11. In the left panel, under Environment > Machines, click New.
12. Give your machine a name, then click **OK**.

    The Summary of Machines page opens.
13. In the right panel, drill down to the machine name.
14. Select servers.
15. Select **Add** to add a server.
16. Select the Select an existing server option.
17. Select the existing server that you just created in Enter the server name in the Server Name field. Then, click **Next**.
18. In the right panel, click the name of the sever that you have just created.
19. Select Server Start.
20. Set Java Home.

    For information on which JDK to use, see the Supported Configurations document.
21. Set the Java Vendor.
22. Set WebLogic Home. For example:

    `<WL_HOME>`
23. Configure the JVM initialization properties.

    The JVM initialization properties are defined under the Server Start subtab located under the Configuration tab on the Server Settings page.

    For Solaris, the following arguments go with the Sun JDK:

```
Class Path:
    <ExtDir>/props;
   <ExtDir>/jdbc/ojdbc6.jar;
   <ExtDir>/lib/streambuffer.jar;
   <ExtDir>/lib/jaxb-impl.jar;
   <ExtDir>/lib/jaxws-api.jar
   <WL_HOME>/wlserver_10.3/server/lib/weblogic.jar;
   <ExtDir>/lib/sjsxp.jar;
   <WL_HOME>/<JDK>/lib/tools.jar
Arguments:
   -server -d64
   -ms1536m
   -mx4096m
   -XX:MaxPermSize=256m
   -Dcom.modeln.propFile=local_weblogic
   -Dcom.modeln.jndiCtxFactory=
         com.modeln.infr.env.config.CMnPropCtxFactory
-Djava.net.preferIPv4Stack=true
-Dfile.encoding=UTF-8
-verbose:gc -Xloggc:/<temp_directory>/gc.log
```

For Linux and Windows, Model N recommends the following arguments go with the JRockit JDK

```
Class Path:
   <ExtDir>/props;
   <ExtDir>/jdbc/ojdbc6.jar;
   <ExtDir>/lib/streambuffer.jar;
   <ExtDir>/lib/jaxb-impl.jar;
   <ExtDir>/lib/jaxws-api.jar
   <WL_HOME>/wlserver_10.3/server/lib/weblogic.jar;
   <ExtDir>/lib/sjsxp.jar;
   <WL_HOME>/<JDK>/lib/tools.jar
<JROCKIT HOME>/lib/tools.jar
Arguments:
   -server
   -ms1536m (1)
   -mx4096m (2)
   -Dcom.modeln.propFile=local_weblogic
 -Dcom.sun.management.jmxremote
   -Dcom.modeln.jndiCtxFactory=
         com.modeln.infr.env.config.CMnPropCtxFactory
-Djava.net.preferIPv4Stack=true
-Dfile.encoding=UTF-8
-Xverbose:gc -XverboseLog:<log_directory>/gc.log (3)
-Djrockit.oomdiagnostics=true
-Djrockit.oomdiagnostics.filename=<path to file>
```

(1) The initial heap size for a 64-bit system is `-ms1536`. The initial heap size for a 32-bit system is `-ms1400`.

(2) The maximum heap size for a 64-bit system is `-mx4096`. The maximum heap size for a 32-bit system is `-mx1400`.

(3) If you want to track garbage collection, add these arguments. These arguments are vendor-specific.

Note: If you are using JRockit, use `-Xverbose:gc` and `-XverboseLog:/<log_directory>/gc.log`. If you are using SUN's JVM, use `-verbose:gc` and `-Xloggc:/<log_directory>/gc.log`. These are not required settings.

---

24. Set up the remaining managed server properties according to your enterprise requirements.
25. Click Save.
26. Click Activate Changes.
27. Click Lock and Edit.

6.1.1.1 How to Create a JDBC Data Source

1. To create a JDBC connection pool for the application instance, click **<yourDomain> > Services > JDBC > Data Sources** in the left navigation pane and then click New in the right navigation pane.
2. Configure the JDBC connection pool using the following settings:

Name	**<AppName>**`DataSource`
JNDI Name	**<AppName>**`DataSource`
Database Type	`Oracle`
Database Driver	`Oracle's Driver (Thin) for Instance connections; Versions: 9.0.1, 9.2.0,10,11`

3. Click **Next**.
4. Specify the following transaction options:

Supports Global Transactions	`Enabled`
Emulate Two-Phase Commit	`Enabled`

5. Click **Next**.
6. Specify the following connection properties:

Database Name	`<DBInstance>`
Host Name	`<DBHost>`
Port	`<TNSPort>`

Database User Name	`<DBUser>`
Password	`<DBPass>`
Confirm Password	`<DBPass>`

7. Click **Next**.
8. To test the data source configuration, click **Test**.

   The Driver Classname should auto populate to `oracle.jdbc.OracleDriver`.

   The JDBC URL should autopopulate to `jdbc:oracle:thin:@`**`<DBHost>:<TNSPort>:<DBInstance>`**
9. Add the following JDBC property in the Properties text box:

   ```
   protocol=thin
   ```
10. Click Next to select a target server.
11. Target the new Connection Pool to your new Server instance: **`<AppName>`**
12. Click Finish.

    The Summary of JDBC Data Sources page appears.
13. Click the name of the data source that you have just created.
14. Set the following connection pool properties in the Connection Pool tab:

Initial Capacity	`20`
Maximum Capacity	`40`
Capacity Increment	`1`
Statement Cache Type	`LRU`
Statement Cache Size	`0`

15. Click Save.
16. Expand the Advanced portion of the page and set the following properties, which appear as check boxes in the WebLogic Server Administration Console:

Test Connections On Reserve	`Enabled`
Ignore In-Use Connections	`Enabled`
Remove Infected Connections	`Enabled`

17. Click Save.

18. Select the Diagnostics subtab under the Configuration and set the following properties:

Profile Connection Usage	`Disabled`
Profile Connection Leak	`Disabled`

19. Click Save.

6.1.2 Enabling the WebLogic Server Plug-In

To obtain the correct client IP address (browser address) through the WebLogic plug-in for IIS, you need to set the `WeblogicPluginEnabled` property on the WebLogic Server. Otherwise, the client IP address is always 127.0.0.1. The correct IP address is needed for the Performance Monitoring Application (PMA) session tracker.

For more information, see the WebLogic documentation for Web Server Plug-Ins on Oracle's documentation site at:

http://www.oracle.com/technology/documentation/index.html

To enable the WebLogic Server plug-in:

1. Login to the WebLogic Administration Server.
2. In the left hand navigation, click Servers under Domain.
3. If you are running WebLogic Server 9.2, click the Lock & Edit button.
4. Click the managed server you have created.
5. If you are running WebLogic Server 8.x, click Advanced at the bottom of the page.
6. If you are running WebLogic Server 9.x, click Show under Advanced at the bottom of the page.
7. Click WebLogic Plug-In Enabled.
8. Save your changes.

6.1.3 Setting Transaction Time out

1. Under **Services > JTA**, set the time out seconds to 2147483647 seconds.
2. Click Save.
3. Click Activate Changes.

6.1.4 Configuring the JMS Channel

The steps in this section are required only if you have data flows that use the JMS channel and if the JMS server is configured on WebLogic.

6.1.4.1 How to Create a JDBC Data Source for Connections Participating in Global Transactions

The connection pool within this data source will be used by the Model N application to obtain connections that participate in global transactions with the JMS server configured on WebLogic.

In order to take advantage of two-phase commits involving the JMS server and theModel N database, which ensures data consistency between Model N and the JMS server, Model N uses global transactions in data flows when interacting with the JMS channel and the Model N database.

Note: Global transactions in data flows are currently supported only on WebLogic with the JMS server on WebLogic.

To create a JDBC data source for global transactions:

1. Create a data source as described in How to Create a JDBC Data Source on page 90, but with the following values:
    - **Database Driver:** Oracle's Driver (Thin XA) for Instance connections; Versions:9.0.1,9.2.0,10,11
    - The driver class should auto-populate to oracle.jdbc.xa.client.OracleXADataSource.
    - On the **Configuration > Connection Pool** tab, set the following values:

Initial Capacity	`1`
Maximum Capacity	`20`
Capacity Increment	`2`
Statement Cache Type	`LRU`
Statement Cache Size	`0`

    - The data source name and pool name should be `<AppName>XADataSource` and `<JDBCName>XAPool` respectively.

6.1.4.2 How to Create a JMS Connection Factory

This section describes how to create a JMS connection factory.

1. Create a JMS connection factory that will be used by data flows that use the JMS channel.
    a. Click **Services > Messaging > JMS Modules**.
    b. Click **New** to create a new module.
    c. Enter a name for the module. For example:

    `<AppName>SystemModule`

    d. Click **Next**.

   e. Target the server by selecting the check box next to the server: `<AppName>`
   f. Click **Next**.
   g. Choose to add resources to the JMS Module.
   h. Click **Finish**.
   i. Under the Summary of Resources table, click **New**.
   j. Select **Connection Factory**.
   k. Click **Next**.

2. Configure the connection factory using the following settings:

Name	`<AppName>ConnectionFactory`
JNDI Name	`<AppName>ConnectionFactory`

3. Click **Finish**.
4. In the Summary of Resources table, click the name of the resource you just created.
5. In the **Configuration > Default Delivery** tab, verify that **Default Deliver Mode** is set to `Persistent`.
6. In the **Configuration > Transactions** tab, make sure that the **XA Connection Factory Enabled** check box is selected.
7. Click **Save**.

6.1.4.3 How to Create a Message Persistent Store

1. Create a JMS file store for storing persistent messages by clicking **Services > Persistent Stores**.
2. Click **New**.
3. Select **Create FileStore** from the drop-down list.
4. Name the file store `<AppName>JmsStore.`
5. Choose the `<AppName>`server target from the **Target** drop-down list.
6. Create a directory on your server to be used by the JMS File Store. For example:

   `/modeln/jms/<AppName>JmsStore`
7. In the **Directory** field, enter this directory path.
8. Click **Finish**.

   The Summary of Persistent Stores page appears.
9. Click the name of the file store that you just created.
10. Click **Advanced** on the **Configuration** tab.
11. In the **Synchronous Write Policy** drop-down list, select `Cache-Flush`.
12. Click **Save**.

6.1.4.4 How to Create a Message Paging Store

1. Create a JMS file store as a message paging store.
    a. Go to **Services > Persistent Stores**.
    b. Click **New**.
    c. Select **Create FileStore** from the drop-down list.
2. Enter a name.
3. Create a directory on your server to be used by the JMS File Store. For example:

   `/modeln/jms/<AppName>JmsPagingStore`
4. In the **Directory** field, enter this directory path.
5. Click **Finish**.

   The Summary of Persistent Stores page appears.
6. Click the name of the file store that you just created.
7. Click **Advanced**.
8. In the **Synchronous Write Policy** drop-down list, select `Cache-Flush`.
9. Click **Save**.

6.1.4.5 How to Create a JMS Server

1. Create a JMS Server that will manage connections and message requests from the JMS Channel.
    a. Go to **Services > Messaging > JMS Servers**.
    b. Click New.
2. Configure the connection factory using the following settings:

Name	`<AppName>JmsServer`
Persistent Store	`<AppName>JmsStore`

3. Click **Next**.
4. Target the application server.
5. Click **Finish**.
6. In the JMS Servers table, click the JMS server that you just created.
7. Under the **Configuration > General** tab, specify the Paging Directory. For example:

   `/modeln/jms/<AppName>JmsPagingStore`
8. Enter message buffer size based on your server's system configuration.
9. Click **Save**.

10. Create the JMS destination queues on the new JMS Server that will be used by the JMS channels.
    a. Click **Services > Messaging > JMS Modules**.
    b. Click the `<AppName>`SystemModule.
    c. Click **New**.
    d. Select **Queue**.
    e. Click **Next**.
11. Configure the queues using the following settings:

Name	`<AppName>`Queue
JNDI Name	`<AppName>`Queue

12. Click **Next**.
13. Click **Create a New Subdeployment**.
14. Configure the subdeployment with the following setting:
    - **Subdeployment Name:** <AppName>Subdeployment
15. Click **OK**.
16. Back on the Create a New JMS System Module Response page, select the subdeployment you just created in the **Subdeployments** drop-down list.
17. Target the JMS Server <AppName>JmsServer created in How to Create a JMS Server on page 95.
18. Click **Finish**.

6.2 Installation of External Properties

The external application property files refer to the host-specific configuration files that the Model N web application uses during application initialization. These property files are distributed outside of the EAR file to allow the application to be easily configured for a specific host without updates to the web application (WAR) file. The settings within these configuration files override the standard property settings that are included in the WAR file.

For details about the external property files see Application Server Support Files. This section focuses on the properties that apply only to WebLogic deployments.

To install the external property files:

1. Extract the application support files described in Application Server Support Files into an appropriate location on the application server.

2. To configure the Model N web application, update the `./props/local_weblogic.properties` file with host-specific settings:

Code 6-1: WebLogic Specific Runtime Property File

```
#=========================================================
# Model N GA WebLogic Specific Runtime Property File
# =========================================================
# Load AppServer independent properties from a  property file
com.modeln.packaging.vertical=pharma (1)
com.modeln.packaging.configuration=FullSuitePackage
include_010=local_common (2)
com.modeln.Log.FILE_NAME=$(logDir)/application.log
com.modeln._default.appserver=Weblogic
com.modeln.jdbcDriverClass=weblogic.jdbc.jts.Driver
com.modeln._default.jdbcUrl=jdbc:weblogic:jts:<AppName>Data
Source(3)
com.modeln._default.jtsDataSourceName=<AppName>DataSource(3)
com.modeln._default.jndiUrl=t3://localhost:<WebLogicPort>/
com.modeln.jtsTransactionName=javax.transaction.UserTransaction
com.modeln.jtsUserTransactionName=javax.transaction.User
Transaction
com.modeln.jtaJndiUrl=${com.modeln._default.jndiUrl}
com.modeln.jndiCtxFactory=com.modeln.infr.env.config.
CMnPropCtxFactory
#
# XA Datasource settings
#
com.modeln.XADataSource.jtsDataSourceName=
<AppName>XADataSource (4)
com.modeln.XADataSource.jndiUrl=${com.modeln._default.jndiUrl}
com.modeln.XADataSource.jdbcUrl=
jdbc:weblogic:jts:<AppName>XAPool (5)

#########################################################
###Please place all custom additions below this point ###
#########################################################
```

Code Section Notes:

(1) The packaging configuration statements define which application configuration to use. For more information on packaging, see Packaging.

(2) The `include` statements are used to include settings from other property files. Any property settings specified *before* the include statements are overridden by the included settings. Similarly, any property settings specified *after* the include statements override the included settings. Any property files must be located somewhere in the web application classpath so the classloader can be used to locate the files.

(3) The transactional data source (`TxDataSource`) properties determine what JNDI resource the Model N web application attempts to use when performing database queries. These property settings must correspond to the WebLogic data source created in How to Configure the Environment.

(4) The XA data source properties determine which JNDI resource the Model N web application will attempt to use during data flow runs when initiating a global transaction involving the JMS Channel and the Model N database. This property setting must correspond to the WebLogic XA data source created in Configuring the JMS Channel. This property must be set if the JMS server is also configured on WebLogic.

(5) The XA connection pool provided connections that can participate in global transactions. This property must correspond to the XA Connection Pool created in Configuring the JMS Channel. This property must be set if the JMS server is also configured on WebLogic.

6.3 Installation of the EAR File

This section describes how to actually deploy the Model N system. The Model N enterprise application (EAR) file contains the Model N system. The EAR file, `<AppName>.ear`, is included within the deployment package:

6.3.1 How to Install the EAR File

To install the EAR file, using the WebLogic Server Administration Console:

1. Click the Lock & Edit button.
2. Click **Deployments**.
3. Click Install.
4. Browse to the `pharma.ear` or the `meddev.ear` file.
5. Click Next.
6. Click **Install this deployment as an application**.
7. Click Next.
8. Select the managed server on which you would like to deploy the application.
9. Click Next.
10. Give the deployment a name.
11. Accept the default settings.
12. Click Next.
13. Click Finish.
14. Click Activate Changes.

6.3.2 How to Start the WebLogic Server

To start the WebLogic Server:

1. Under the domain you have created, click **Environment > Servers**.

2. In the Summary of Servers page, click the managed server that you created in How to Install the EAR File on page 98.
3. In the Settings For page, click the **Control** tab.
4. At the bottom of the page, select the check box next to the managed server you created.
5. Click **Start**.
6. Click **Yes** when asked if you would like to start this server.
7. Once the managed server is running, in the Domain Structure right pane, click **Deployments**.
8. Select the application you have deployed.
9. Click **Start > Start Servicing all requests**.

6.4 How to Re-deploy an .ear File

This section describes how to remove and re-install an application under a previously configured instance. The WebLogic managed server must be running during the entire redeployment. However, if a new database import is being deployed in conjunction with the application, you must shut down the managed server to allow existing database connections to be released. Once the database import is complete, re-start the managed server only *after* the Model N enterprise application has been removed. See also How to Re-deploy an Application.

To re-deploy an application:

1. Delete the Model N system as follows:
    a. In the WebLogic administration console, select **Deployments >Applications >** **`<AppName>`**.
    b. Right click the **`<AppName>`** node in the tree and select **Delete** **`<AppName>`**.
    c. Choose **Yes** when prompted.

    The deletion could take a few minutes to complete.
2. Import the new EAR file as described in Installation of the EAR File.
3. Deploy the docroot JAR as described in Install the static content into the HTTP server directory.

6.5 HTTP Server Configuration

This section explains how to configure the HTTP server for iPlanet/Sun Java System Web Server and for Microsoft IIS.

6.5.1 How to Configure iPlanet/Sun Java System Web Server

If the application server and the web server reside on the same host, the path to proxy library (`libproxy.so` on Windows and `libproxy_61.so` on Solaris) can be referenced directly from the web server configuration file. If the application server resides on a different host, you must copy the proxy library from the application server to the web server and reference it from its new path on the web server.

Note: The following configuration examples contain lines that are longer than can be displayed on the width of a page. Lines which have been broken for page formatting reasons will be indicated with the pilcrow (¶) symbol. Do not separate these lines in the actual file because that may prevent the configuration from working correctly.

---

1. Install the static content into the HTTP server directory.

   The static content required by the application is contained in the **`<AppName>`**`-docroot.jar` file provided. Extract the contents of this archive into the HTTP server docroot. The docroot directory is typically found in **`<httpdHome>`**`/htdocs/en_US.`

2. Add the following lines to the end of the `magnus.conf` file using `libproxy.so` on Windows and `libproxy_61.so` on Solaris:

Code 6-2: magnus.conf

```
Init fn="load-modules" ¶
funcs="wl-proxy,wl-init" ¶
shlib="<BEA INSTALLATION HOME>/server/plugin/solaris/sparc/
   libproxy.so"

Init fn="wl-init"
Init fn="define-perf-bucket" name="wl-bucket"¶
description="WebLogic bucket"
```

3. Create a proxy path definition to map the **`<AppName>`** application path to the root of the server. This is necessary because the application generates dynamic HTML pages that reference resources relative to the root of the application server.

4. Add the following object entry to `obj.conf`:

Code 6-3: obj.conf

```
<Object name="weblogic"

ppath="<webserver ContextRoot>/*"

Service fn="wl-proxy" ¶
    WebLogicHost="localhost" ¶
    WebLogicPort="<WebLogicPort>" ¶
bucket="wl-bucket" ¶
    HungServerRecoverSecs=7200 ¶
    ErrorPage="/SiteUnavailable.html"

</Object>
```

The web server docroot path refers to the absolute path on the web server where the docroot has been deployed.

6.5.2 How to Configure Microsoft IIS 5/6

To configure Microsoft IIS to proxy the Model N application, ensure that the following files from the WebLogic installation are accessible by the IIS instance:

These files are located within the WebLogic server binary directory, `<WL_HOME>/server/plugin/win/x64`:

- `iisproxy.dll`
- `iisforward.dll`

If IIS is installed on the same host as WebLogic, the files do not need to be copied anywhere. If a separate IIS host is used, copy these files to that host and place them in a directory together.

Install the static content into the HTTP server directory. The static content required by the application is contained in the **<package>**`-docroot.jar` file provided. Extract the contents of this archive into the HTTP server docroot. The docroot directory is typically found in **`<httpdHome>`**`/htdocs/en_US`.

6.5.2.1 How to Configure the WebLogic Forwarding Module

To register the `iisforward.dll` as an ISAPI filter in the IIS WebServer configuration, using the Internet Information Service Manager:

1. Go to the Internet Information Service Manager.
2. In the tree view on the left, select **Default Web Site**, under Local computer > Web Sites.
3. Right-click and select **Properties**.
4. On the ISAPI Filters tab in the properties dialog box, click **Add**.
5. For the Filter name parameter, enter: `IIS Forward`.
6. For the Executable parameter, browse to the directory above and select `iisforward.dll`.
7. Click **OK**.
8. Click **Apply**.
9. Click **OK**.

6.5.2.2 How to Configure the Proxy as an Application Extension

To register and configure the `iisproxy.dll` as an application extension (MIME type) using the Internet Information Service Manager:

1. Go to the Internet Information Service Manager.
2. In the tree view on the left, select **Default Web Site**, under Local computer > Web Sites.
3. Right-click and select **Properties**.
4. On the Home Directory tab in the properties dialog box, click **Configuration**.
5. Under Application extensions, click **Add**.
6. For the Executable parameter, browse to `iisproxy.dll`.

7. For the Extension parameter, enter `.wlforward`.
8. For version 6+ of IIS, clear **Check that file exists**.
9. Click **OK**.
10. Click **OK**.

6.5.2.3 How to Allow the New ISAPI Extension to be Executed

To set the new ISAPI extension permissions to `Allow` so that it can be executed by IIS, using the Internet Information Service Manager:

1. Go to the Internet Information Service Manager.
2. In the tree view on the left, select **Local computer**.
3. Select **Web Service Extensions**.
4. Select **All Unknown ISAPI Extensions** from the list at right.
5. Click **Allow**.
6. Click **Yes**.

6.5.2.4 How to Create the iisproxy.ini File

To create an `iisproxy.ini` file in the `<WL_HOME>/server/plugin/win/32` folder (the folder where the `.dll` files are located), in order to configure the behavior of the ISAPI extension:

1. Create a text file called `iisproxy.ini` in the same directory as `iisproxy.dll` and `iisforward.dll`.
2. Edit the text file to include the following entries:
    - `WlForwardPath=/<ContextRoot>`
    - `WebLogicHost=localhost`
    - `WebLogicPort=<yourPort>`
    - `FileCaching=OFF`
    - `WLIOTimeoutSecs=900`

    The `WebLogicHost` and `WebLogicPort` settings represent the WebLogic host and port to be used.

6.5.2.5 How to Restart the IIS Windows Services

The IIS windows services must be re-started for the above edits to take effect. To re-start the IIS Windows Services, using the IIS Administration Console:

1. Close the IIS Manager if it is open.
2. In the Services control panel, stop IIS Admin Service.

    All dependent services will be stopped.)

3. Start IIS Admin Service.
4. Start World Wide Web Publishing Service.

You should now be able to access the application by the port on which IIS is configured to listen (default port 80), using the following URL:

`http://`**`<hostname>:<port>`**`/`**`<ContextRoot>`**`/BusinessNetwork`

6.5.3 How to Configure IIS 7

To configure Microsoft IIS to proxy the Model N application, ensure that the following files from the WebLogic installation are accessible by the IIS instance:

These files are located within the WebLogic server binary directory, `<WL_HOME>/server/plugin/win/x64`:

- `iisproxy.dll`
- `iisforward.dll`

If IIS is installed on the same host as WebLogic, the files do not need to be copied anywhere. If a separate IIS host is used, copy these files to that host and place them in a directory together.

Install the static content into the HTTP server directory. The static content required by the application is contained in the **<package>**`-docroot.jar` file provided. Extract the contents of this archive into the HTTP server docroot. The docroot directory is typically found in **`<httpdHome>`**`/htdocs/en_US.`

To register the `iisforward.dll` as an ISAPI filter in the IIS WebServer configuration, using the Internet Information Service Manager:

1. Go to the Internet Information Service Manager.
2. In the tree view on the left, select **Default Web Site**, under Local computer > Sites.
3. Right-click and select **Manage Web Site > Advanced Settings**.
4. Set the Physical Path property to `<WL_HOME>/server/plugin/win/x64`.
5. Select the Default We Site and double-click the ISAPI Filters shortcut.
6. Right click and select **Add**.
7. For the Filter name parameter, enter: `wlforward`.
8. For the Executable parameter, browse to the directory above and select `iisforward.dll`.
9. Click **OK**.
10. Go back to the **Default Web Site** home and double-click on **Handler Mappings**.
11. Right click and select **Add Script Map**.
12. Enter the following values:

Request Path	*.wlforward
Executable	`<WL_HOME>/server/plugin/win/x64`/iisproxy.dll
Name	IIS Proxy

13. Click **OK**, then accept the confirmation dialog box.
14. Restart IIS.

6.5.4 How to Configure Apache

To configure the Apache server to work with WebLogic Server:

1. Install the static content into the HTTP server directory.

   The static content required by the application is contained in the **`<AppName>`**`-docroot.jar` file provided. Extract the contents of this archive into the HTTP server docroot.

   If you do not know where the docroot is located, look in the following file:

   ```
   <APACHE_HOME>/conf/httpd.conf
   ```

   for the following entry:

   ```
   DocumentRoot "</location/of/static/content>"
   ```

   Note: If you have installed the Red Hat RPM, the docroot directory is found in `/var/www/html`.

2. Use one of the following commands to verify that the `mod_so.c` was enabled when the Apache software was compiled:

   ```
   <APACHE_HOME>/bin/apache -l
   ```

   or

   ```
   <APACHE_HOME>/bin/apachectl -l
   ```

   You should see `mod_so.c` in the compiled modules list.

   Note: If you have installed the Red Hat RPM, the binary is located in `/usr/sbin/apachectl`.

3. Locate the object file (`mod_wl_22.so`) under `<BEA_HOME>wlserver_10.3/server/plugin/linux/x86_64`.
4. Copy the WebLogic object file into the Apache module directory as follows:

   ```
   cp <BEA_HOME>/wlserver_10.3/server/plugin/linux/x86_64/
   mod_wl_22.so <APACHE_HOME>/modules/
   ```

   Note: The object file version varies depending on the Apache version and the encryption strength chosen. For example, if your environment uses Apache 2.2 and you chose the regular strength encryption, the object file version is: `mod_wl_22.so`. For more information, see WebLogic documentation at http://edocs.bea.com/wls/docs100/plugins/apache.html.

   Note: If you have installed the Red Hat RPM, the Apache module directory is `/etc/httpd/modules`.

5. Add the following configuration to Apache's main configuration file, `<APACHE_HOME>/conf/httpd.conf`, which is `/etc/httpd/conf/httpd.conf`, if you have installed the Red Hat RPM.
   a. To load the module upon starting the Apache server, add the following line in the `LoadModule` section:

   ```
   LoadModule weblogic_module modules/mod_wl_22.so
   ```

   b. Configure the plug-in by adding the following statement:

```
<IfModule mod_weblogic.c>
        WebLogicHost localhost
        WebLogicPort <WebLogic Port>
        KeepAliveEnabled On
</IfModule>
```

   c. To map proxy requests, add the following configuration to the `httpd.conf` file:

```
<Location /<ContextRoot>>
        SetHandler weblogic-handler
</Location>
```

6. Restart Apache.

Note: You can verify the syntax of the `httpd.conf` file by running `<APACHE_HOME>/bin/apachectl -t` at anytime.

7

Deploying with IBM WebSphere

This chapter describes how to install the Model N enterprise application under WebSphere using a base configuration running on a single server.

Note: Make sure that you have set up Oracle already, so that the Model N application database is available prior to starting the WebSphere server.

You can deploy the application using a base WebSphere installation running on a single server or using a network deployment across multiple servers. Model N recommends network deployment for deployments that require load balancing across multiple application servers. For instructions on how to configure network deployment, see How to Configure Network Deployment.

During the final steps of the installation, review the list of newly-selected ports to ensure that none conflict with the existing WebSphere installation or with other network processes.

If you have overridden the default port for the administration server, make note of the port, as it will be used later to access the administration web interface.

These instructions were written to install WebSphere on Linux. The following two changes must be noted when using these instructions to install WebSphere on Windows:

- change absolute paths so that they apply to Windows
- change `.sh` files to `.bat` files.

Note: Some examples in this chapter may contain lines which are longer than can be displayed on the width of a page. Lines which have been broken for page formatting reasons will be indicated with the pilcrow (¶) symbol. Do not separate these lines during actual configuration because doing so may cause the application to fail to initialize correctly.

7.1 Configuring the Environment

This section describes the modifications required to ensure that WebSphere has the correct JVM and shell environment settings including information on configuring the web application using a local property file. To customize the WebSphere application server and configure the web application:

1. Install the support files from the Model N distribution by extracting the external Tar file **`<AppName>`**`-external.zip` file into a directory of your choosing (i.e. `<ExtDir>`).

   The Model N application loads files from these directories using the WebSphere class loader. These directories are added to the classpath by modifying the environment variables in step two below.

2. Configure the WebSphere environment variables.

   A sample property file will be included as part of the `external.tar` file, which is part of the deployment package. Assuming the `external.tar` file has been extracted into a permanent location on the application server, the property file can be found at `./props/local_websphere.properties`.

   The following is an example of a property file for a standard WebSphere deployment:

Code 7-1: Application Property File

```
#=========================================================
# Model N 5.1 GA WebSphere Specific Runtime Property File
#=========================================================
# Load AppServer independent properties from a shared property
file
com.modeln.packaging.vertical=pharma (1)
com.modeln.packaging.configuration=FullSuitePackage
include_010=local_common (2)
com.modeln.Log.FILE_NAME=${logDir}/application.log
com.modeln._default.appserver=WebSphere
com.modeln._default.jndiUrl=iiop://localhost:2810 (3)
com.modeln.jtaJndiUrl=${com.modeln._default.jndiUrl}
com.modeln.jtsTransactionName=jta/usertransaction
```

Code 7-1: Application Property File (Continued)

```
com.modeln.jtsUserTransactionName=jta/usertransaction
com.modeln._default.jtsDataSourceName=java:comp/env/jdbc/
<DBUser>OraDatasource (4)
com.modeln.ejbJndiCtxFactory=\
com.ibm.websphere.naming.WsnInitialContextFactory
##########################################################
## Please place all custom additions below this point ###
##########################################################
```

Code Section Notes:

(1) The packaging configuration statements define what application configuration to use. For more information, see Packaging.

(2) The `include` statements are used to include settings from other property files. Any property settings specified before the `include` statements are overridden by the included settings. Similarly, any property settings specified after the `include` statements override the included settings. Any property files must be located somewhere in the web application classpath so the class loader can be used to locate the files.

(3) The JNDI port specified here must correspond to the port specified during WebSphere installation and configuration. Update the property file with the appropriate JNDI port number. This port will be generated automatically later in the configuration process.

(4) The data source properties determine what JNDI resource the Model N web application uses when performing database queries. The property must correspond to the resource reference entry in the `web.xml` file (as `java:comp/env/<resource-ref@name>`) instead of the datasource's JNDI name. Therefore, do not edit this property without also editing the `web.xml` file.

7.2 Administering WebSphere

The following instructions explain how to create the WebSphere environment under which the Model N enterprise application is installed. The instructions are written for a trained WebSphere administrator using the WebSphere administration client. Although you can use the WebSphere command-line utilities to configure the application, Model N recommends that these utilities be used only by experienced WebSphere administrators.

Before you perform any administration tasks, the WebSphere Administration Server must be running. To start the server, use the appropriate WebSphere shell script such as:

```
/usr/WebSphere/AppServer/bin/startServer.sh server1
```

Use the WebSphere administration web interface to perform the configuration steps described below. The administration console can be accessed using the host and port information specified during the WebSphere installation:

`http://hostname:adminport/admin/`

The administration port defaults to 9043. However, the port number may be different if this installation co-exists with additional WebSphere instances or the default port was modified during installation.

Note: The URL of Model N application needs the full domain name. Therefore, if you are using Virtual Host, make sure that the web server's virtual host and the application server's virtual host setting both have the full domain name.

To configure the WebSphere application server:

Select or configure a virtual host to serve the Model N static content.

The Model N application must serve static content from port 80. To do this using WebSphere, you can do one of the following:

- Use the default virtual host.
- Establish a virtual host that is listening on port 80.
- Set up an external HTTP server.

If you have example WebSphere applications that are listening on port 80 and wish to create an additional virtual host that listens on port 80, change the default virtual host port to something other than port 80. The virtual host ports and IP addresses correspond to the HTTP configuration settings discussed in How to Configure the IBM HTTP Server (IHS).

Create a new application server instance for the Model N application.

Using the WebSphere administration scripting program (wsadmin), create a new application server instance for the current WebSphere node. If multiple application servers are required, Model N recommends that the administrator who is configuring the server record the server and port information in order to use these values during subsequent configuration steps. You can create a new application server using the command-line interface as in the instructions below, which create the application server instance. The first three lines listed below are part of a single command line. The next two lines, which consist of wsadmin prompts, are interactive prompts where you enter commands.

```
<WEBSPHERE HOME>./wsadmin.sh -conntype SOAP -user
websphere -password websphere/bin/wsadmin.sh

wsadmin>  $AdminTask createApplicationServer <node>

{ -name <AppName> -templateName default }

wsadmin>  $AdminConfig save

wsadmin>  exit
```

Note: You must use brackets { } around the `-name` and `-templateName` arguments as shown in the preceding commands and you must set your WebSphere username and password as shown in the preceding example.

Once the server has been created, log off and then log back into the administration web console to configure the server instance:

1. Click **Servers > Server Types > WebSphere Application Servers**.
2. Click the **`<AppName>`** link.

   If you are installing multiple application instances, it may be desirable to append a unique number to the application name in order to distinguish the instances.
3. Under the Container Settings section, click **Container Services > Transaction Service**.
4. Set the **Total transaction lifetime timeout** value to 0.
5. Set the **Maximum transaction timeout** value to 0.
6. Click **OK**.
7. Under the Server Infrastructure section, expand Java and Process Management.
8. Click **Process Definition**.
9. Under Additional Properties, click **Java Virtual Machine**.
10. For **Classpath**, enter `<ExtDir>/props:<ExtDir>/jdbc/ojdbc6.jar`.
11. For **Boot Classpath**, enter `/xml.jar:/xerces.jar`.
12. Select **Verbose Garbage Collection**.
13. For a 64-bit system, set **Initial Heap Size** to `1536`.
14. For a 64-bit system, set **Maximum Heap Size** to `4096`.
15. For Generic JVM arguments, on a single-processor machine enter:

    ```
    -Xgcpolicy:optthruput -Xk35000
    -Dcom.modeln.propFile=local_websphere
    -Djava.net.preferIPv4Stack=true -Xmxcl50000
    -Dfile.encoding=UTF-8
    ```

    For Generic JVM arguments, on a multi-processor machine (where the `4` in `-Xgcthreads4` represents the number of processors and the `2` in `-Xconcurrentbackground2` represents half the number of processors), enter:

    ```
    -Xminf0.05 -Xmine50M -Xmaxf1 -Xgcthreads4
    -Xconcurrentbackground2 -Xgcpolicy:gencon -Xk35000
    -Dcom.modeln.propFile=local_websphere
    -Djava.net.preferIPv4Stack=true -Xmxcl50000
    -Dfile.encoding=UTF-8
    ```

    Omit the locale and file extension when specifying the property file. Java searches for the property files that are appropriate for the locale settings of the current installation. For example, a property file with the name, `test.properties`, must be specified as `-Dcom.modeln.propFile=test` when setting the JVM arguments.
16. Click **OK**.
17. Under Additional Properties, click **Environment Entries**.
18. Click **New**.

19. Under General Properties, create the following name-value pairs. Click **OK** after you enter each name-value pair:

Table 7-1: Name-Value Pairs

Name	Value
`LANG`	`en_US.UTF-8`
`IBM_HEAPDUMP`	`true`
`IBM_HEAPDUMPDIR`	`/home/<AppName>_dump`
`IBM_JAVACOREDIR`	`/home/<AppName>_dump`
`IBM_JAVA_MMAP_JAVA_HEAP`	`true`
`LDR_CNTRL`	`MAXDATA=0x10000000`

Note: When specifying an output directory, ensure that the amount of available free space at that location is greater than the size of the JVM heap. If a heap dump is performed, the resulting heap file will contain all of the information that was resident in memory at the time.

20. In the Messages box, click Save.

    In the left navigation pane:

21. Click **Servers > Server Types > WebSphere Application Servers**.
22. Click `<AppName>`.

    In the right navigation pane:

23. Under the Communications section, expand Ports.
24. Note the port used for the `BOOTSTRAP_ADDRESS`.
25. Go to the <AppName> server configuration page by clicking **Servers > Server Types > WebSphere Application Servers >** *AppName.*
26. Click **Web Server Plug-in properties** under Additional Properties.
27. Update the Read/Write time out value from 60 to 300.
28. Save the change and restart the server.
29. Generate the web-server plug-in.
30. Validate that the ServerIOTimeout parameter is configured to 300 in the `plugin-cfg.xml` file.

Create Database Authentication Credentials for a JDBC Provider

This section includes information on how to create the database authentication credentials needed for a JDBC provider and information on how to create a JDBC provider.

Create the database authentication credentials:

1. Click **Security > Secure administration, applications, and infrastructure**.
2. Under Authentication, click **Java Authentication and Authorization Service > J2C authentication data**.
3. Click **New**.
4. Fill in the following configuration properties:
    - **Alias:** `<AppName>Auth`
    - **User ID:** <DBUser>
    - **Password:** <DBPass>
    - **Description:** Model N database
5. Click **OK**.

Create a New JDBC Provider

1. Select **Resources > JDBC > JDBC Providers**.
2. From the drop-down list, select the server scope that corresponds to the `<AppName>` server.
3. Click **New**.
4. Fill in the following JDBC provider fields:
    - **Database type:** Oracle
    - **Provider type:** Oracle JDBC Driver
    - **Implementation type:** Connection pool data source
    - **Name:** Oracle Non JTA Non XA
5. Click **Next**.
6. Specify the JDBC driver classpath: `<ExtDir>/jdbc/ojdbc6.jar`
7. Click **Next**.
8. Click **Finish**.
9. Under Resources, click **JDBC > Data Sources**.
10. From the drop-down list, select the server scope that corresponds to the `<AppName>` server.
11. Click **New**.
12. Fill in the following data source information fields:
    - **Data source name:** `<AppName>OraDataSource`
    - **JNDI name:** `jdbc/<AppName>OraDataSource`
13. Select the `<AppName>Auth J2C` authentication alias from the drop-down list.
14. Click **Next**.
15. Select **Select an existing JDBC provider**.
16. Select the "Oracle Non JTA Non XA" JDBC provider defined earlier.

17. Click **Next**.
18. Specify the properties for the data source.

    URL: `jdbc:oracle:thin:@<DBHost>:<TNSPort>:<DBInstance>`

    Data store helper class name: Oracle 9i and prior data store helper.

    Select "Use this data source in container managed persistence (CMP)" from the drop-down list.
19. Click **Next**.
20. Review the settings and click **Finish**.
21. Click `<AppName>OraDataSource` from the table of available data sources.
22. Under Additional Properties, click **Connection Pool Properties**.
23. Modify the following configuration properties:
    - Maximum connections: 40
    - Minimum connections: 20
24. Click **OK**.
25. In the messages box, click **Save** to commit changes.
26. Select the datasource.
27. Select WebSphere Application Server data source properties.
28. Set the **Statement cache size** to 0 (default 10) and save your changes.

Test the Connection

1. Select **Resources > JDBC > Data sources**.
2. From the drop-down list, select the server scope that corresponds to the `<AppName>` server.
3. Select the check box that corresponds to the `<AppName>OraDataSource.`
4. Click **Test Connection**.

Create a Shared Library

1. Select **Environment > Shared Libraries**.
2. Select the server created for the application as the scope for the shared library.
3. Click **New**.

   Specify Name, `<AppName>SharedLibrary`, for the shared library.
4. Enter the paths to the following JAR files in the classpath:

```
<ExtDir>/lib/jaxb-impl.jar

<ExtDir>/lib/jaxb-api.jar

<ExtDir>/lib/jaxws-api.jar

<ExtDir>/lib/sjsxp.jar
```

Note: Entries must not contain path separator characters (such as ; or :). Press ENTER to separate class path entries.

5. Click **OK**.
6. Click **Save** to save the configuration.

Install the new enterprise application

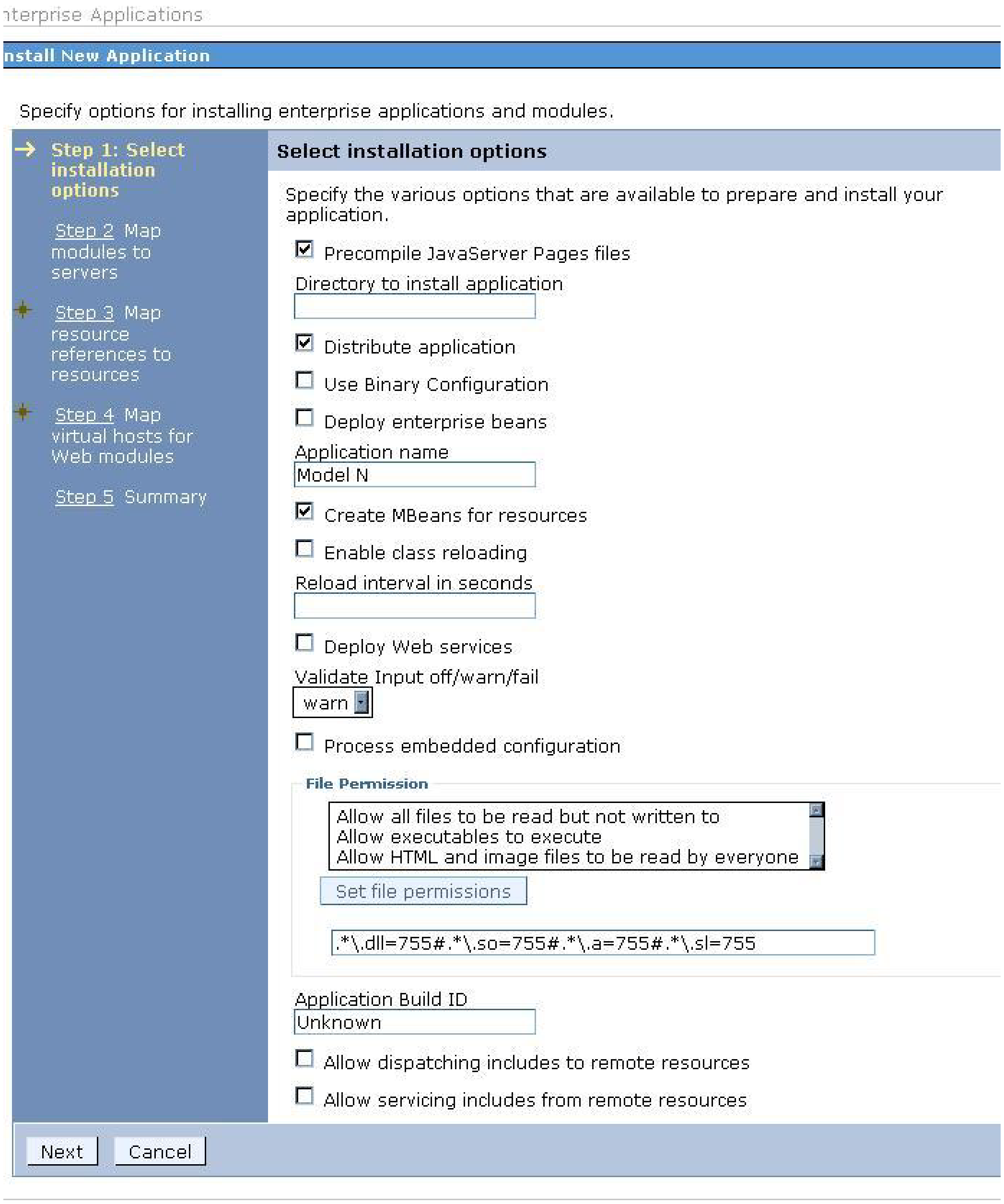
To install a new enterprise application, navigate to **Applications > Install New Application** in the administration console. This initiates the application installation wizard. A series of configuration screens are presented to gather information about various aspects of the application. The following text lists each of the screens and any information that you must specify. You can use the **Next** button to skip any screens that require no information.

1. In the Preparing for the application installation screen, under Specify the EAR, WAR, JAR, or SAR module to upload and install, select the radio button that contains the location of the Model N EAR file - either the local file system or the remote file system.
2. Enter the location of the EAR file to be deployed. You can do this by browsing to it and selecting it.
3. Click **Next**.
4. Select **Detailed - Show all installation options and parameters**.

   The Install New Application wizard opens.
5. For *Step 1: Select installation options*:
   a. Select **Pre-compile JavaServer Pages files**.
   b. Leave the **Directory to Install Application** field empty.
   c. Make sure that **Distribute Application** is selected.
   d. Make sure that **Use Binary Configuration** is not selected.
   e. Make sure that **Deploy enterprise beans** is not selected.
   f. Enter <AppName>App in the **Application Name** field if needed.
   g. Make sure that **Create MBeans for Resources** is selected.
   h. Make sure that **Enable Class Reloading** is not selected.
   i. Leave the **Reload Interval in Seconds** field empty.
   j. Make sure that **Deploy Web services** is not selected.
   k. Leave the other fields as their default.
   l. Click **Next**.

The following shows the correct settings for this screen:

Figure 7-1: Select Installation Options



nterprise Applications

nstall New Application

Specify options for installing enterprise applications and modules.

Step 1: Select installation options

Step 2 Map modules to servers

Step 3 Map resource references to resources

Step 4 Map virtual hosts for Web modules

Step 5 Summary

Select installation options

Specify the various options that are available to prepare and install your application.

☑ Precompile JavaServer Pages files

Directory to install application

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: Model N

☑ Create MBeans for resources

☐ Enable class reloading

Reload interval in seconds

☐ Deploy Web services

Validate Input off/warn/fail: warn

☐ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

Set file permissions

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

☐ Allow dispatching includes to remote resources

☐ Allow servicing includes from remote resources

6. For *Step 2: Map modules to servers:*
    a. In the **Clusters and Servers:** field, select the appropriate web server and application server to which you would like to deploy the Model N application. To select them both at the same time, press the CTRL key while selecting.
    b. Select the appropriate WAR file.
    c. Verify that the correct server name displays next to the WAR file.

    d. Click **Apply**.
    e. Click **Next**.

7. For *Step 3: Provide options to compile JSPs:*
    - Enter `15` for the **JDK Source Level**, then click **Next**.
8. For *Step 4:* Provide JSP reloading options for Web modules, then click **Next**.
9. For *Step 5:* Map shared libraries, then click **Next**.
10. For *Step 6:* Map shared library relationships, then click **Next**.
11. For *Step 7:* Map resource references to resources:
    a. Under Specify authentication method, select **Use default method (many-to-one mapping)**.
    b. From the drop-down list, select the appropriate node with the appropriate authentication type.
    c. Select the Model N WAR file.
    d. Click **Apply**. The Login configuration information is populated.
    e. In the Target Resource JNDI Name field, click Browse. The Enterprise Applications screen opens.
    f. Select the appropriate JNDI, which was created during the server creation.
    g. Click **Apply**.
    h. Click **Next**.
12. For *Step 8: Map virtual hosts for Web modules:*
    a. Select the Model N WAR file.
    b. Select the virtual host from the drop-down.
    c. Click **Next**.
13. For *Step 9:* Map context roots for Web modules, then click **Next**.
14. Once the message Application <AppName>App installed successfully. is displayed, click the **Save directly to the master configuration**. link to commit the application deployment. The Enterprise Applications screen is displayed.

Associate the Shared Library with the Application

To associate the shared library with the application:

1. Navigate to **Enterprise Applications > Installed Applications > <AppName>App**.
2. Click **Shared Library References** under References.
3. Select the check box next to <AppName>App application and click **Reference shared libraries**.
4. Select `<AppName>SharedLibrary` from the Available list and move it to the Selected list.
5. Click **OK**.

6. Make sure that <AppName>SharedLibrary is listed under Shared Libraries for <AppName>App.
7. Click **OK**.
8. Click **Save** to save the configuration.

Change the application (EAR) class loader policy to PARENT_LAST

To change the application (EAR) class loader policy to `PARENT_LAST`:

1. Navigate to **Enterprise Applications > Installed Applications > <AppName>App**.
2. Select **Class Loading and Update Detection** under the Detail Properties section.
3. Under Class Loader Order, select **Classes Loaded with Application Class Loader** first.
4. Set the polling interval to `0`.

   This setting is not required and has no effect since you are not selecting the Reload Classes option. However, a value is needed for the polling interval because WebSphere does not allow this field to be empty when saving this page.
5. Click **OK**.
6. Click **Save** to save the configuration.
7. Re-start the server after saving the changes.
8. To start the application server from here, select it and click **Start**.
9. To start the application from the command line, enter `startServer.sh <AppName>.`

For information on how to configure the IBM HTTP Server, see How to Configure the IBM HTTP Server (IHS).

7.3 How to Re-deploy an Application

This section describes how to remove and re-install an application under a previously configured instance. See also How to Re-deploy an Application.

To re-deploy an application:

1. Delete the enterprise application:
   a. Select **Applications** > **Enterprise Applications**.
   b. Select the application to uninstall.
   c. Click **Uninstall**.
   d. Click the **Save** link in the Messages box.
   e. Click **Save**.
2. Import the new EAR file. See Install the new enterprise application
3. Deploy the docroot JAR. See Install the static content into the IHS directory.

7.4 How to Configure Network Deployment

This section assumes that the you have installed WebSphere network deployment.

The following modules should be considered during the installation process:

- Deployment Manager (required)
- Web Services (optional)
- Embedded Messaging (optional)

7.4.1 How to Add a Base Node to Network Deployment Cell

To add a base node to Network Deployment Cell:

1. Shut down the base administration console and the base application server.
2. Install the WebSphere Network Deployment and create a profile that contains the deployment manager role (a cell profile or a deployment manager profile).
3. Start the network deployment manager using the appropriate WebSphere start-up script. For example:

   ```
   <WAS_HOME>/bin/startManager.sh
   ```

4. You can access the administration console using the host and port information specified during the WebSphere installation process:

   ```
   http://hostname:adminport/admin/
   ```

5. Look up the port number that will be used to add a base node:
   a. Select System Administration
   b. Select Deployment Manager
   c. Select Endpoints
   d. For `SOAP_CONNECTOR_ADDRESS`, set:

      ```
      Port = 8879
      ```

6. On the machine that has the base application server, run the following script to add the base node to the deployment manager with the preceding port number:

   ```
   <WAS_HOME>/bin/addNode.sh <deployMgrHost> 8879
   -includeapps
   ```

The preceding process uploads all of the configuration information from the base node to the Network Deployment Cell and adds the node to the cell. You can perform all configuration of the node from this point on from the Deployment Manager Administration console. A node agent process will be started within the base installation. The agent process will perform the communication between the node and the deployment manager. You can start and stop the node agent using the appropriate start and stop command:

```
/bin/startNode.sh -profileName <profile>
```

```
/bin/stopNode.sh -profileName <profile>
```

7.4.2 How to Create a Cluster of Application Server Instances

Use the Deployment Manager administration console to perform the following configuration steps:

1. Select **Servers**.
2. Select **Clusters**.
3. Click **New**.
4. For **Cluster Name**, enter:

   `<AppName>Cluster`

5. For **Existing Server**, select:

   ```
   Create the member by converting an existing application
   server:
   ```

6. In the drop-down list, select the base application server that was created earlier.
7. Click **Next**.
8. Give this new server a name in the **Member Name** field.
9. From the **Select Node** drop-down list, select a node to host this server.
10. In the **Weight** field, enter a weight.
11. Select the **Generate unique HTTP ports** check box.
12. Click **Add Member**.
13. Repeat steps 9 through 13 to add as many members as you need.
14. Click **Next**.
15. Click **Finish**.
16. Click **Save** in the Messages box.

7.4.3 How to Configure a Virtual Host and a JDBC Provider for Clustered Servers

In the Network Deployment administration console, navigate to the virtual host configuration (**Environment > Virtual Hosts**) and edit the default host or create a new virtual host. Set the desired listen port for the web server.

Once you have completed the configuration steps and configured the virtual host, verify that the changes have been synchronized to the base node. You can verify the synchronization status on the node administration page (**System Administration > Nodes**). A synchronized icon appears next to the base node name when synchronization is complete.

For steps on creating the JDBC provider, see Create Database Authentication Credentials for a JDBC Provider. In the step, From the drop-down list, select the server scope that corresponds to the `<AppName>` server., instead of `<AppName>` server, select the entire cell. In the step, Modify the following configuration properties:

Set the connections as follows:

- Maximum connections: 40 x Number of servers
- Minimum connections: 20 x Number of servers

set the maximum connections to 40 times the number of servers in the cluster (for two servers, set it to 80), and set the minimum connections to 20 times the number of servers in the cluster (for two servers, set it to 40).

Note: The URL of Model N application needs the full domain name. Therefore, if you are using Virtual Host, make sure that the web server's virtual host and the application server's virtual host setting both have the full domain name.

7.4.4 How to Configure Model N Application Properties for Each Clustered Member

To configure Model N application properties for each clustered member:

1. Create a new application property file in the `<externalDirVariable>/props` directory.

   The application property file will be used to start the new application that was deployed using the Network Deployment Manager. The property file name must match the name specified in the previous configuration steps:

   `new_local_websphere`

```
##########################################################
# Configuration for WebSphere
##########################################################
include_05=local_websphere

extDir=/usr/mn/release/<AppName>

com.modeln._default.jndiUrl=iiop://localhost:2811
com.modeln.jtaJndiUrl=${com.modeln._default.jndiUrl}
com.modeln.dbsync.mode=Skip
```

Note: The `jndiUrl` port number must match the Bootstrap Address port number.

   To check that these port numbers match:

   a. Go to **Servers > Server Types > WebSphere Application Servers > click the server > Communications > Ports**.

   b. Verify that the Bootstrap Address port number is the same as the jndiUrl port number.

2. Start the clustered servers.

   To start the clustered servers in the administration console, either go to **Servers > Clusters > Check the cluster > Start** or view the list of application servers

(**Servers > Server Types > WebSphere Application Servers**), select an application server and click **Start**.

Note: When installing Model N on a cluster of application servers, you must also configure commands to enforce single-threaded thread pools for some applications. For each application with a single-threaded thread pool, you must designate one application server to have the thread pool, and include commands for that application on the designated application server, excluding them on other application servers. See Thread Pools for more information.

Important: For the GP calculations to properly execute, the application configurator needs to set exclusion filters on all the servers in the cluster except for the one instance that is designated to run the GP calculations. Log in to the Model N application as a user with an Application Admin Management role. Navigate to **Administration > Nodes > Node** name for the nodes you want to exclude. Under the Node section, select Job Designation and create the exclusion filter.

7.5 How to Configure the IBM HTTP Server (IHS)

This section describes the configuration steps required to allow the IHS to proxy requests back to the application server. If a virtual host has been configured to listen on a port other than port 80, you must also configure the HTTP server to listen on that port. To configure the IHS to listen on additional ports use the `Listen` directive within the IHS configuration (`httpd.conf`) file.

To configure the IHS:

1. Install the static content into the IHS directory.

   The static content required by the application is contained in the <AppName>-`docroot.jar` file provided. Extract the contents of this archive into the IHS docroot. The docroot directory is typically found in **`<httpdHome>`**`/htdocs/en_US`.

   You will need to update this content every time you deploy a new build.

2. Start the IHS administration server.

   If the IHS resides on an unmanaged node that does not have a node agent that links to the deployment manager, start the IHS administration server to handle the communications as follows:

   ```
   * <IHS_HOME>/bin/adminctl start
   ```

3. Add the web server definition to the WebSphere deployment manager.

   A distributed server environment can have multiple web server definitions. If the web server plug-in is installed locally (on the same machine as the application server), the definition is automatically generated. To add a new definition for a remote web server:

a. On the web server machine, find its web server configuration script (`configure<web_server_name>.sh` under the `<plug-in_home>/bin` directory). If the machines are on different platforms (one machine is running on AIX or Linux and the other machine is running on Windows), find the script in the `<plug-in_home>/bin/crossPlatformScripts/` directory instead.

For example:

```
<IHS_HOME>/Plugins/bin/configurewebserver1.sh
```

```
<IHS_HOME>/Plugins/bin/crossPlatformScripts/
configurewebserver1.sh
```

b. Copy this configuration script to the application server machine under the `<WAS_HOME>/bin` directory.

c. On the application server machine, open a new command window, cd into the `<WAS_HOME>/bin` directory and run `configure<web_server_name>.sh`.

Note: Always open a new command line window to execute the script. Use the `-ihsAdminPassword` option if the IHS is on an unmanaged node. If the server is on a managed node, verify that this node is synched to the deployment manager on the administration console (**System administration > Nodes >** check the status column).

---

d. To verify that the web server definition is added, find the web server in administration console (**Servers > Web servers**).

This web server can be administered from within the network deployment administration console for tasks such as stop, start, generate, and propagate plug-in.

4. Generate the plug-in.
   a. Right-click on **Server > Web servers**.
   b. Check the installed web server instance and click **Generate Plug-in**.
   c. Check the installed web server instance and click **Propagate Plug-in**.
5. Restart the IHS by either:
   a. Checking the installed Web server instance and clicking start.

   - or -

   b. Enter `<IHS>/bin/apachectl restart` on the Web server machine.

For more information, see IBM WebSphere 6.1 documentation. Specifically, see chapter 7, Managing Web Servers, in “WebSphere Application Server V6.1: System Management and Configuration”.

8

Deploying with Red Hat JBoss Enterprise Application Platform

This chapter describes how to install the Model N enterprise application under JBoss Enterprise Application Platform (EAP).

Note: Before you begin:

- Make sure that the Oracle setup has been performed prior to this so that the Model N application database is available prior to starting the JBoss server.
- JBoss EAP must already be installed.

8.1 Installation of External Properties

The external application property files refer to the host-specific configuration files that the Model N web application uses during application initialization. These property files are distributed outside of the .ear file to allow the application to be easily configured for a specific host without updates to the web application (.war) file. The settings within these configuration files override the standard property settings that are included in the .war file.

For details about the external property files see Application Server Support Files on page 27, which focuses on the properties that apply only to JBoss deployments.

8.1.1 How to Install the External Property Files

1. Extract the application support files described in Application Server Support Files on page 27 into an appropriate location on the application server.
2. To configure the Model N web application, update the `./props/local_jboss.properties` file with host-specific settings:

Code 8-1: JBoss-Specific Runtime Property File

```
#========================================================
# Model N GA JBoss Specific Runtime Property File
# =========================================================
# Configure the Model N modules
com.modeln.packaging.vertical= pharma (1)
com.modeln.packaging.configuration= FullSuitePackage

# Load AppServer independent properties from a shared property file
include_010=local_common (2)

com.modeln._default.appserver=JBoss

com.modeln.Log.FILE_NAME=${logDir}/application.log

com.modeln._default.jndiUrl=t3://localhost:<JBossPort>/
com.modeln.jtaJndiUrl=${com.modeln._default.jndiUrl}
com.modeln.jtsTransactionName=java:comp/UserTransaction
com.modeln.jtsUserTransactionName=${com.modeln.jtsTransactionName}
com.modeln.ejbJndiCtxFactory=org.jnp.interfaces.NamingContextFactory

com.modeln._default.jtsDataSourceName=java:jdbc/<AppName>DataSource(3)

#########################################################
###Please place all custom additions below this point ###
#########################################################
```

Code Section Notes:

(1) The packaging configuration statements define which application configuration to use. For more information on packaging, see Packaging on page 221.

(2) The include statements are used to include settings from other property files. Any property settings specified before the include statements are overridden by the included settings. Similarly, any property settings specified after the include statements override the included settings. Any property files must be located somewhere in the web application classpath so the classloader can be used to locate the files.

(3) The data source name determines what JNDI resource the Model N web application attempts to use when performing database queries. This setting must correspond to the JBoss data source created in Configuring JBoss.

8.2 Configuring JBoss

This section explains how to create the JBoss server configuration under which the Model N application is deployed. For these instructions to work, you must have already installed JBoss EAP 4.3.

1. Follow the steps under section **Operations Tools > How to Configure the Environment** to setup the Ops Tool.
2. To create the JBoss server configuration for the Model N application:
   a. Go to the ant script directory:
      - On Windows: `c:\modeln\curBuild\tools\build\ant`
      - On Linux: `/opt/mn/release/tools/build/ant`

   b. Run the following command:

```
ant -f ops.xml ConfigureJBoss
```

   c. Answer the following prompts. Press **Enter** to accept the default answer:

Table 8-1: ConfigureJBoss ant Target Prompts

Prompt	Response
Please specify the name of the Model N JBoss property file: [local_jboss]	Enter the name of the Model N property file used by the Model N application deployed on JBoss. It is not necessary to enter a path. The system will search for it by using the `[external directory/props/` *`(prop filename)`*`]`.
Enter JBoss configuration name [modeln]:	Enter the name of the JBoss server configuration that should be created for deploying the Model N application. A directory by this name will be created under `<JBossHome>/jboss-as/server`, therefore, use a name with no white-spaces. If deploying multiple instances of the Model N application on this server, assign a unique, un-used port number for each of the instances.

Table 8-1: ConfigureJBoss ant Target Prompts (Continued)

Prompt	Response
Enter a unique number for the JBoss instance if deploying multiple instances of the Model N application on this server: [0]	This number will be used as a multiplier to generate unique port numbers for JBoss services to use for this instance of the application. **Note:** You also need to configure the property `com.modeln.mnInstancePrefix` with the same value which will be used by the Model N application to generate unique port numbers for its own services.
Please specify if you need to configure SSL on JBoss?	Enter 'y' if you need to configure SSL on JBoss. Otherwise, enter 'n' to skip the next 3 SSL related questions.
Enter the keystore format (for example: JKS, PKCS11 or PKCS12):: [JKS]	Enter the keystore format.
Enter the keystore file path:	Enter the location of the keystore.
Enter the key store password:	Enter the keystore password.
Enter JBoss port number [7001]:	JBoss server port, <JBossPort> that the server will listen on. If deploying multiple instances of the Model N application on this server, assign a unique, un-used port number for each of the instances.
Enter Data Source name (for example: mn551phOraDataSource) (Required):	Enter the name of the data source to be created on the JBoss server configuration for the Model N application.
Enter database user name (Required):	Enter the name of the Model N database user name.
Enter database password (Required):	Enter the Model N database password.
Enter database JDBC Url (for example: jdbc:oracle:thin:@dbserver:1521:mndb11g) (Required):	Enter the JDBC URL for the Model N database.
Please specify if you need to configure the ETL datasource?	Enter 'y' if you need to configure the ETL data source. Otherwise, enter 'n' to skip the next 4 ETL data source related questions.

Table 8-1: ConfigureJBoss ant Target Prompts (Continued)

Prompt	Response
Enter ETL Data Source name (for example: mn551phaOraDataSource) (Required):	Enter the name of the Datasource to be created on the JBoss server configuration for the ETL database.
Enter ETL database user name (Required):	Enter the name of the ETL database user name.
Enter ETL database password (Required):	Enter the ETL database password.
Enter ETL database JDBC Url (for example: jdbc:oracle:thin:@dbserver:1521:mndb11g0 (Required):	Enter the JDBC URL for the ETL database.
Would you like to backup any existing files in the <JBoss_Configuration> configuration?	Enter 'y' if you would like to create a backup of the JBoss server configuration files. The backup files are created in a timestamped folder under <JBossHome>/jboss-as/server/<JBoss_Configuration>/backup/<timestamped-folder>.

d. The JBoss server configuration files are created under `<JBossHome>/jboss-as/server/<JBoss_Configuration>`. These files can be edited directly to make any other required configuration changes.

Note: Re-running the ConfigureJBoss target will replace any manual changes to the configuration files so it is recommended to take a backup for the configuration folder.

---

e. Helper scripts are created in the tools `build/bin` directory to start and stop the JBoss server configuration created using the target.

- Go to the tools build/bin directory:
  On Windows: `c:\modeln\curBuild\tools\build\bin`
  On Linux: `/opt/mn/release/tools/build/bin`
- Use `./startModelN.sh(.bat)` to start JBoss
- Use `./stopModelN.sh(.bat)` to start JBoss

8.3 Administering JBoss

8.3.1 How to Start JBoss Server Configuration with a Model N Deployment

1. Go to the tools build/bin directory:
    - On Windows: `c:\modeln\curBuild\tools\build\bin`
    - On Linux: `/opt/mn/release/tools/build/bin`
2. Run the command:
    - On Windows: `startModelN.bat`
    - On Linux: `./startModelN.sh`

8.3.2 How to Stop JBoss Server Configuration with a Model N Deployment

1. Go to the tools `build/bin` directory:
    - On Windows: `c:\modeln\curBuild\tools\build\bin`
    - On Linux: `/opt/mn/release/tools/build/bin`
2. Run the command:
    - On Windows: `stopModelN.bat`
    - On Linux: `./stopModelN.sh`

8.3.3 Running JBoss as Windows Service

The JBoss server configuration can be deployed as a Windows service and the service managed through the Windows Services panel.

8.3.3.1 How to Install the Windows Service for JBoss

1. Go to the tools `build/bin` directory:

    ```
    c:\modeln\curBuild\tools\build\bin
    ```

2. Run the command:

    ```
    installService.bat
    ```

8.3.3.2 How to Remove the Windows Service for JBoss

1. Go to the tools `build/bin` directory:

    ```
    c:\modeln\curBuild\tools\build\bin
    ```

2. Run the command:

    ```
    uninstallService.bat
    ```

8.4 How to Redeploy a .ear File

To redeploy an application under a previously configured JBoss server configuration:

1. Shut down the JBoss server.
2. Copy the new .ear file in the location of the earlier Model N distribution .ear file deployed on JBoss.

8.5 How to Configure Apache HTTP Server 2.2.3 for JBoss

To configure the Apache server to work with JBoss:

1. Install the static content into the HTTP server directory.
   The static content required by the application is contained in the `<`*`AppName`*`>-docroot.jar` file provided. Extract the contents of this archive into the HTTP server docroot. If you do not know where the docroot is located, look in the file `<APACHE_HOME>/conf/httpd.conf` for the following entry:

   ```
   DocumentRoot "</location/of/static/content>"
   ```

   Note: If you installed the Red Hat RPM, the docroot directory is found in `/var/www/html`.

   ---

2. Download the `mod_jk` (version 1.2.31) package from Tomcat's 'Download Tomcat connector section' page (http://tomcat.apache.org/download-connectors.cgi). Under the Headlines section, click the "binaries" link to go to the download page and then download the correct file depending on your platform.
3. Rename the lib `mod_jk.so` and add it to the modules directory under the Apache HTTP server installation.
4. Open the `httpd.conf` file and add the following line at the very bottom of file.

   ```
   # Include mod_jk configuration file

   Include conf/mod-jk.conf
   ```

5. Copy the `mod_jk.conf`, the `uriworkermap.properties` and the `workers.properties` files from the following tools directory to the conf directory in the Apache HTTP server installation directory.
   - On Windows:
     `c:\modeln\curBuild\tools\build\templates\appserver\jboss\apache`
   - On Linux:
     `/opt/mn/release/tools/build/templates/appserver/jboss/apache`
6. Restart Apache.

Note: You can verify the syntax of the `httpd.conf` file by running `<APACHE_HOME>/bin/apachectl -t` at anytime.

9

Installing the Reporting Engine

Model N reporting provides reporting capabilities to the Model N application suite. Model N reporting allows you to create, execute, schedule, and store reports.

Model N reporting is based on Cognos. Cognos is a reporting solution provided by IBM Inc. It is a complete application with components that reside on a web server, an application server, and tables in a database.

Cognos consists of several applications including:

- Cognos Connection (a web-based portal for accessing the content store and other services)
- Cognos Report Studio (a web-based report design tool running only on Internet Explorer)
- Cognos Query Studio (a lighter weight web-based query tool)
- Cognos Report Viewer (a web-based tool for running and viewing reports)
- Framework Manager (a Windows-based tool for design reporting models)

To use Model N reporting, you must have Cognos installed, configured, and customized for Model N. The steps to achieve this are not automated since they are highly dependent on how you choose to deploy Cognos. The steps below describe a basic single node Cognos installation.

Note: See the *Cognos Quick Start and Configuration Guide* for additional background information as well as information specific to your operating system. This guide is shipped with Model N's documentation.

This chapter covers the following topics:

- User Roles
- Cognos Installation Requirements
- Installing and Setting Up Cognos
- Installing and Setting Up to Integrate Model N and Cognos
- Frequently Asked Questions

9.1 User Roles

Reporting users fall into the following roles:

- Report End Users
- Report Administrators
- Report Developers

9.1.1 Report End Users

Report end users are users who run and view reports. For these users only, Cognos reporting is completely embedded into the Model N application. Report end users can refer to the Model N *Reporting Guide*. Business authors are a subset of report end users who create ad-hoc reports using Cognos Query Studio which is embedded within the Model N application. Therefore, they need to understand how to use Cognos Query Studio. For information on Cognos Query Studio, refer to the *Cognos Query Studio User's Guide*.

9.1.2 Report Administrators

Report administrators manage the reporting system as a whole and also individual reports. To do this, they need to interact with both the Model N application and the Cognos applications and to understand and be aware of all of them.

Report administrators typically use the Model N application to control access to reports. They use various Cognos tools including Cognos Configuration and Cognos Connection to configure and troubleshoot the system. Report administrators can use Model N-provided scripts to automate packaging and deployment to Cognos.

9.1.3 Report Developers

Report developers develop reports for use by report end users. Report developers interact with the Model N application and the Cognos applications and need to understand and be aware of both.

Report developers create reports using Cognos Report Studio and run them from within the Model N application. Model N has extended Cognos prompts so they must be executed by the Model N application. Prompts are a mix of built-in Cognos Report Studio functionality, Model N extensions using Cognos extension points, and Model N Java based plug-ins. For more information on prompts, see Chapter 6, Creating and Using Model N Prompts, in the Model N *Reporting Guide*.

Report developers use the Model N report generator to generate models. For more information on models, see the Model N *Reporting Guide*.

9.2 Cognos Installation Requirements

In order to install Cognos the following is required:

- A database with a UTF character set for storing the Cognos content store. Most installations are typically installed on an Oracle database. For information the UTF character set, see How to Choose a Character Set.

Note: It is recommended that Cognos be installed on a separate instance from the Model N application suite.

- Cognos 8.4.1 requires Java 5.0. Use the latest release of Java 5.0 for environment.
- On Linux, Cognos 8.4.1 requires RedHat 5.1 or later. Cognos Configuration uses open motif 2.2.
- A Java application server to host Cognos Java-based services. The Java application server can be WebLogic, WebSphere, or Tomcat. Cognos comes with an embedded version of Tomcat which would typically be used for simple installations.
- A web server to host the Cognos gateway static web content. A separate web server is recommended in situations where the Cognos application is being deployed to an application server. For default installations that use the embedded Tomcat installation, a web server may not be required.
- An Oracle client on the machines on which Cognos is deployed. Cognos uses Oracle's native SQL*Net client to connect to the Model N database.
- A 32-bit or 64-bit machine. See the Supported Platform Matrix for specific details.
- Although Cognos 8.4.1 includes 64-bit binaries, it requires Oracle's 32-bit libraries. When connecting to Oracle 11g R2, you must install the Oracle 11g R2 32-bit client.

Cognos is clusterable and load-balanced with the same services on multiple machines or different services on different machines and with different tiers (web, application, and content) on the same or different machines. However, Cognos can also be used in a single node configuration. Refer to the *Cognos Architecture and Planning Guide* for additional information about options for deploying Cognos.

9.2.1 Performance Planning

In a default configuration, Cognos reports against the Model N transactional database rather than a separate reporting database. In your configuration, factor in the performance effects of reports including scheduling reports, report sizes, and the number of concurrent users. Refer to the *Cognos Architecture and Planning Guide* for performance planning guidelines.

9.2.2 Cognos Functionality

All Cognos functionality other than Framework Manager is available for use during deployment. Only a portion of Cognos functionality is embedded within Model N. If you would like to use certain Cognos functionality such as bursting, batching, or emailing, which are not embedded within the Model N application, you are free to do so. Framework Manager is licensed only to Model N Professional Services.

9.3 Installing and Setting Up Cognos

This section provides steps for:

- How to Install Cognos
- Setting Up Cognos

9.3.1 How to Install Cognos

The following steps describe how to install base Cognos, Cognos 8 SDK, and any Cognos hotfixes. Install Cognos prior to configuring or customizing Model N.

Note: It is recommended that Cognos be installed on a separate instance from the Model N application suite.

Refer to the *Cognos Installation and Configuration Guide* and the *Cognos Quick Start and Installation Guide* for additional background information and for information specific to your operating system.

Note: When installing Cognos on a UNIX operating system, Cognos must be installed as root if installed under the default `/opt` directory.

9.3.1.1 How to Install IBM Cognos 8 Business Intelligence

The following steps describe how to install IBM Cognos 8 Business Intelligence. Model N recommends that you read the *Cognos Quick Start Installation and Configuration Guide* and refer to it as you go through the following steps.

Note: You must have Java on your machine to run the Cognos Installation Wizard. Before you install Cognos, ensure that the `JAVA_HOME` environment variable is set to point to the installation location of your Java Runtime Environment (JRE).

To install IBM Cognos 8 Business Intelligence:

1. On the command line, unzip the Cognos gzip file that is included with the Model N application by entering:

   ```
   gzip -d filename.tar.gz
   ```

   where *filename* is the file specific to the operating system as noted in the following table.

   Table 9-1: Cognos gzip File Names

Operating System	File Name
Sun Solaris 10	CZA8FMZL.tar.gz
Windows Server 2003 (64-bit)	CZA87ML.tar.gz
Red Hat Enterprise Linux RHEL 5.2 (32-bit)	C1XA1EN.tar.gz

   You can obtain Cognos from the same place where you obtained the Model N application binaries, that is, either from the Model N Customer Portal or from the FTP site.

2. Decompress the IBM Cognos 8 Business Intelligence tar file, which is created after you unzip the gzip file, by entering

   ```
   tar -xvf filename.tar
   ```

   on the command line.

3. Change the directory to the appropriate operating system-specific directory for the operating system you are using.
4. Launch the Cognos Installation Wizard.
   - On Windows, double-click the `issetup.exe` file.
   - On UNIX, run `./issetup`.
   - On Linux, run `./issetup`.

   On UNIX and Linux, the `issetup` program requires XWindows. If you do not have XWindows, you can use the `issetupcc` program instead.

   The Cognos Installation Wizard window appears.

5. In the window that appears, select **English**.
6. Click the **Guide** button to locate one place to view the Cognos installation documents.
7. Click **Next**.
8. Accept the license agreement.
9. Click **Next**.
10. In the Cognos Installation Wizard, choose the installation directory.

The directory used for IBM Cognos 8 Business Intelligence is `/opt/cognos/c8`. In Linux, the `opt` directory is used for all software and add-on packages that are not part of the default installation.

11. Click **Next**.
12. Click **Yes** to create the directory.
13. Choose to install *all* components, which is the default choice.
14. Click **Next**.

    The current settings are displayed.

15. Click **Next**.

    Cognos is installed. This may take several minutes.

16. Click **Finish**.

9.3.1.2 How to Install Cognos SDK

To run the model generation tool, you need to install the Cognos SDK on the Cognos server that the generation tool is pointing to. You do not need to install the Cognos SDK if you are not generating models.

To install Cognos SDK:

1. Unzip the Cognos SDK tar/gzip file, which is included with the Model N application, by entering `gzip -d filename.tar.gz` on the command line.

   The file name is `CZAE4EN.tar.gz` for 32bit and `CZAE5EN.tar.gz` for 64bit.

2. Decompress the Cognos SDK tar file, which is created after you unzip the gzip file, by entering `tar -xvf filename.tar` on the command line.
3. Change the directory to the appropriate operating system-specific directory for the operating system you are using.
4. Launch the Cognos Installation Wizard by running the script, `./issetup`.

   The Cognos Installation Wizard window appears.

5. In the window that appears, select English.
6. Click the Guide button to locate one place to find Cognos installation documents.
7. Click Next.
8. Accept the license agreement.
9. Click Next.
10. Choose the installation directory.

    The directory used for IBM Cognos 8 Business Intelligence is `opt/cognos/c8`. In Linux, the `opt` directory is used for all software and add-on packages that are not part of the default installation.

11. Click Next.
12. Click Yes to create the directory.
13. Choose to install all components which is the default choice.
14. Click Next.

The current settings are displayed.

15. Click Next.

    Cognos is installed. This may take several minutes.

16. Click Finish.

9.3.2 Setting Up Cognos

This section describes what to do when running Cognos.

9.3.2.1 How to Set Environment Variables

Set the following environment variables whenever you run Cognos:

1. Copy a Sun JRE (java runtime environment) that is at least as recent as `sun_jre_1.5` to a subdirectory within the directory where you have installed Cognos.

   For information on how to update the Java environment, refer to the Update the Java Environment section in the *Cognos Quick Start Installation and Configuration Guide.* Model N recommends that you have a separate JRE to isolate Cognos' additions to the JRE.

   Cognos requires that authentication plug-ins be compiled against a Java version that is not newer than the version of Java in which Cognos is running. Model N's authentication plug-in is compiled against Java 1.5.

2. Set `JAVA_HOME` to the one of the following paths:

   If you have the JRE installed, set `JAVA_HOME` to the JRE directory.

   If you have a JDK installed, set `JAVA_HOME` to the JDK's JRE subdirectory.

Note: For all Windows environments, if Cognos is installed locally and uses its own version of Java, make sure that the `PATH` environment variable does not contain the path to `JAVA_HOME/bin`.

3. Set `CRN_ROOT` to the path of the installed Cognos application.

4. Unset `JAVA_OPTS`.

5. Set `ORACLE_HOME` to the path to Oracle.

   This is probably already set if you have Oracle client set up for the root user.

6. Copy $ORACLE_HOME/jdbc/lib/ojdbc5.jar to $CRN_ROOT/webapps/p2pd/WEB-INF/lib/

7. Copy bin/jre/1.5.0/lib/ext/bcprov-jdk14-134.jar to /opt/cognos/c8_64/sun-jdk-1.5/jre/lib/ext.

Note: You can not use the bcprov-jdk14-134.jar supplied in the Cognos 8.4.1 installation. You must use the version available from the Model N Customer Portal.

8. If you are running on UNIX, append the `<ORACLE_HOME>/lib` directory to the appropriate library path environment variable:
   - For Linux, `LD_LIBRARY_PATH`
   - For Solaris, `LD_LIBRARY_PATH`

     For Cognos to work on 64-bit UNIX environments, it needs Oracle 32-bit libraries. The `LD_LIBRARY_PATH` has to have `<ORACLE_HOME>/lib32` before `<ORACLE_HOME>/lib`.
9. If you are running on UNIX, for information on appending the `$CRN_ROOT/bin` directory, refer to the Install the Server Components section in the *Cognos Quick Start Installation and Configuration Guide*.

9.3.2.2 How to Set Up Java with Security Files

For steps on how to set up Java with security files the first time you run Cognos, follow the Update the Java Environment section in the *Cognos Quick Start Installation and Configuration Guide.*

Note: The `/dev/random` device is available and configured to initialize automatically on server start-up.

9.3.2.3 How to Set Up Cognos to Connect to Your Database

For steps on how to set up Cognos to connect to your database, see the Set Up the Database Client for the Content Store section of the *Cognos Quick Start Installation and Configuration Guide.*

How to Configure the ReportSchema Data Source

Model N reports and models use a designated data source named `StarSchema`. Note that if you use the cognos schema prefix, this will actually be `<CognosSchema>StarSchema`. This data source must point to the ETL reporting database using the correct TNS name, userid, and password.

Setting up the ETL Data Source in Cognos

Verify that you have created the ETL database instance, then link Cognos to this Star schema by doing the following:

1. Go to your Cognos UI at http://<server>:<port>/cognos8.
2. In the top right corner, go to **Launch > Cognos Administration**.
3. Select the **Configuration** tab, then the menu **Data Source Connectios**.
4. Create a new data source, called *StarSchema*, and click **Next**.
5. In **Type**, select *Oracle,* then click **Next**.
6. Enter the **SQL*Net connect string** (this is usually similar to *mndb11g*).
7. Select the **User Id** and **password** check boxes, then enter the username and password that Cognos will use to connect to the ETL database.
8. Test the connection, ensuring that it succeeds. To test it, click **Directory > Data Sources > ReportingSchema > Test**.

9. Click **Next** to finish.

9.3.3 How to Create the Cognos Content Store Database

This section provides important information about creating the Cognos content store database. If you have already created a database instance that meets the Cognos requirements, you do not need to follow the instructions in section 3.3.

9.3.3.1 How to Choose a Character Set

For steps on how to choose a character set, see the Steps for Oracle section in Setting Up the Environment in the *Cognos Quick Start Installation and Configuration Guide.*

If you already have a database that supports international character sets you need only to create a new user and assign permissions to it. If you do not have a database, you need to create one. A typical way to create an Oracle database is to use the Oracle Database Configuration Assistant (Oracle DBCA). If you already have a database, but it does not support international character sets, you can choose to update your database to support international character sets by exporting your entire database, removing the database, recreating your database with an international character set such as UTF-8, and then importing the database. This converts the imported data to the database's character set.

9.3.3.2 How to Determine User Account Permissions

To determine user account permissions, refer to the Steps for Oracle section in Setting Up the Environment in the *Cognos Quick Start Installation and Configuration Guide.*

For this user, Model N recommends that you create separate tablespaces, default and temporary, at least for your production environment. To determine the tablespace size, consult the *Cognos Architecture and Planning Guide*. For information on how to create these tablespaces, consult your database administrator.

The following is an example of SQL data control language (DCL) for creating a user with the required permissions for the user `mnadmin` with DBA and RESOURCE role assignments to include the SQL data manipulation language (DML) and data definition language (DDL) permissions:

```
CREATE USER mnadmin IDENTIFIED BY mnadmin;

GRANT CONNECT TO mnadmin;

GRANT RESOURCE TO mnadmin;

GRANT CREATE VIEW TO mnadmin;

GRANT CREATE TRIGGER TO mnadmin;
```

Note: The last two grant statements may not be required on Oracle 9i. You may only need them on Oracle 10g.

You should now have a base Cognos application. However, for this application to work with Model N, additional installation steps are needed. These steps are described in Installing and Setting Up to Integrate Model N and Cognos.

Note: See the section on performance tuning in the *Cognos Architecture and Planning Guide* for sizing requirements.

9.3.4 How to Configure Cognos

Cognos stores most of its configuration information in the Cognos content store. Bootstrap information needed to configure Cognos is stored in `$CRN_ROOT/configuration/cogstartup.xml`.

The shell script, `cogconfig.sh`, which is stored in the `$CRN_ROOT/bin` directory, launches the application Cognos configuration to set up the values in this file. This application also has controls to start and stop Cognos. Also in the `$CRN_ROOT/bin` directory, the shell scripts, `startup.sh` and `shutdown.sh`, can also be used to start up and shut down Cognos respectively.

Note: All URIs configured for Cognos must specify a full DNS name. For example, you should use `pddev.modeln.com` instead of `pddev` or `localhost`.

Note: The `inactivityTimeout` parameter in the `$CRN_ROOT/configuration/cogstartup.xml` file must be larger than the interval used by the Model N application to check the Cognos connection.

To configure Cognos:

1. Start Cognos Configuration by running the `cogconfig.sh` script.
2. Configure the content store under Data Access / Content Manager, and save the configuration.

   The steps to configure the content store are in the Set Database Connection Properties for the Content Store section in the Configuring the Server Components section in the *Cognos Quick Start Installation and Configuration Guide.*

Note: Enter the User ID and Password that you created in How to Create the Cognos Content Store Database. The Service name that you enter is the SID for the Oracle database. To test that Cognos can connect to the content store, in Cognos Configuration right click on the Content Store node in the Explorer tree and select **Test**.

---

3. If you are using a web server, refer to the Configure Web Server section in the *Cognos Quick Start Installation and Configuration Guide.*

If you are using Apache, add the following to `httpd.conf`:

```
<VirtualHost *:80>
    ScriptAlias /cognos8/cgi-bin/<Path to CRN_ROOT>/cgi-bin
    Alias /cognos8 <Path to CRN_ROOT>/webcontent
</VirtualHost>
```

Where `CRN_ROOT` is replaced with the actual directory paths.

Note: If you are running a version of Apache that has `suexec` compiled with it, make sure that the user that Apache runs this host as has execute permissions for `$CRN_ROOT/cgi-bin` and for `$CRN_ROOT/webcontent`. For information on how to do this, refer to Apache reference material. Apache needs to be re-started for changes to take effect.

---

4. Disable the Cognos Application Firewall (CAF):
    a. In Cognos Configuration, in the left navigation panel, go to **Security > Cognos Application Firewall**.
    b. In the right navigation panel, choose False from the Value drop-down to answer the question, “Enable CAF validation?”.
    c. Save your changes.

9.3.5 How to Start Cognos

You should now be able to start Cognos. For more information on how to start Cognos, refer to the Test the Server Components section in the *Cognos Quick Start Installation and Configuration Guide*. To start Cognos:

1. Click the green arrow icon to start Cognos.

   Cognos reports whether startup was successful. The first startup may take a minute or two because Cognos creates tables in the content store.

2. While Cognos tests the mail server connection, you will see the following warning:

   The test phase has warnings. Do you want to continue?

   Click Yes to continue.

3. If startup was not successful, check the log files contained in `$CRN_ROOT/logs` for any errors in the `tomcat.log` file and the `cogstartup.xml` file.

4. Double check that Cognos is successfully started by using a browser to go to the Cognos connection web page at `http://<machine_name>:80/cognos8/cgi-bin/cognos.cgi`.

5. For more information on why Cognos did not start, consult the Frequently Asked Questions as well as the *Cognos Troubleshooting Guide*.

9.3.6 How to Configure Web Browser Clients

For information on configuring web browser clients, refer to the Configure Web Browsers section of the *Cognos Quick Start Installation and Configuration Guide*.

9.4 Installing and Setting Up to Integrate Model N and Cognos

This section describes the steps used to install Model N Customizations for both Cognos and Model N.

1. How to Install the Model N Customization Package
2. How to Configure Cognos for Model N
3. How to Configure Model N for Cognos

9.4.1 How to Install the Model N Customization Package

Model N customizes several Cognos user interface components as well as provides Model N reports and models.

The Model N customization files:

- contain Model N reports and models broken out by Model N module;
- override several JavaScript and .xml files to show only relevant functions and for legacy multi-realm support;
- override Cognos ReportStudio configuration files to add Model N prompts;
- add a Model N authentication provider plug-in into Cognos' authentication.;
- configure the Cognos user interface to hide some functionality;
- and provide a Model N skin.

  To use the skin, you must assign it to the Model N Cognos user using the Users and Groups tab in Cognos Connection.

These customizations and the reports and models are packaged in the following zip files. Extract these zip files into appropriate locations in order to overwrite the existing Cognos user interface components and to place the reports and models into the Cognos deployment directory so that you can deploy them later:

- the `cognos-docroot.zip` file
- the `cognos-webapp.zip` file

To install the Model N customization package:

1. Shut down Cognos.
2. Extract the files in the `cognos-docroot.zip` file to the document root of your web server application. For example, if you deploy Cognos on the Apache web

server with the below configuration, you need to unzip the `cognos-docroot.zip` file to the `$CRN_ROOT`.

```
<VirtualHost *:80>

    ScriptAlias /cognos8/cgi-bin/ <Path to CRN_ROOT>/cgi-bin

    Alias /cognos8 <Path to CRN_ROOT>/webcontent

</VirtualHost>
```

3. Extract the files in the `cognos-webapp.zip` file to the `$CRN_ROOT`.
4. To ensure that Model N and Cognos can share browser cookies, you need to configure Cognos and Model N to be in the same domain. This step ensures that Cognos is configured for your domain. Change all of your environment properties so that they use your full machine name including domain, instead of localhost.
    a. Start Cognos Configuration by running the `cogconfig.sh` script in Linux or cogconfig.bat script in Windows.
    b. Select the Environment node in the left navigation.
    c. In the right Environment - Group Properties section, change all URIs to refer to your full domain name instead of localhost.

    For example, the default Gateway URI value is `http://localhost:80/cognos8/cgi-bin/cognos.cgi`. If your machine name is pd.modeln.com, you would change the Gateway URI to `http://pd.modeln.com:80/cognos8/cgi-bin/cognos.cgi`.

    d. Save your settings.

    Cognos Configuration is a tool that modifies the Cognos bootstrap configuration file called `cogstartup.xml`.
5. Start Cognos.

9.4.2 How to Configure Cognos for Model N

This section describes how to configure Cognos for Model N.

9.4.2.1 How to Create a Data Source Using Cognos Connection

The models and report templates provided in the `cognos-webapp.zip` file refer to a data source named `ReportingSchema`.

To create a data source named `ReportingSchema` on the Cognos server referencing the Model N database, you can either create it through the Cognos user interface or you can configure Model N to create the data source as needed whenever you run the Model N application or run Model N reporting tools. Both methods are described in this section.

You must configure the number of connections used by Cognos so that the total number of connections used by all applications (Model N, Cognos, any other) does not exceed the number of connections allowed by your Oracle database configuration. For information on how to configure these connections, see the Database Connections section in the Monitoring chapter of the Model N *Operations Guide*.

How to Create the Data Source Through the Cognos User Interface

To create the data source through the Cognos user interface:

1. Start the Cognos server.

   Make sure that when you start the Cognos server, the `LD_LIBARY_PATH` is set to refer to Oracle 32 bit libraries before the regular libraries.

   If you are using the default configuration, start Cognos by starting the Jakarta Tomcat application server that is bundled with Cognos after installing Cognos by either clicking the button (the green arrow icon) on the toolbar, or by running the `startup.sh` (on Windows, `startup.bat`) script in the `bin` directory of the Cognos installation.

   If you are using a different application server, perform the steps that are specific to your application server to start Cognos. For information on starting Cognos on WebLogic, see *Configuring WebLogic for Cognos*. For information on starting Cognos on WebSphere, see *Installing and Configuring WebSphere for Cognos*, *Configuring WebSphere and Distributed Cognos*, and *Cognos and WebSphere Clustering*.

2. Open Cognos Connection using the external gateway URI:

   `http://localhost:80/cognos8/cgo-bin/cognos.cgi`.

3. In the Tools drop-down menu, click **Directory**.
4. Click Data Sources.
5. Click New Data Source.

   The New Data Source Wizard appears.

6. Enter `ReportingSchema` in the Name: field.
7. You can enter a description and a screen tip, which is the text that appears when you mouse over an icon.
8. Click Next.
9. In the Connection page, select Oracle for the Type: in the Type: drop-down list. This is your reporting database type.
10. By default, the isolation level is selected. Use the default.
11. Click Next.

    The connection string page for the selected database is displayed.

12. Specify parameters for the database connection string. For Oracle, the SQL*Net connect string is the Oracle database tnsname entry (SID) for your reporting database.
13. Check the user ID and password. Type in a user ID and password to use when signing on to the database. Leave the "*Create a signon that the Everyone group can use*" check box selected.
14. Leave the collation sequence blank since there is none.
15. Click Test to test the connection.

    The status should say succeeded. If the connection is not successful, consult the Frequently Asked Questions or the *Cognos Troubleshooting Guide*.

16. Click Close.
17. Click Finish.

    You have successfully created a data source that Model N uses to run reports against.

How to Create the Data Source Through the Model N Application

Configuring the data source through the Model N application is useful if you are deploying multiple Cognos instances.

The following flags drive whether the Model N application creates or updates the data source connection:

Flag Name	`com.modeln.ReportingSvc.createDataSource`
Description	If set to false, the data source is not created. If set to true, the data source is created.
Possible Values	True and False
Default Value	False

Flag Name	`com.modeln.ReportingSvc.updateExistingDataSource`
Description	If set to true and if `com.modeln.ReportingSvc.createDataSource` is set to true, the existing data source will get updated even if one already exists. Otherwise, no updates will occur.
Possible Values	True and False
Default Value	False

Flag Name	`com.modeln.ReportingSvc.dataSource.username`
Description	Data source user name which is the user name to login to the Model N database.
Possible Values	the data source user name
Default Value	mnph

Flag Name	`com.modeln.ReportingSvc.dataSource.password`
Description	The data source password which is the password used to log in to the Model N database.
Possible Values	the data source password
Default Value	mnph

Flag Name	`com.modeln.ReportingSvc.dataSource.connectionString`
Description	The data source connection string (for Oracle, the SQL*Net connection string). For example, the `tnsname` entry for the Model N database on the client.
Possible Values	the data source connection string
Default Value	mndb9i

The data source creator only creates Oracle data source connections. If you need to create a different type of connection, you can alter the generated Cognos connection string using the following prefix and suffix properties.

The string is generated by concatenating a prefix and a suffix to `com.modeln.ReportingSvc.dataSource.connectionString`.

the prefix property	`com.modeln.ReportingSvc.dataSource.connectionPrefix`
the default value	`^User ID:^?Password:;LOCAL;OR;ORACLE@%s@`

the suffix property	`com.modeln.ReportingSvc.dataSource.connectionSuffix`
the default value	`/%s@COLSEQ=`

9.4.2.2 How to Shut Down Cognos

To shut down Cognos, click the Stop button (the black square icon next to the Play button) on the toolbar, or run the `shutdown.sh` (on Windows `shutdown.bat`) script in the `bin` directory of the Cognos installation.

Note: In Windows, Cognos installs a Windows service that allows you to control the startup and shutdown process.

9.4.2.3 How to Configure Cognos to Use a Custom Authentication Provider

Cognos supports authentication. Authentication can be through Cognos itself, through LDAP, and through custom Java authentication plug-ins. By default, Cognos authentication is disabled. Therefore, you can connect to Cognos without logging in.

Model N comes with a sample Java authentication plug-in. This plug-in reads the `Authentication.properties` file in Cognos installation's configuration directory and compares the user ID and password passed in against the values in the file.

Note: If you install Cognos on WebSphere or WebLogic, the configuration directory will be placed into the `EAR/WAR` file. Whenever you update the properties, regenerate the `EAR/WAR` file and redeploy it.

The following steps describe how to enable Model N authentication. For additional information on Cognos authentication, see the Security Model chapter in the *Cognos Administration and Security Guide*.

How to Add the Model N Custom Authentication Provider

1. Start Cognos Configuration.

   In Linux, run `./cogconfig.sh`. In Windows, run Cognos Configuration.

   The graphical user interface is displayed.
2. Right click Authentication.
3. Mouse over New Resource.
4. Click Namespace.
5. Type a name in the window that appears.
6. Choose Custom Java Provider from the drop-down list.
7. In the Namespace ID field, type MN.
8. In the Java class name type `com.modeln.bp.reporting.authentication.CMnAuthenticator`.
9. Save.
10. Click **Cognos** under **Security > Authentication**.
11. Change allow anonymous access? to False which forces the user to login whenever going to the Cognos Connection page.

    The default user ID is mnadmin and the default password is mnadmin.

12. Restart Cognos. See the section called How to Start Cognos.

How to Use the Authentication.properties File

The Java plug-in registered above compares the user ID and password provided by a user with the values stored in the `Authentication.properties` file, which is in the Cognos `configuration` subdirectory. To change the user ID and password, change them in the `Authentication.properties` file. To change the Namespace ID, change it both in the Cognos Configuration application and in the `Authentication.properties` file.

Whenever you make changes to the `Authentication.properties` file, or change the configuration through Cognos Configuration, you need to restart Cognos. See the section called How to Start Cognos.

See the *Cognos Administration and Security Guide* for more information about security.

Values in the Cognos authentication configuration above, including the namespace name and ID, must equal property values in the Model N property file. If values other than the default namespace name and ID (`MN`) are used, you must change the Model N property file. For information on setting the namespace, user ID, and password, that the Model N application will pass to Cognos, see How to Configure Model N for Cognos.

9.4.3 How to Configure Model N for Cognos

Configure Model N Reporting to communicate with Cognos by setting URL and authentication properties in a property file (in the deployment environment, the *external* property file).

The Model N application integrates with Cognos through URL integration and through SDK integration. Each requires Cognos URLs.

To ensure that Model N and Cognos can share browser cookies, you need to configure Cognos and Model N to be in the same domain. Make sure that your Model N server application URI includes your domain.

The authentication provider compares the namespace, user ID, and password that are passed into it with the namespace, user ID, and password that are stored in the `Authentication.properties` file. If they match, the user is authenticated. The Model N application needs to be configured to pass the correct namespace, user ID, and password.

Each of the Model N Reporting properties to set is listed below, with an example of values and a description of the meaning of the property.

The following properties are for URL/SDK integration:

Table 9-2: Report Server URL

Property Name	`com.modeln.ReportingSvc.reportServerURL`
Default Value	`http://`**`<FullCognosServerName>`**`:80/cognos8/cgi-bin/cognos.cgi`

Table 9-2: Report Server URL

Description	This is the Cognos external gateway URL. You must specify a full DNS name instead of localhost. For example, if your machine name is `pddev.modeln.com`, your `reportServerURL` should be `http://pddev.modeln.com:80/cognos8/cgi-bin/cognos.cgi`. If you are using only Tomcat, the value of this property is the Cognos report dispatcher URL instead. You can configure Cognos to use https, in which case the value will be `https://pddev.modeln.com:81/cognos8/cgi-bin/cognos.cgi`.

Table 9-3: Report Dispatcher URL

Property Name	`com.modeln.ReportingSvc.reportDispatcherURL`
Default Value	`http://`**`<FullCognosServerName>`**`:9300/p2pd/servlet/dispatch`
Description	`This is the external dispatcher URL for Cognos.`

The following properties are for authentication:

Table 9-4: Cognos Domain

Property Name	`com.modeln.ReportingSvc.cognosDomain`
Default Value	`modeln.com`
Description	This is the domain that Cognos is installed in. This must be the same domain as the Model N application's domain. This is required for passing along authentication information.

Table 9-5: Cognos Path

Property Name	`com.modeln.ReportingSvc.cognosPath`
Default Value	`/cognos8`
Description	This is the web server path to Cognos. For example, if you set up Apache as above, you specified that everything beginning with `/cognos8` would go to the Cognos directories. This is required for passing along authentication information.

Table 9-6: Authentication Required

Property Name	`com.modeln.ReportingSvc.authenticationRequired`
Default Value	`true` or `false`
Description	The value is `true` if custom authentication is required. Otherwise, the value is `false`.

Table 9-7: Authentication Namespace

Property Name	`com.modeln.ReportingSvc.authenticationNamespace`
Default Value	`MN`
Description	This is the namespace ID specified for the custom authentication provider. This property is not required if the property, `com.modeln.ReportingSvc.authenticationRequired`, is set to `false`.

Table 9-8: Authentication User

Property Name	`com.modeln.ReportingSvc.authenticationUser`
Default Value	`mnadmin`
Description	This is the user name used to login to Cognos. This property is not required if the property, `com.modeln.ReportingSvc.authenticationRequired` is set to `false`.

Table 9-9: Authentication Password

Property Name	`com.modeln.ReportingSvc.authenticationPassword`
Default Value	`mnadmin`
Description	This is the password used to login to Cognos. This property is not required if the property, `com.modeln.ReportingSvc.authenticationRequired`, is set to `false`.

Note: The authentication property values in the Model N property file, including the namespace ID, and the authentication user and password, must equal values in the Cognos authentication configuration, in the Cognos `Authentication.properties` file.

The following property enables or disables Cognos-based Model N reporting:

Table 9-10: Enables or Disables Cognos-Based Model N Reporting

Property Name	`com.modeln.AppSwitch.disableCognosReporting`
Default Value	`true`
Description	`By default, Cognos-based reporting is disabled. You must set this property to false to enable Cognos-based reporting.`

9.4.4 Importing and Deploying Models and Reports

The steps in the How to Install the Model N Customization Package section place pre-packaged Model N models and reports in the Cognos deployment directory. To import and deploy these models and reports into the Cognos content store, run the following ant task with the `import` option:

```
ant -f ops.xml ReportDoAll
```

Note: As with all ant tasks, these tasks are run on the Model N server, not the Cognos server. The ant tasks link to the application server libraries, but do not run on the application server.

This task:

- Updates the schema name references in the models by running the ant task, ReportSchemaResolver.

- Imports the Cognos archives into the Content Store by running the ant task, ReportImporter.
- Deploys standard, custom, and in-line reports for use by the Model N application by running the ant task, ReportDeployer.

The task only performs these actions against Model N modules that your system is configured with. For example, if you have the Government Pricing module, the task will import and deploy the government pricing models and reports. For more information on the ReportDoAll ant task, see ReportDoAll.

You can execute any of the preceding three tasks individually.

If your Cognos instance is on a separate server from your Model N instance, the deployment tools will have to remotely access the Cognos deployment directory on the Cognos server.

The recommended way to access the deployment tools is to use a directory. To expose the Cognos deployment directory, in Linux installations, you can create a soft link from your archive directory to the Cognos deployment directory. In Windows, you can share the Cognos deployment directory. Update your archive folder path property to point to this directory.

If you do not share the directory and your Cognos archives are not in the Cognos deployment directory, you must run the tools separately as follows:

1. Copy the Cognos archives from the Cognos deployment directory to your archive folder path.

   These Cognos archives were unzipped as part of the Model N Cognos customizations. You need to copy only the Cognos archives to the archive folder path.
2. Run the ReportSchemaResolver tool.
3. Copy the updated Cognos archives to the Cognos deployment directory.
4. Run the ReportImporter tool.
5. Run the ReportDeployer tool.

9.4.4.1 Updating Schema References

Run the report schema resolver ant task with the `add` option:

```
ant -f ops.xml ReportSchemaResolver
```

This updates references to the database schema of the data source in the models.

9.4.4.2 Importing Models and Reports

Run the report import ant task:

```
ant -f ops.xml ReportImporter
```

This imports the appropriate zip files from the Cognos deployment directory. Unzipping the `cognos-webapp.zip` file in the How to Install the Model N Customization Package section copied Module N archive files to the Cognos deployment directory.

9.4.4.3 Deploying Reports

Run the report deployment ant task:

```
ant -f ops.xml ReportDeployer
```

This deploys reports from the Cognos content store folders that the import step created to the Model N deployed reports folders (standard, inline, and template). Custom reports are not deployable this way.

Tip: Make sure that the ant task completes as the tool copies the reports to the appropriate location. If the tool does not complete, some of the reports will not work.

9.4.4.4 Generating Models

Model N comes with pre-packaged reports and models. If you extend the Model N application and wish to report against the additional objects and attributes, you have to re-generate the models.

To re-generate the report models, follow the instructions in the Migrating Model N Reporting chapter in the Model N *Reporting Guide*.

9.4.4.5 Testing Your Installation

1. Log in as a reporting administrator.
2. Click **Reporting > Standard Templates**.

   You should see some reports.
3. Select a report.
4. Fill in report parameters.
5. Run the report.

   You should be able to run it and see results. If you are unable to run the report and see results, consult the Troubleshooting chapter in the Model N *Reporting Guide* or the *Cognos Troubleshooting Guide* or look at the log files for both Model N and Cognos.

9.4.4.6 Applying Patches to Model N Reporting

When a patch to Model N Reporting is released, you must repeat the following installation steps to apply the patch. To apply a patch:

- If the patch specifies that it modifies Cognos, repeat the steps in How to Install the Model N Customization Package.
- If the patch specifies that it modifies models or reports, repeat the steps in Importing and Deploying Models and Reports.

9.5 Frequently Asked Questions

If I do not have a web server, can I still get Cognos up and running?

Yes, you can go directly to the dispatcher, although there are some limitations to doing this. To go directly to the dispatcher:

1. Using Cognos Configuration, set the gateway URL to what your dispatcher URL is currently.
2. Change references to the gateway in the Model N property file.
3. Fix references to the Cognos web server path to `/webapps/p2pd`.
4. Copy the contents of `CRN_ROOT/webcontent` to `$CRN_ROOT/webapps/p2pd/`.
5. Restart Cognos.
6. Restart Model N.

The limitations to going directly to the dispatcher are that:

- You are bypassing the Cognos firewall.
- You need to open up the Cognos port (default 9300) to the outside world.
- If you receive Model N updates, you will need to copy updates from `CRN_ROOT/webcontent` to `CRN_ROOT/webapps/p2pd`.

What if I want to use `https` instead of `http`?

You can. Instructions for doing this are in the *Cognos Installation Guide*.

What if I can't see reports in Model N?

First, determine whether:

- Cognos is available.

  You can determine if Cognos is available by determining whether you can get to the Cognos Connection web page.
- There are reports in Cognos.

  You can determine if there are reports in Cognos by going to Cognos Public folders and then going to the standard folder. If there are reports in that folder, there are reports in Cognos.
- Cognos can connect to the database by testing the datasource connection.
- Model N can connect to Cognos.

  You can determine if Model N can connect to Cognos by:

  1. Checking to see whether `com.modeln.AppSwitch.disableCognosReporting` is set to false.
  2. Looking for reporting exceptions in the Model N logging output.

     The log output is configured from the Model N application. Log into the Model N application as an administrator, go to **Administration > Management > Nodes**. Select the node you wish to configure and select the Logs link in the left panel.
- The Model N user has access rights to the reports.

You can determine if the Model N user has access rights to the standard reports by using Model N administration to look at the roles the user has. The user you log in as must have the Report Administrator role.

How do I know if Cognos started up correctly?

You can look at the Cognos log files.

If you are using the embedded Tomcat application server, you can look at its log file first. This log file will be located in `COGNOS_INSTALLATION/logs/tomcat.log`. A common error is that Tomcat is already started or the port is being used. In that case, you will see the following in the log file:

```
ERROR [main] org.apache.coyote.http11.Http11Protocol - Error
initializing endpoint: <some date, for example: 09 May 2006
19:07:29,599>
```

```
java.net.BindException: Address already in use: <port number by
default this would be 9300>
```

You can determine if a port is being used by using the `netstat` command. If you have browsers pointing to Cognos applications, close them and check to see if the connections are now closed using `netstat`.

If Tomcat successfully starts up, you should see:

```
INFO [main] org.apache.coyote.http11.Http11Protocol: Starting
Coyote HTTP/1.1 on port <port number>: <date>
```

You can then look at the `cogserver.log` file where you should see a series of success messages.

I have enabled the authentication, but now Cognos does not start up.

Check the log files as described in the previous question.

If you are getting a version error in your `cogserver.log` file, your JVM is older than what the authentication `jar` file was compiled with. Upgrade the JVM you use to run Cognos with.

I am getting a firewall error when I try to run a report. How do I fix it?

This error occurs because the Cognos firewall is enabled which limits requests to clients from the same domain as the Cognos server. You must specify the full domain name for the server in the Cognos Configuration URIs. Then either specify all the machines or domains you want to serve or disable Cognos firewall. Refer to the *Cognos Installation Guide* for additional information.

If I am running the ReportDeployment tool and I get validation errors, what do I do?

Test that your data source is correct:

1. Make sure that your database is up to date (that you have run any necessary migration or installation tasks).
2. From Cognos Connection, check that you are logging in to the right database with the right user name and password by testing your data source.
3. Check that the table is present in the database:
    a. Create a report (choose the list view).
    b. Go to the query view.

   c. Drag SQL from the toolbox to the query.
   d. Select the data source in the lower left details section.
   e. Type in a select statement against the table.
   f. Go to the page view.
   g. Select the report page in the tree.
   h. Select the middle model icon in the upper left navigation section.
   i. Drag some attributes to the list.
   j. Run the report (the blue right-hand pointing icon in the toolbar).
   k. If you can run this, your data source is correct (or points to another Model N database).

4. Ensure that the archives contain the correct database schema:

   a. Open the archive containing the model that your report references. For example, the Your Business Models model is contained in the `pharmamodel.zip` file.

   You can determine the model by opening the report in Report Studio and examining the package in its properties or by looking at the contents of the query area in the upper left section (select the left most model icon).

   b. Unzip the archive.
   c. Look in the various `package<n>.xml` files for the keyword `cmDataSource`.
   d. Look for the `ReportingSchema` tag after the keyword.

   The text value contains the schema name which should correspond to the user you used in your data source.

   If the schema name does not correspond to the user you used in your data source, rerun ReportDoAll after configuring it with the correct schema name.

   See ReportDoAll on page 38 for information on how to run ReportDoAll.

   e. Check to see whether the report validates.

If the report still does not validate:

1. Stop Cognos.
2. Clear the contents of the Cognos `data/cqe/RTModels` directory.
3. Start Cognos.
4. Rerun the ReportDeployment tool.

What if I get the Cognos error CAM-AAA-0135?

The Cognos error:

```
Error CAM-AAA-0135: The user is already authenticated in all available
namespaces.
```

may result if the Cognos configuration changes from using authentication to using anonymous access. If this change was made, then all users will need to log out, then log back in before the anonymous access will work for them.

10 Installing Revenue Planning & Intelligence

The Revenue Planning & Intelligence (RPI) application provides pharmaceutical CFOs, finance executives, government pricing and compliance managers with revenue impact and risk assessment capabilities driven by data from revenue management and ERP systems.

10.1 Prerequisites

The following items are required for RPI installation:

- Starter Database dump file of RPI application
- RPI Application Archive WAR file
- SQL script file to create RPI views on Model N application schema - RPIViews.sql

10.2 How to Set Up Views for RPI on the Model N Schema

Execute the script RPIViews.sql on the Model N schema.

10.3 How to Set Up Oracle for RPI

The RPI application uses a different database schema then the Model N application. This schema must be created and database links must be established between the Model N schema and RPI application schema.

1. Create the RPI tablespace with the following command:

```
SQL> CREATE TABLESPACE MN_RPI
DATAFILE '<mount_point>\oradata\<DBInstance>\mn_RPI.dbf'
SIZE 500M AUTOEXTEND ON NEXT 10M MAXSIZE 2048M
EXTENT MANAGEMENT LOCAL UNIFORM SIZE 1M;
```

2. Create the user for RPI on your database. The following SQL commands must be executed using a database administrator account.

```
SQL> CREATE USER <RPI Schema Name>
    IDENTIFIED BY <RPI Schema password>
    DEFAULT TABLESPACE MN_RPI
    TEMPORARY TABLESPACE TEMP
    PROFILE DEFAULT
    ACCOUNT UNLOCK;
```

3. Grant the privileges to the user. The following SQL commands must be executed using a database administrator account.

```
SQL> GRANT CONNECT, RESOURCE, SELECT_CATALOG_ROLE, CREATE
MATERIALIZED VIEW TO <RPI Schema Name>;
```

4. Grant SELECT privilege on the RPI schema to objects on Model N schema. The following SQL commands must be executed using a database administrator account.

```
SQL> GRANT SELECT on <Model N Schema>.MN_RPI_NDC11_VW to
<RPI Schema Name>;
SQL> GRANT SELECT on <Model N Schema>.MN_RPI_NDC9_VW to
<RPI Schema Name>;
```

5. Import the starter database dump to the RPI schema using the command:

```
$ imp userid=system/<system password>@<database-SID>
  file= Exp_<FromRPIDBUser>.dmp
  fromuser=<FromRPIDBUser>
  touser=<RPIDBUser>
  commit=Y
   indexes=N
   buffer=1048576
```

6. Open the RPIProductViews.sql for update. Currently, the script assumes that the name of the Model N schema is MODELN. If this is not the case, update the script with the right Model N schema name.
7. Execute the script RPIProductViews.sql on the RPI schema.

10.3.1 Linking the Model N Database to the RPI Database

After the schema has been imported, database links must be made between the RPI schema and MODEL N schema using the following scripts. Run these scripts while logged in as the RPI user:

```
CREATE DATABASE LINK "<DB-TNSNAMES-ENTRY>@LOOPBACK"
   CONNECT TO "<MODEL-N ORACLE USER>" IDENTIFIED BY "<MODEL N ORACLE 
pwd>"
   USING '<DB-TNSNAMES-ENTRY>';
```

```
CREATE OR REPLACE VIEW an_product_vw AS
SELECT ndc11 PRODUCT,ndc11,description,PRODUCT_GROUP,ver_start_date 
eff_start_date,
ver_end_date eff_end_date FROM MN_RPI_NDC11_VW@<DB-TNSNAMES-
ENTRY>@LOOPBACK;
```

```
CREATE OR REPLACE VIEW an_product_group_vw AS
SELECT
PRODUCT_GROUP,
description,
ver_start_date eff_start_date,
ver_end_date eff_end_date
FROM
MN_RPI_NDC9_VW@<DB-TNSNAMES-ENTRY>@loopback;
```

10.4 How to Deploy the RPI .war File for WebLogic

This section of the guide describes configuring WebLogic for RPI ant the installation of the RPI WAR file.

10.4.1 How to Configure WebLogic for RPI

1. To create a JDBC connection pool, click **<yourDomain> > Services > JDBC > Data Sources** in the left navigation pane and then click New in the right navigation pane.
2. Configure the JDBC connection pool using the following settings:

Name	`RPIDataSource`
JNDI Name	`jdbc/mn`
Database Type	`Oracle`
Database Driver	`Oracle's Driver (Thin) Versions: 9.0.1, 9.2.0,10,11`

3. Click **Next**.

4. Specify the following connection properties:

Database Name	`<DBInstance>`
Host Name	`<DBHost>`
Port	`<TNSPort>`
Database User Name	`<RPIDBUser>`
Password	`<RPIDBPass>`
Confirm Password	`<RPIDBPass>`

5. Click **Next**.
6. To test the data source configuration, click **Test**.

   The Driver Classname should auto populate to `oracle.jdbc.OracleDriver`.

   The JDBC URL should autopopulate to `jdbc:oracle:thin:@`**`<DBHost>:<TNSPort>:<DBInstance>`**
7. Add the following JDBC property in the Properties text box:

   ```
   protocol=thin
   ```
8. Click **Next** to select a target server.
9. Target the new Connection Pool to your new Server instance: **`<AppName>`**
10. Click **Finish**.

    The Summary of JDBC Data Sources page appears.
11. To create a JDBC connection pool, click **<yourDomain> > Services > JDBC > Data Sources** in the left navigation pane and then click New in the right navigation pane.
12. Configure the JDBC connection pool using the following settings:

Name	`RPIIntegrationDataSource`
JNDI Name	`jdbc/di`
Database Type	`Oracle`
Database Driver	`Oracle's Driver (Thin) Versions: 9.0.1, 9.2.0,10,11`

13. Click **Next**.
14. Specify the following connection properties:

Database Name	`<DBInstance>`

Host Name	`<DBHost>`
Port	`<TNSPort>`
Database User Name	`<MNDBUser>`
Password	`<MNDBPass>`
Confirm Password	`<MNDBPass>`

15. Click **Next**.
16. To test the data source configuration, click **Test**.

    The Driver Classname should auto populate to `oracle.jdbc.OracleDriver`.

    The JDBC URL should autopopulate to `jdbc:oracle:thin:@`**`<DBHost>:<TNSPort>:<DBInstance>`**
17. Add the following JDBC property in the Properties text box:

    ```
    protocol=thin
    ```
18. Click **Next** to select a target server.
19. Target the new Connection Pool to your new Server instance: **`<AppName>`**
20. Click **Finish**.

    The Summary of JDBC Data Sources page appears.

10.4.2 How to Install the RPI .war File for WebLogic

This section describes how to actually deploy the RPI application. The RPI web application (WAR) file contains the RPI application. The WAR file, `rpi.war`, is included within the deployment package. To install a new enterprise application, navigate to **Applications > Install New Application** in the WebLogic Administration Console. This initiates the application installation wizard. A series of configuration screens are presented to gather information about various aspects of the application.

To install the WAR file, using the WebLogic Server Administration Console:

1. If using WebLogic 9.2 or earlier, extract the `rpi.war` file to an RPI folder under the Model N directories., such as `C:\modeln\rpi`.
1. Click **Lock & Edit**.
2. Click **Deployments**.
3. If this is a re-deployment, remove the existing RPI application by selecting it and clicking **Delete**.
4. Click **Install**.
5. Browse to the `rpi.war` file (or directory where you extracted it for earlier WebLogic versions).
6. Click **Next**.
7. Click **Install this deployment as an application**.

8. Click **Next**.
9. Select the managed server on which you would like to deploy the application.
10. Click **Next**.
11. Enter a deployment a name.
12. Accept the default settings.
13. Click **Next**.
14. Click **Finish**.
15. Click **Activate Changes**.

10.5 How to Deploy the RPI WAR File for WebSphere

10.5.1 How to Install the RPI WAR File for WebSphere

To install a new enterprise application, navigate to **Applications > Install New Application** in the administration console. This initiates the application installation wizard. A series of configuration screens are presented to gather information about various aspects of the application.

To install the RPI WAR file:

1. In the Preparing for the application installation screen, under Specify the EAR, WAR, JAR, or SAR module to upload and install, select the radio button that contains the location of the RPI WAR file - either the local file system or the remote file system.
2. Enter the location of the WAR file to be deployed. You can do this by browsing to it and selecting it.
3. Specify the context root as `rpi` for the RPI application.
4. Select the Prompt me only when additional information is required radio button.
5. Click **Next**.

   The **Install New Application** wizard opens.
6. For *Step 1: Select installation options:*
   a. Select **Pre-compile JavaServer Pages files**.
   b. Leave the **Directory to Install Application** field empty.
   c. Make sure that **Distribute Application** is selected.
   d. Make sure that **Use Binary Configuration** is not selected.
   e. Make sure that **Deploy enterprise beans** is not selected.
   f. Enter <RPIAppName>App in the **Application Name** field if needed.
   g. Make sure that **Create MBeans for Resources** is selected.
   h. Make sure that **Enable Class Reloading** is not selected.

   i. Leave the **Reload Interval in Seconds** field empty.
   j. Make sure that **Deploy Web services** is not selected.
   k. Leave the other fields as their default.
   l. Click **Next**.

7. For *Step 2: Map modules to servers:*
   a. In the Clusters and Servers: field, select the appropriate web server and application server to which you would like to deploy the RPI application. To select them both at the same time, press the CTRL key while selecting.
   b. Check the appropriate WAR file.
   c. Verify that the correct server name displays next to the WAR file.
   d. Click **Apply**.
   e. Click **Next**.

8. For *Step 3: Map resource references to resources:*

   The RPI application requires a connection to two data sources. The Model N data source is used to generate reports against the Model N data. The RPI data source is used to store RPI configuration and report information. The following steps are used to configure the RPI WAR to connect to both data sources.

   a. Under Specify authentication method:, select **Use default method (many-to-one mapping)**.
   b. From the drop-down list, select the appropriate node with the appropriate <RPIAppName>Auth authentication type.
   c. The table at the bottom of the page displays two data source entries. Select the entry associated with the RPI data source. This is identified by the “jdbc/mn” entry in the Resource Reference column of the table.
   d. Click **Apply**. The Login configuration information is populated.
   e. From the drop-down list, select the appropriate node with the appropriate <AppName>Auth authentication type.
   f. The table at the bottom of the page displays two data source entries. Select the entry associated with the Model N data source. This is identified by the `jdbc/di` entry in the Resource Reference column of the table.
   g. Click **Apply**. The Login configuration information is populated.
   h. Click **Next**.

9. For *Step 4: Map virtual hosts for Web modules:*
   a. Select the Model N WAR file.
   b. Select the virtual host from the drop-down.
   c. Click **Next**.

10. For *Step 5: Summary, perform the following actions:*
    a. Verify that the settings are correct.
    b. Click **Finish**.

11. Once the message Application <RPIAppName>App installed successfully. is displayed, click the Save directly to the master configuration. link to commit the application deployment. The Enterprise Applications screen is displayed.

10.5.2 How to Create a JDBC Provider for RPI with WebSphere

1. Log into the WebSphere administration condole
2. Under Authentication, click **Java Authentication and Authorization Service > J2C authentication data**.
3. Click **New**.
4. Fill in the following configuration properties:
    - **Alias:** `<RPI>Auth`
    - **User ID:** `<RPIDBUser>`
    - **Password:** `<RPIDBPass>`
    - **Description:** RPI database
5. Click **OK**.
6. If the JDBC data source Oracle Non JTA Non XA has not been created yet, see Create a New JDBC Provider on page 113, then continue.
7. If the JDBC data source Oracle Non JTA Non XA has not been created yet, see Create a New JDBC Provider on page 113, then continue.
8. Under Resources, click **JDBC > Data Sources**.
9. From the drop-down list, select the server scope that corresponds to the `<AppName>` server.
10. Click **New**.
11. Fill in the following data source information fields:
    - **Data source name:** `<RPIAppName>OraDataSource`
    - **JNDI name:** `jdbc/<RPIAppName>OraDataSource`
12. Select the `<RPIAppName>Auth J2C` authentication alias from the drop-down list.
13. Click **Next**.
14. Enable "Select an existing JDBC provider".
15. Select the "Oracle Non JTA Non XA" JDBC provider defined earlier.
16. Click **Next**.
17. Specify the properties for the data source.
    - **URL:** `jdbc:oracle:thin:@<DBHost>:<TNSPort>:<DBInstance>`
    - **Data store helper class name:** Oracle 9i and prior data store helper.
    - Select “Use this data source in container managed persistence (CMP)” from the drop-down list.
18. Click **Next**.

19. Review the settings and click **Finish**.

11 Installing & Configuring Performance Analytics

The Model N Performance Analytics application uses QlikTech's QlikView as the visualization layer. The QlikView deployment consists of the QlikView Server and QlikView Publisher. The QlikView Server displays the analytics while QlikView Publisher performs the data load from the data mart to the QlikView Server.

11.1 Prerequisites

The following items are required for the QlikView Server:

- Model N system
- Windows 2003 or 2008 Server
- A user on the Windows Server with administrative privileges, under which all QlikView services will run.
- Microsoft .Net 3.5 Framework
- Oracle client installed
- Data mart created

11.1.1 NTLM Authentication

QlikView uses NTLM authentication to authenticate the Model N application with the QlikVew server. Model N can accept either NTLMv1 and NTLMv2 responses.

Important: Check with the your network administrator to confirm if users can be authenticated with the local domain. If not, create the QlikView user in the network domain.

11.2 QlikView Server Installation and Deployment

This section covers:

- creation of the data mart
- installation of the QlikView Server and Publisher
- installation of the Performance Analytics file
- creation of data sources
- configuration of the QlikView Server

11.2.1 How to Create the Data Mart

1. Unzip the `tools-analytics.jar` file.
2. Update the `bin/properties.sh` on UNIX or `bin\properties.bat` on Windows with the following values:

```
MN_PA_DB_USER=<username for the data mart schema>
MN_PA_DB_PASSWORD=<password for the data mart schema>
MN_PA_DB_NAME=<SID of the data mart schema>

MN_PA_DB_SYS_USER=<system username for the data mart database>
MN_PA_DB_SYS_PASSWORD=<system password for the data mart database>

MN_SRC_DB_USER=<username for the Model N schema>
MN_SRC_DB_PASSWORD=<password for the Model N schema>
MN_SRC_DB_NAME=<SID of the Model N schema>

MN_PA_DATA_START_DATE=<<Cut off date from which you need to bring
   data from source system to data mart during ETL process>>
MN_PA_BASE_CURR=<<Base currency for Data mart>>
```

3. In the `bin/ConfigOptions.csv`, update setup options for the ETL. After the initial deployment, all of the setup options are populated from the `ConfigOptions.csv` file. A number of these options are also available as application switches from the Model N user interface. When full and incremental loads are run, these options are updated in the data mart with the values from the

application switches. If `reloadconfig.sh` is run, these values will also be overwritten by the `ConfigOptions.csv` file.

Note: Base currency and Data start date reside in Properties.sh/bat file.

Important: The first four options must be set to get the correct ETL results.

Table 11-1: ETL Setup Options

Option name	Available as application switch	Applicable values (bold is default)	Description
Base currency	Y	**SKIP**/LOAD	This is a base currency
Data start date	Y	**SKIP**/LOAD	Only data after the start date will be extracted to the data mart. All other data will be ignored.
CURRENCY_TYPE	N	**Currency Rate Type**	This value must be set to a valid exchange rate type
SYSTEM_TIMEZONE	N	**GMT**(any valid time zone).	This is the time zone information. It must be set to the Application Server time zone, otherwise some of the data might not get pushed to the data mart for some time.
DIRECT_SALE	Y	**LOAD**/SKIP	Load direct sales data
INDIRECT_SALE	Y	**LOAD**/SKIP	Load indirect sales data
CUSTOM_SALE	Y	**LOAD**/SKIP	Load direct sales data

Table 11-1: ETL Setup Options (Continued)

Option name	Available as application switch	Applicable values (bold is default)	Description
MCD_DATA	Y	**LOAD**/SKIP	Load managed care data
MCO_DATA	Y	**LOAD**/SKIP	Load medicaid data
EDI852_SALE	N	**LOAD**/SKIP	Load EDI 852 sales data
REBATE_DATA	N	**LOAD**/SKIP	Load rebates data
CUSTOMER_PRICE	N	**SKIP**/LOAD	Create price points for each customer. This might create a sizable data and require up to several hours, based on the data set size
CUSTOMER_PF	N	**SKIP**/LOAD	Create price floors for each customer. This might create a sizable data and require up to several hours, based on the data set size
FG_DATA	N	**LOAD**/SKIP	Create Off invoice FG data
OA_DATA	N	**LOAD**/SKIP	Do not enable this step. This will create optional consolidated data for direct sales.
EXTERNAL_DATA	N	**LOAD**/SKIP	Load data from external data source

Table 11-1: ETL Setup Options (Continued)

Option name	Available as application switch	Applicable values (bold is default)	Description
FACT_EXTRACT_DELAY	N	**5**	This is a delay between start of the ETL and fact extraction Time in minutes to avoid submissions being missed by the ETL due to the long running commits. Default value is sufficient for all installations.
TRACE	N	**DISABLED**/ ENABLED	Trace. Can be enabled to generate Oracle trace for the ETL.
UNITTEST_DIM	N	**ENABLED**/ DISABLED	Flag to run unit tests for dimensions, don't change this.
UNITTEST_FACT	N	**ENABLED**/ DISABLED	Flag to run unit tests for facts, don't change this.
UNITTEST_EXT_DIM	N	**ENABLED**/ DISABLED	Flag to run unit tests for external dimensions, don't change this.
UNITTEST_EXT_FACT	N	**ENABLED**/ DISABLED	Flag to run unit tests for external facts, don't change this.

Table 11-1: ETL Setup Options (Continued)

Option name	Available as application switch	Applicable values (bold is default)	Description
RAISE_COT_ERROR	N	**DISABLED**/ ENABLED	Raise duplicate class of trade exception. If the ETL determines more than one COT assigned to a customer for the same time period it will pick one. You can enable exception to be raised in this case.

Set values to "LOAD" if the corresponding data is present in the Model N schema or "SKIP" if the data is not present.

4. Run `bin/setup.sh` on UNIX or `bin\setup.bat` on Windows to create data mart set up tables and load the configuration values in the data mart.

   A full refresh of the data mart must occur before the data mart can be used.

5. Run `bin/run.sh -c` on UNIX or `bin\runetl.bat -c` on Windows to refresh the data mart. The data mart must be refreshed before QlikView Server and Publisher are configured.

11.2.2 How to Install QlikView Server and Publisher

To set up QlikView Server and Publisher for Performance Analytics:

1. Create a new Windows administrator level user to be used to run the QlikView services.
   a. Select the Computer Management application from the Administrative Tools options under the **Start** menu.
   b. Select the **Users** folder under the Local Users and Groups item in the left panel.
   c. Select the **New User** option under the **Action** menu.
   d. Enter the user name in the **User name** field.
   e. Enter the password in the **Password** field.
   f. Reenter the password in the **Confirm Password** field.
   g. Make sure that the "User must change password" and "Account is disabled" are not selected.
   h. Click **Create**.

2. Download the QlikView Server and Publisher distributable file from the customer portal. Both QlikView Server and Publisher are included in the same installer.

Note: The QlikView Server and Publisher require 64-bit Windows.

---

3. Run the file `QvsSetupRedist.exe` to start the installer.
   The Welcome screen opens.
4. Complete the installation wizard to fit your installation, with the following required values:
   - On the Destination Folder and Installation and Customization screen, clear the **Install Typical components** check box.
   - On the Features screen, select **Microsoft IIS support**.
   - On the Logon Information screen, browse to select the newly created Windows user, and enter the password and logon information.

11.2.2.1 Starting QlikView Services

1. Go to Windows **Computer Management > Services** and verify that all QlikView services (there are six) have been started.
2. Add the newly created Windows user to the QlikView Administrator group.
   a. Start the Computer Management application from the Administrative Tools options under the Start menu.
   b. Select the User folder under the Local Users and Groups item in the left panel
   c. Select the newly created user.
   d. Select the **Properties** option under the Action menu.
   e. Select the **Member Of** tab.
   f. Click **Add**.
   g. Ensure that **Select this object type** is set to `Groups` and **From this location** is set to the local server.
   h. Enter `QlikView Administrators` in the Enter the object names to select field.
   i. Click **OK** to close the dialog box.
   j. Click **OK** to add the user as a member of the QlikView Administrators group.

11.2.3 How to Install the Performance Analytics File

To install the Performance Analytics file on the QlikView server:

1. Unzip the Model N `tools.jar` file.
2. To configure the analytics installation tool, run:

```
ant -f ops_analytics.xml configure
```

   Java and Ant must be installed on the QlikView server to run this tool.

3. Enter the file name of the performance analytics file.
   The default file name is `mn_analytics`.
4. Enter the full path name of the directory where the QlikView Documents are stored.
   The default location is `C:\Documents and Settings\All Users\Application Data\QlikTech\Documents`.
5. Enter the full path name of the QlikView Program files directory.
   The default location is `C:\Program Files\QlikView`.
6. Enter the full path name of the directory which contains the unzipped `tool-analytics.jar` file.
7. To install the files into the correct locations, run:

```
ant -f ops_analytics.xml setup
```

11.2.4 How to Configure QlikView

1. Start the QlikView Enterprise Management Console.
2. Go to **System** tab > **License** tab.
3. Under QlikView Server & Publisher:
   a. For the QlikView Server, enter the serial number and control.
      Use the QlikView Server License only; do not use license keys from any other QlikView product.
   b. Click **Apply**.
   c. For QlikView Publisher, enter the serial number and control.
      Use the QlikView Publisher License only; do not use license keys from any other QlikView product.
   d. Click **Apply**.
4. Go to **System** tab > **Setup** tab.
5. Expand the QlikView Servers folder and select the server.
6. Select the **Folders** tab and make note of the **Root Folder** value.
7. Select the **Security** tab.
   a. Select the **Allow anonymous** option button for Clients group.
   b. Select the **On local computer** option button for Anonymous Account group.
   c. Select **DMS authorization** option button for Authorization group.
   d. Click **Apply**.
8. Expand the QlikView Web Servers entry and select the web server.
9. Select the **Access Point** tab.
   a. Select the **Never** option button for the Authentication group.
   b. Select the **AJAX zero footprint** option button for the Default Preferred Client group.
   c. Click **Apply**.

   d. Select the **AJAX** tab.
   e. Select the **Always Anonymous** check box.
   f. Click **Apply**.

10. Expand Distribution Services and select the server.
11. Select the **General** tab.
    a. Enter the Root Folder value from the QlikView Servers Folders tab into the **Path** field.
    b. Click **Apply**.
12. Select **Documents** tab.
13. Select the **User Documents** tab.
14. Expand the server entry.
    a. Select the QlikView Document file.
    b. Ensure that the **AJAX** check box is selected under AccessPoint access methods.
    c. Click **Apply**.
    d. Select the **Authorization** tab.
    e. Select Add link.
    f. Select Change link.
    g. Select the Anonymous check box.
    h. Leave the Always radio button selected in the Access group.
    i. Click **OK.**
    j. Click **Apply**.
15. Change the QlikView web server timeout.
    a. Go to the directory where the QlikView application files are stored. By default it is `C:\Documents and Settings\All Users\Application Data\QlikTech`.
    b. Open the file `config.xml` present in the `\QvWebServer` folder.
    c. Change the `<QvsTimeout>` value to 300.

11.2.5 How to Create a System Data Source

1. Open **Data Sources (ODBC)** from the Windows Control panel.
2. Select the **System DSN** tab.
3. Click **Add**.
4. Select Oracle in <oracle home>.
5. Click **Finish**.
6. Enter a data source name.
7. Enter the tns service name.

8. Enter the user ID of the data mart schema.
9. Click **Test Connection**.
10. Enter the password for the data mart schema (the service name and user name should be pre-populated).
11. Click OK to test the connection.
12. Click OK to add the data source.

11.2.6 How to Add a Full Reload Publisher Task

1. In the QlikView Enterprise Management Console, go to **Documents > Source Documents > QDS@*servername* > *doc folder* > mn_analytics.qvw.**
2. Select + to add a new task.
3. Enter `full_reload_mn_analytics` in the task name field.
4. Leave the **Enabled** check box selected.
5. Select the **Reload** tab.
6. Leave the **Perform reload Enable** check box selected.
7. Enter `SERVER_PROP`S in the parameter name.
8. Enter `FULL,<DB_USERNAME>,<DB_PASSWORD>,<DATASOURCE_NAME>` in the parameter value field where:
   - `<DB_USERNAME>` and `<DB_PASSWORD>` are the user name and password of the data mart.
   - `<DATASOURCE_NAME>` is the data source name defined earlier.
9. Select the **Triggers** tab.
10. Click the plus (+) icon to add a trigger. The Configure trigger dialog box opens.
11. Under the Start the task field, select the **On an external event** option.
12. Verify that the **Enabled** check box is selected.
13. Enter `mnanalytics` in the Password field.
14. Click **Apply**.
15. To test the full refresh trigger, go to **Status > Tasks > QDS@<servername> > Default > full_reload_mn_analytics** and click the run button (the run button is the black triangle).

Important: The full_reload_mn_analytics trigger must be run after the .qvw file is installed so that the correct data is loaded into the file. After the refresh has completed, the QlikView Server Windows Service must be restarted.

11.2.7 How to Add an Incremental Reload Publisher Task

1. In the QlikView Enterprise Management Console, go to **Documents > Source Documents > QDS@*servername* > *doc folder* > mn_analytics.qvw**.

2. Select + to add a new task.
3. Enter `incr_reload_mn_analytics` in the task name field.
4. Leave the **Enabled** check box selected.
5. Select the **Reload** tab.
6. Leave the **Perform reload Enable** check box selected.
7. Enter `SERVER_PROP`S in the parameter name.
8. Enter `INCR,<DB_USERNAME>,<DB_PASSWORD>,<DATASOURCE_NAME>` in the parameter value field where:
   - `<DB_USERNAME>` and `<DB_PASSWORD>` are the user name and password of the data mart.
   - `<DATASOURCE_NAME>` is the data source name defined earlier.
9. Select the **Triggers** tab.
10. Click the + icon to add a trigger. The Configure trigger dialog box opens.
11. Under the Start the task field, select the **On an external event** option.
12. Verify that the **Enabled** check box is selected.
13. Enter `mnanalytics` in the Password field.
14. Click **Apply**.

Note: A full refresh needs to run before an incremental refresh runs or the data will not be available.

11.3 How to Configure the Model N System for Performance Analytics

To configure the Model N system for Performance Analytics, add the following information to the `local_common.properties` file.

Code 11-1: Properties to Configure Model N for QlikView

```
com.modeln.AppSwitch.analytics.performance.qv.host=<hostname of
   QlikView Server>
com.modeln.AppSwitch.analytics.performance.qv.mn_analytics.document=<na
   me of the qlikview file including the qvw extension; default:
   mn_analytics.qvw>
com.modeln.AppSwitch.analytics.performance.qv.server.username=<username
   of user in the QlikView Administrators group>
com.modeln.AppSwitch.analytics.performance.qv.server.password=<password
   of user>
com.modeln.AppSwitch.analytics.performance.qv.server.domain=<domain of
   user>
com.modeln.AppSwitch.analytics.performance.qv.mn_analytics.publisher.fu
   ll.triggertask=<name of the trigger used to perform a full refresh
   of the qlikview document; default: full_reload_mn_analytics >
```

Code 11-1: Properties to Configure Model N for QlikView (Continued)

```
com.modeln.AppSwitch.analytics.performance.qv.mn_analytics.publisher.fu
   ll.triggerpassword=<password for the full refresh trigger task;
   default: mnanalytics >
com.modeln.AppSwitch.analytics.performance.qv.mn_analytics.publisher.in
   cr.triggertask=<name of the trigger used to perform an incremental
   refresh of the qlikview document; default: incr_reload_mn_analytics
   >
com.modeln.AppSwitch.analytics.performance.qv.mn_analytics.publisher.in
   cr.triggerpassword=<password for the incremental refresh trigger
   task; default: mnanalytics >
```

11.3.1 How to Add Additional QlikView Documents

The Model N server is configured to use one QlikView document in each deployment. To add additional QlikView documents and make them available on the Analytics page, add the following set of properties to the `local_<appserver>.properties` file for each additional QlikView document, then restart the server.

Code 11-2: Properties for Additional QlikViewDocuments

```
com.modeln.AppSwitch.analytics.performance.qv.files+=<Document
Identifier>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.document=<file name of the QlikView document including the
.qvw extension>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.displayName=<title used for display on the Performance
Analytics page>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.publisher.full.triggertask=<name of the full refresh
trigger task defined in QlikView Publisher>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.publisher.full.triggerpassword=<password of the full
refresh trigger task defined in QlikView Publisher>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.publisher.incr.triggertask=<name of the incremental refresh
trigger task defined in QlikView Publisher>
com.modeln.AppSwitch.analytics.performance.qv.<Document
Identifier>.publisher.incr.triggerpassword=<password of the incremental
refresh trigger task defined in QlikView Publisher>
```

Note: Model N supports connecting to one datamart instance and one QlikView Server and Publisher.

11.3.2 How to Create the Data Source

You will need to create a data source connection from you application server to your analytics database.

11.3.2.1 WebSphere

For WebSphere, add a JDBC connection to the ETL database with the following information:

```
com.modeln.etl.jndiUrl=${com.modeln._default.jndiUrl}
com.modeln.etl.jtsDataSourceName=java:comp/env/jdbc/
    mn54phaOraDataSource
com.modeln.dataSources=_default
com.modeln.nonTxDataSources=etl
```

To create the data source, follow the steps of Create a JDBC Provider in the chapter Deploying with IBM WebSphere with `mn54aphOraDataSource` as the data source name and set custom properties `nonTransactionalDataSource=true`.

11.3.2.2 WebLogic

For WebLogic, add a JDBC connection to the ETL database with the following information:

```
com.modeln.etl.jdbcUrl=jdbc:weblogic:jts:mn54phaOraDataSource
com.modeln.etl.jndiUrl=${com.modeln._default.jndiUrl}
com.modeln.etl.jtsDataSourceName=mn54phaOraDataSource
com.modeln.dataSources=_default
com.modeln.nonTxDataSources=etl
```

To create the data source, follow steps 27-41 of How to Configure the Environment in the chapter Deploying with Oracle WebLogic Server with mn54aphOraDataSource as the data source name and set custom properties `nonTransactionalDataSource=true`.

11.4 AJAX and QlickView’s Internet Explorer Plug-in

Model N is configured to use AJAX pages to display performance analytics with QlikView. AJAX has the advantage of using the standard http port and does not require any extra installation on the client computer. Model N can be configured to use QlikView's Internet Explorer plug-in instead. The plug-in provides a richer user experience. but requires an open port (port 4747 is used by default) and a separate installation of the plug-in on each client computer.

To install the plug-in on the client computer, copy the `QvPluginSetup.exe` installation program from the `C:\Program Files\QlikView\Server\QvClients\QvPlugin` on the QlikView server computer to the client. Run the install program on the client and follow the instructions presented.

To configure the Model N server to use the plug-in, set the following property in the property file:

```
com.modeln.AppSwitch.analytics.performance.qv.server.qvwurl=${com.model
n.AppSwitch.analytics.performance.qv.server.pluginurl}
```

and restart the Model N application.

To set the Model N server back to using the AJAX pages, set the following property in the property file:

```
com.modeln.AppSwitch.analytics.performance.qv.server.qvwurl=${com.model
n.AppSwitch.analytics.performance.qv.server.ajaxurl}
```

and restart the Model N application.

Note: The behavior of the mouse scroll wheel is different with plug-in than with AJAX pages. With the plug-in, the scroll wheel will not work in charts that do not have scroll bars.

11.5 Installation and Configuration of a Clustered QlikView Server Environment

All clustered QlikView environment requires the QlikView Enterprise Management Console. QlikView Server (QVS) supports load sharing of documents across multiple physical or logical computers. This sharing includes the ability to share, in real time, information about collaboration objects, automated document loading and unloading (through DMS), and user license CALs. Special licensing is available to enable multiple server instances to share the same license number.

This license must have the following values:

```
NUMBER_OF_CLUSTER_NODES;2;;
NUMBER_OF_SESSIONCALS;30;;
```

To utilize load sharing between multiple QlikView Servers, all document and support files must be shared between the servers. In other words, all servers must point to the same physical location for the files. In addition to the standard `.qvw` QlickView file types, QlikView Server creates and maintains additional files to store load sharing data. These files have a file type extension of `.pgo` (Persistent Group Object) and are located in the same folder as the QlikView documents. Operating System Load Balance or Failover configurations are external to the QlikView Server load sharing configuration, and QlikView Server has no control over those systems.

Server configuration settings are shared between all clustered QlickView Servers and can be maintained through the QlikView Enterprise Management Console connected to any of the clustered servers. Performance of a particular QVS system can be monitored through the Management Console by connecting to that system. How the load balancing is made (that is, which QlikView Server the client should be directed to) is set in the QlikView web server's configuration file.

Since DMS data is shared among the QV Servers, any automated document load/unload procedures are performed on all Servers. DMS Authorization is also shared among all clustered QVS.

11.5.1 How to Set Up a Cluster

This procedure covers how to set up a cluster of two servers. For clarification purposes, the servers are referred to as SERVER1 and SERVER2 in the documentation.

11.5.1.1 Prerequisites

In addition to the prerequisites for a standalone environment (see Prerequisites on page 169 for more information), a clustered environment also needs:

- two servers that both belong to the same domain where SERVER1 is the domain controller.

- both servers must share a common disk space where QlikView Documents will be located. This could be done with a SAN or a shared portion of disk that is referable by both servers.

11.5.1.2 How to Set Up the Cluster

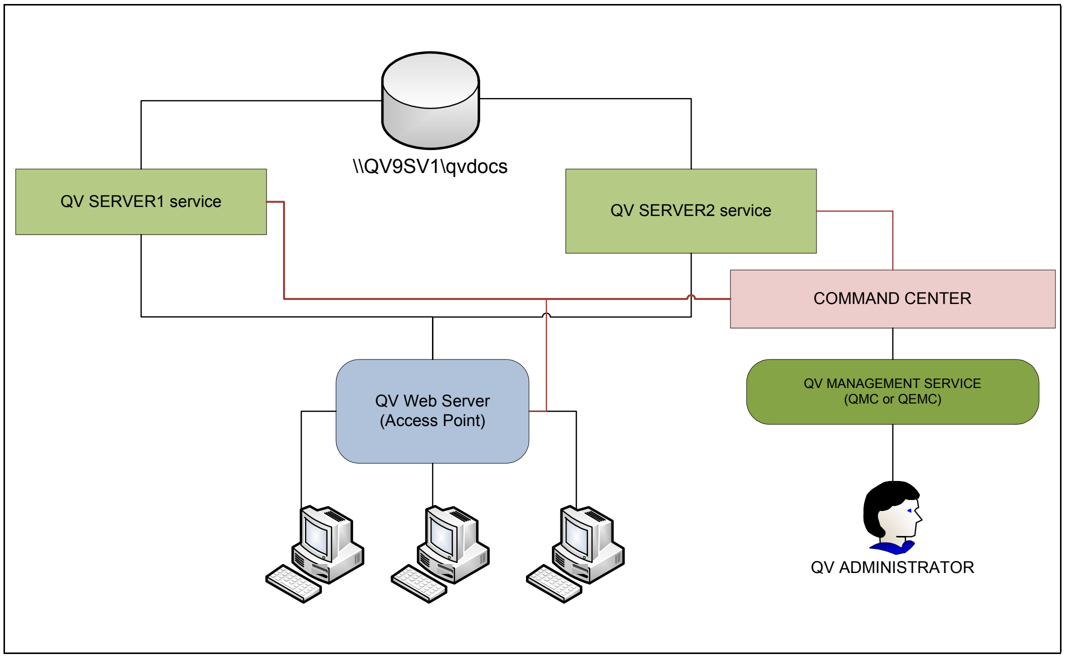
The servers should have the following items installed on them by the end of this procedure.

Table 11-2: QlikView Clustered Environment Installation Items

SERVER1	SERVER2
QvServer	QvServer
QvWebServer	QvWebServer
Qv Management Service	Qv Management Service
Qv Publisher Command Center	
Qv Distribution Service	
Qv Directory Service Connector	

The following diagram provides a visual layout of the configuration.

Figure 11-1: Clustered QlikView Environment



To setup a clustered environment:

1. Perform a complete installation of QV9SV1 on SERVER1, including QlikView Server, QlikView Distribution Service and QlikView Management Services of the cluster.

   Note: The account running the Management Service must be a member of the QlikView Administrators group and a member of the local Administrators group on each "slave" QlikView Server computers in order to restart all QlikView Servers from the QlikView Enterprise Management Console (QEMC).

   ---

2. License the installation.
   a. Start QEMC (http://localhost:4780/qemc/SystemSetup.htm#) on MASTER.
   b. Go to **System > Licenses** and enter the license information, then click **Apply**.
3. Within QlikView Enterprise Management Console (QEMC), go to **System > Setup > QlikView Servers > General** and enter the service name that will be used for cluster service, then click **Apply**.
4. Go to **System > Setup > QlikView Servers > Folders** and enter a shared drive for the documents that is accessible from both nodes, then click **Apply**.
5. Install QlikView Server on SERVER2, including QVS, Server and Management Services.
6. In QEMC on SERVER1, go to **System > Setup > QlikView Servers > Cluster**.
   a. Add the URL for the second cluster. You should then have two servers listed.
   b. Enter the Control Number, then click **Apply**. A confirmation dialog box regarding licenses opens.
7. Go to **System > Setup > QlikView Web Servers > Access Point** and verify that your cluster name is selected under Server Connections.
8. Verify that everything is running by:
   - Going to **Status > Services**.
   - Going to http://localhost/QvAjaxZfc/QvsStatus.aspx to see that both nodes are listed.
9. Go to **Status > QVS Statistics** to see the usage for both servers.

11.5.2 How to Test the Installation

Load balancing only works in a clustered environment if you go to QlikView AccessPoint through `http://SERVER1/qlikview`, where SERVER1 is name of the domain-controlling server. Using a direct URL for AccessPoint will only take you to one server.

To test the clustered installation:

1. Go to QlikView AccessPoint: `http://SERVER1/qlikview`.
2. Click on a document and verify which server received the request by checking **Status > QVS Statistics**.
3. Shut down that server.

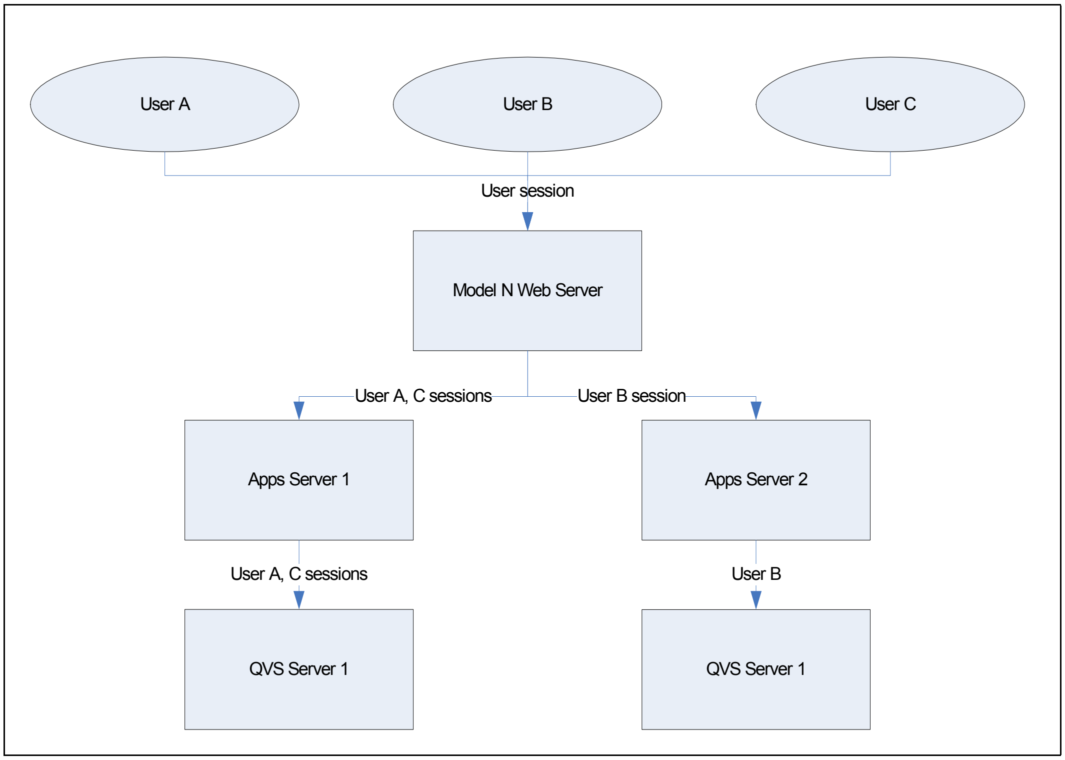
4. In AccessPoint, click on the selected document again. A connection lost message should be displayed and you will be redirected to the other server. Verify this on the **Status > QVS Statistics** page.
5. Shut down the second server and bring up the first. You should be redirected again.
6. Bring up the second server and click on different documents to verify that you are randomly directed to different servers for the different documents.

11.5.3 How to Set Up Model N for a Clustered QlikView Environment

A current design limitation on the QlikView Server side does not allow using failover and loading balancing directly in the clustered setup.

In the current implementation, each Model N application server should be connected to a QlikView cluster node directly and the load balancing will be done by Model N web server.

Figure 11-2: Model NApplication Server in a QlikView Clustered Environment



When a user accesses the Model N user interface, the web server assigns a session to a particular application server. The application server is configured (usingthe `analytics.properties` property file) to connect to one node of the QVS server and the user is always directed to this node when Performance Analytics is accessed from the given application server.

If the QVS server goes down, all users on the attached application server will be affected and will not be able to access Performance Analytics.

12 Installation Verification

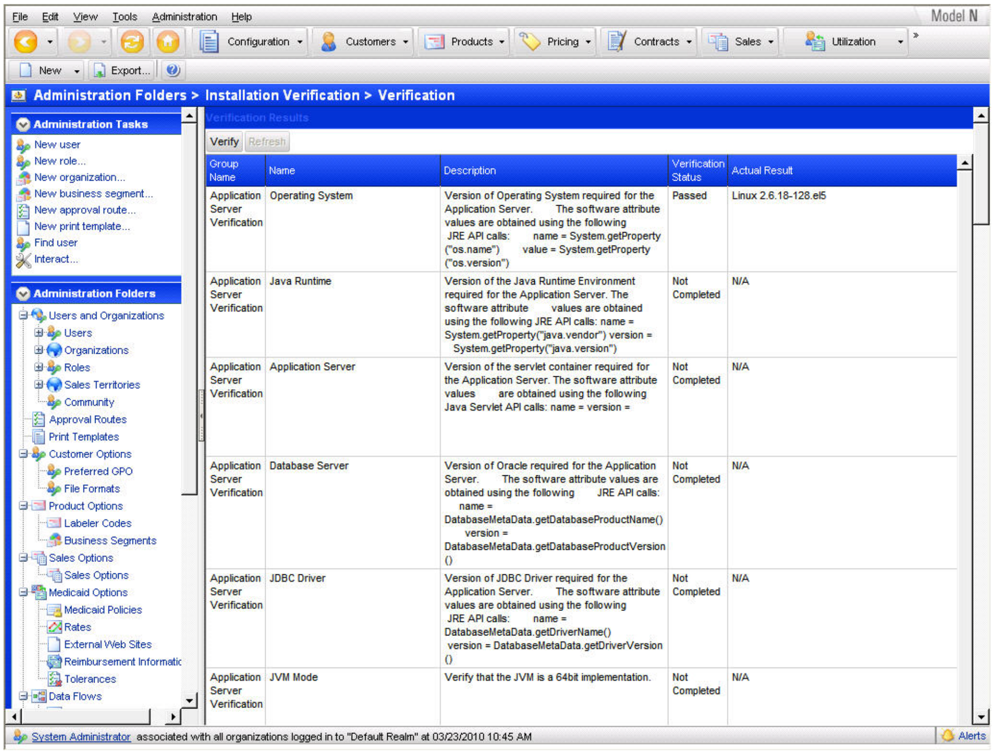
A full installation of the Model N application can have many software components including a web server, an application server, a reporting server, a database server, and an analytics server. IT administrators usually need to verify component versions and configuration settings to ensure a proper installation, but because the Model N application is complex, this can be difficult.

In release 5.4.1, an automated installation verification process that checks the software component versions and configurations was added to make it easier to verify that the system is properly set up. The installation verification information can be collected and included with support tickets to help diagnose issues.

12.1 Installation Verification Page

Navigation Path: **Administration > Installation Verification > Verification**

Figure 12-1: Installation Verification Page



Group Name	Name	Description	Verification Status	Actual Result
Application Server Verification	Operating System	Version of Operating System required for the Application Server. The software attribute values are obtained using the following JRE API calls: name = System.getProperty ("os.name") value = System.getProperty ("os.version")	Passed	Linux 2.6.18-128.el5
Application Server Verification	Java Runtime	Version of the Java Runtime Environment required for the Application Server. The software attribute values are obtained using the following JRE API calls: name = System.getProperty("java.vendor") version = System.getProperty("java.version")	Not Completed	N/A
Application Server Verification	Application Server	Version of the servlet container required for the Application Server. The software attribute values are obtained using the following Java Servlet API calls: name = version =	Not Completed	N/A
Application Server Verification	Database Server	Version of Oracle required for the Application Server. The software attribute values are obtained using the following JRE API calls: name = DatabaseMetaData.getDatabaseProductName() version = DatabaseMetaData.getDatabaseProductVersion ()	Not Completed	N/A
Application Server Verification	JDBC Driver	Version of JDBC Driver required for the Application Server. The software attribute values are obtained using the following JRE API calls: name = DatabaseMetaData.getDriverName() version = DatabaseMetaData.getDriverVersion ()	Not Completed	N/A
Application Server Verification	JVM Mode	Verify that the JVM is a 64bit implementation.	Not Completed	N/A

System Administrator associated with all organizations logged in to "Default Realm" at 03/23/2010 10:45 AM

The Installation Verification page provides the details of the Model N installation in the form of the following information:

- Group Name
- Name
- Description
- Verification Status
- Actual Result
- Expected Result
- Result Log
- Node Name
- Run ID
- Build Version

12.2 How to Get Installation Verification Results

12.2.1 How to Verify Results in the Model N Interface

To get your installation verification information:

1. Go to the Installation Verification page: **Administration > Installation Verification > Verification**
2. Click **Verify**.
3. Click **Refresh**. Depending on performance, you may need to repeat this steps until information displays.

   The verification results are shown.

Note: To view all columns, you need to scroll to the bottom of the page to access the left-right scroll bar.

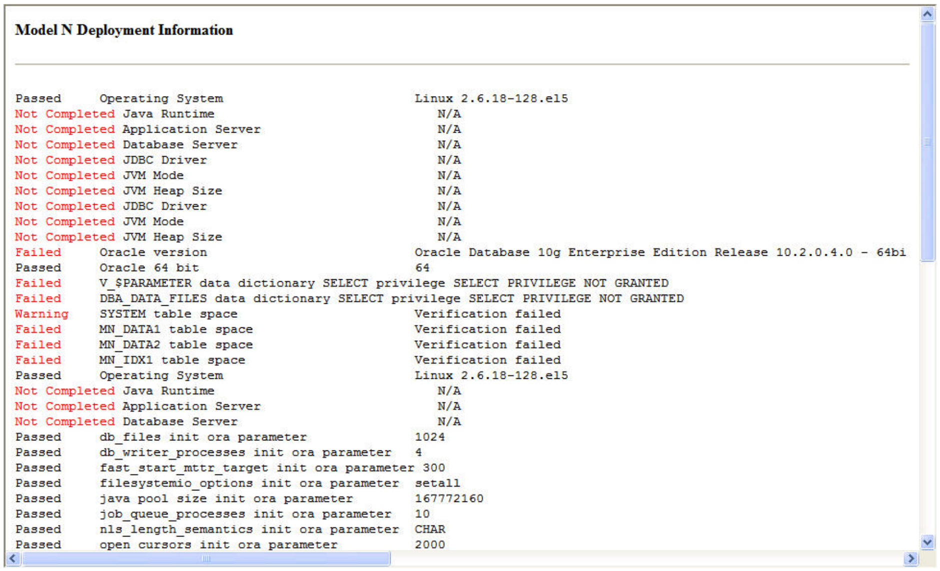
12.2.2 How to Verify Results from JSP Pages

To access results from .jsp pages:

- Go to `http://mnserver/Ops?cmd=verify&act=view&format=text`, where *mnserver* is the Model N system.

Depending on your deployment, the results page looks similar to the following image.

Figure 12-2: Sample Installation Verification JSP Results



Model N Deployment Information

```
Passed     Operating System                         Linux 2.6.18-128.el5
Not Completed Java Runtime                             N/A
Not Completed Application Server                       N/A
Not Completed Database Server                          N/A
Not Completed JDBC Driver                              N/A
Not Completed JVM Mode                                 N/A
Not Completed JVM Heap Size                            N/A
Not Completed JDBC Driver                              N/A
Not Completed JVM Mode                                 N/A
Not Completed JVM Heap Size                            N/A
Failed     Oracle version                           Oracle Database 10g Enterprise Edition Release 10.2.0.4.0 - 64bi
Passed     Oracle 64 bit                            64
Failed     V_$PARAMETER data dictionary SELECT privilege SELECT PRIVILEGE NOT GRANTED
Failed     DBA_DATA_FILES data dictionary SELECT privilege SELECT PRIVILEGE NOT GRANTED
Warning    SYSTEM table space                       Verification failed
Failed     MN_DATA1 table space                     Verification failed
Failed     MN_DATA2 table space                     Verification failed
Failed     MN_IDX1 table space                      Verification failed
Passed     Operating System                         Linux 2.6.18-128.el5
Not Completed Java Runtime                             N/A
Not Completed Application Server                       N/A
Not Completed Database Server                          N/A
Passed     db_files init ora parameter              1024
Passed     db_writer_processes init ora parameter   4
Passed     fast_start_mttr_target init ora parameter 300
Passed     filesystemio_options init ora parameter  setall
Passed     java pool size init ora parameter        167772160
Passed     job_queue_processes init ora parameter   10
Passed     nls_length_semantics init ora parameter  CHAR
Passed     open cursors init ora parameter          2000
```

The verification results contain the verification status, component name, and actual result value.

12.2.3 How to Verify Results from the Command Line

The command line interface lets Operations users and system administrators access to installation verification through the Ops tool (ops.xml). Validation can be executed as follows:

```
ant -f ops.xml VerifyEnv
```

A new verification run is triggered, and the output of that run is displayed in the following format:

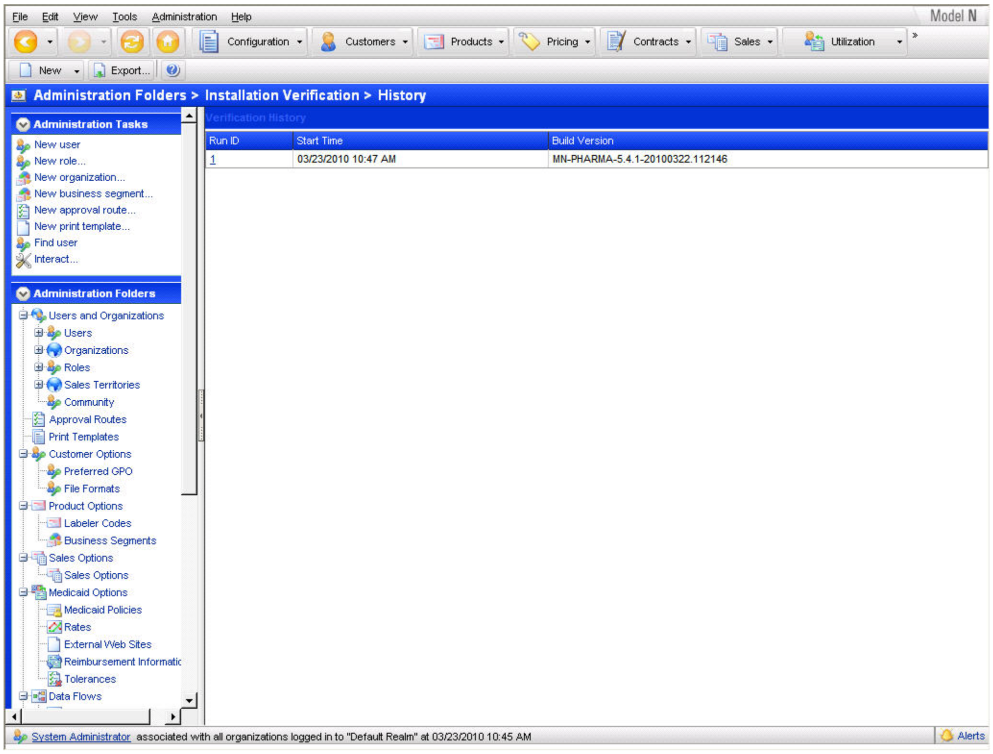
Code 12-1: Sample Installation Verification Command Line Results

```
SystemVerificationResult
========================
Group Name=Application Server Verification|Name=Operating System|Run
Id=5|Node=hdqpdverify8|
Result=The operating system Linux with version 2.6.18-128.el5 is valid.|
Expected=|windows: Microsoft Windows
  windows-2003: 2003
  windows-2008: 2008|solaris: Sun Solaris
  solaris-10: 10|linux: Linux
  linux-5.2: 2.6.18-92.el5
  linux-5.3: 2.6.18-128.el5|
Actual=Linux 2.6.18-128.el5|
Status=passed|
========================
SystemVerificationResult
========================
Group Name=Application Server Verification|Name=Java Runtime|Run
Id=5|Node=hdqpdverify8|
Result=IBM Corporation 1.6.0 is supported.|
Expected=|oracle-jrockit: Oracle JRockit
  oracle-jrockit-version: 1.6.0|sun-jdk: Sun Microsystems Inc.
  sun-jdk-version: 1.6.0_14|ibm-jdk: IBM Corporation
  ibm-jdk-version: 1.6.0|
Actual=IBM Corporation 1.6.0|
Status=passed|
```

12.3 Installation Verification History Page

Navigation Path: **Administration > Installation Verification > History**

Figure 12-3: Installation Verification History Page



Run ID	Start Time	Build Version
1	03/23/2010 10:47 AM	MN-PHARMA-5.4.1-20100322.112146

System Administrator associated with all organizations logged in to "Default Realm" at 03/23/2010 10:45 AM

The Installation Verification History pages contains a list of the verifications run. By clicking on the run ID, you can quickly review the verification information and compare the results between multiple runs to see if anything has changed.

Table 12-1: Installation Verification History Page

Name	Type	Description
Run ID	Link	An ID assigned to the installation verification. Clicking on this link displays the results in a separate Verification Results window.
Start Time	Date	The date and time the verification was run.
Build Version	String	The Model N application build version the verification was run against.

13 Migration

This chapter describes the process of migrating from one version of the Model N application suite to a later version.

Important: Before you begin, review the release notes for any changes to the supported configurations, any changed or added functionality, and the migration tasks to be implemented.

13.1 Migration Overview

The process of migration consists of:

- Comparing Releases
- Migrating Content
- Migrating Sales Validations to the Direct Loader Framework (if necessary)
- Rebates Migration (if necessary)
- Report Migration (if necessary)

Important: You must implement or terminate amended contracts before you begin the migration process.

13.2 Supported Migrations

Direct migration is supported for the following Model N application releases to 5.5.1:

- 5.4.1.x to 5.5.1
- 5.5 or 5.5 maintenance releases to 5.5.1

For all other migration scenarios, contact Model N Customer Support.

13.3 Migration Considerations

There are certain items to take into consideration if migrating to 5.3 or later releases.

13.3.1 Common Pre-migration Tasks for All Migration Paths

For all migration paths:

1. There must be no pending or failed timers in a migrated database. Ensure all the timers are completed prior to migration or delete the timers before you begin the migration.
2. Update the oracle-*.jar file.
   a. Download all oracle-*.jar files from 5.5.1 build directory.
   b. Reload them into Oracle with the following command, making sure ORACLE_HOME is set before you run it:

```
 loadjava -thin -resolve -verbose -user <dbuser>/
<dbpasswd>@<serverName>:<port#>:<ORACLE_SID>
<downloadDirectory>/oracle-platform.jar
```

13.4 Direct Migration from 5.4.1.x to 5.5.1

This section covers the direct migration from 5.4.1.x to 5.5.1.

13.4.1 Pre-migration Tasks

1. See Migration Considerations on page 194 before you begin.
2. Verify that `contentRoot` is set to:

```
com.modeln._default.contentRoot=<baseDir>/migration/
diff5.4.1to5.5.1content
```

3. For customer database migrations, exclude MN sample content by setting the following property:

```
com.modeln._default.contentSets-=sample
```

13.4.2 How to Migrate

1. Run the database upgrade script.
   Run the following command from 5.5.1 ANT environment:

```
ant -f ops.xml DBMigrateDirect541xTo551
```

2. Run the full populate command.

```
ant -f ops.xml RunFullPopulate
```

13.5 Direct Migration from 5.5 to 5.5.1

This section covers the direct migration from 5.5.x to 5.5.1.

13.5.1 Pre-migration Tasks

1. See Migration Considerations on page 194 before you begin.
2. Verify that `contentRoot` is set to:

```
com.modeln._default.contentRoot=<baseDir>/migration/
diff5.5to5.5.1content
```

3. For customer database migrations, exclude MN sample content by setting the following property:

```
com.modeln._default.contentSets-=sample
```

13.5.2 How to Migrate

1. Run the database upgrade script.
   Run the following command from 5.5.1 ANT environment:

```
ant -f ops.xml DBMigrateDirect55To551
```

2. Run the full populate command.

```
ant -f ops.xml RunFullPopulate
```

13.6 Comparing Releases

Before you begin any migration process, review the release notes to determine what has changed. If the current installation of your Model N application suite was customized, Model N Professional Services should help determine whether the changes within the release impact or replace you customizations.

13.7 Migrating Content

The migration of content is divided into the following main steps:

- Properties migration to the configuration directory structure (`MigratePropsToConf`).
- Database migration (`DBMigrate`), which initializes the environment and synchronizes the database schema to the structure expected by the application code. DBMigrate also executes the database migration tasks that cannot be handled by the other content populate tools, such as changing a column and data changes.
- Content population (`RunFullPopulate`) populates content files using Model N

importers and readers.

Tip: The sales migration tasks have dependencies on pharma-only and med dev-only FGOs. However, a single file contains all sales migration tasks in the base application. Tasks that are pharma-only (or med dev-only) are set up with `ignoreErrors=true`. If these tasks fail during a migration, the failure is logged but can be safely ignored. A failure occurs when a pharma-only task is executed in meddev or the other way around.

During migration, if running FullPopulateConfList do not include the CatalogFromXML and PMImplementPricesPopulate data flows. To remove data flows from configurations

1. Add the following data flow configuration XML file to <feature>/conf/fgo/Dataflow.xml:

Code 13-1: Data Flow Configuration

```
<fgo name="DataFlow">
<property name="FullPopulateConfList">
<values>
<!-- not required for Full populate -->
<value importAction="remove"><![CDATA[CatalogFromXml]]></value>
<value importAction="remove"><![CDATA[AnotherDataflowConf]]></value>
</values>
</property>
</fgo>
```

2. Run bootstrap to make sure that the custom configuration is updated.
3. Activate the new configuration unless a currently active configuration set was used during bootstrap.

13.7.1 Configuration Changes

The customization project technical lead should review all the configuration and switches that were added or modified since the original release and determine if these files must be populated to the database or not. Additionally, some switch definitions may move from the properties files to the database and need to be removed from the properties file.

13.7.2 Packaging

In versions prior to 5.2, packaging was defined through inclusions of specific property files that defined features. Dependencies were not enforced and ordering of includes were specified in each deployment.

For information about packaging in the 5.2 and later releases, see Packaging on page 221.

13.7.3 Analyzing Data Statistics

For information on how to analyze data statistics, see Gathering Schema Statistics on page 76.

13.7.4 Dropping Materialized Views

Drop the materialized views using the ant script DBPreExport. See DBPreExport on page 34 for more information.

13.7.5 Setting the Content Information

Set the following properties in the personal property file to determine where data files are exported to:

- `com.modeln._default.contentRoot=`<*location of the delta content root*>
- `com.modeln._default.contentSets=`<*content file sets*>
- `com.modeln._default.featureBuildModule=`<*content location of the specific feature*>
- `com.modeln._default.basePathSuffix=`<>

For example:

```
com.modeln._default.contentRoot=/home/jdoe/dev/54/projectName
com.modeln._default.contentSets=base,product,customer
com.modeln._default.featureBuildModule=company
com.modeln._default.basePathSuffix=content
```

The directory the data files will be exported to is:

`/contentRoot/featureBuildModule/basePathSuffix/contentSets`

- OR -

`/home/jdoe/dev/54/modeln/projectname/content/base/`

13.7.6 Running MigratePropsToConf

The MigratePropsToConf tool migrates your property files to the configuration directory structure.

1. Run MigratePropsToConf.
   a. Configure your ant environment following the instructions in How to Configure the Environment on page 31.
   b. Follow the instructions in MigratePropsToConf on page 39.
2. Copy the generated conf files to the `<feature>/conf/` directory and copy the generated content files to the `<feature>/content` directory.

13.7.6.1 Run Full Populate

Next, populate all of the delta content to the database.

1. Run Full Populate.
   a. Configure your ant environment following the instructions in How to Configure the Environment on page 31.

   b. In `app/conf/fgo/DataFlow.xml`:
   - remove `MapEdge` from the `FullPopulateDoNotLogDagList` section
   - remove PMImplementPricesPolulate from the data flow list. Review the other data flows for any that are not applicable to your migration.

   c. Follow the instructions in RunFullPopulate on page 37.

2. View the `RunFullPopulate.log` to locate any exceptions.

13.7.7 Troubleshooting

If you do not see any product in a search product window, check to see whether your default catalog view name changed. To do this:

1. Determine the default catalog view name values by running the following query:

```
Select value from MN_CONFIGURATION_VALUE where
config_id=(select CONFIGURATION_ID from MN_CONFIGURATION
where CONFIG_NAME like
'com.modeln.AppSwitch.products.defaultCatalogViewName');
```

2. Compare the query result to your application catalog name by looking in the **Model N application > Products > Product Folders > Catalog > Catalog Name**.
3. If the query result is different from your application catalog name, load the attached `defaultCatalogViewNameSwitchOverrides.xml` file using the `ConfigurationFromXml` data flow and then change the value of `com.modeln.AppSwitch.products.defaultCatalogViewName` according to the project catalog name.

13.8 How to Create a Migration Task

To create a migration task for your customizations, write the task and then wrap it in an .xml file as follows:

```
<Entries>
  <MigrationTask (1)
      TaskName="ChangeSomeColToNotNull"
      Description="Changes some_col in mn_some_table to NOT
NULL."
      Version="4.4.6patch1" (2)
      OrderNum="1" (3)
      ProductName="platform" (4)
      IgnoreErrors="false">
    <TaskText>
      ALTER TABLE mn_some_table MODIFY some_col NOT NULL;
    </TaskText>
  </MigrationTask>
</Entries>
```

(1) A migration task has a state that can be "pending", "succeeded", "failed", "skipped", or "invalid". A task with a state of "succeeded" or "skipped" is called completed. A task with a state of "pending", "failed", or "invalid" is called uncompleted. If the `IgnoreErrors` attribute is set to `true`, to allow migration to continue, the task is set to the "skipped" state instead of the "failed" state in the case of errors.

(2) The `Version` is comprised of the major version, minor version, subminor version, category name, and category number. The category name can be "alpha", "beta", "final" (corresponds to a release), or "patch". These names are lexicographically sorted. The category number is a floating point number, so if you want to insert a set of tasks between 4.4.6 patch 2 and 4.4.6 patch 3, you could use 4.4.6 patch 2.5, for example. If the category name and category number are omitted, they are assumed to be "final" and 0, respectively.

(3) The value for `OrderNum` determines the order in which migration tasks are executed within each `Version`, and must be unique. The different `Version`s are ordered with respect to each other by the parts (for example, major version, minor version) and not lexicographically as whole strings. When specifying the `OrderNum`, you must choose the next unused `OrderNum` for the given `Version`.

(4) The `ProductName` is a key field needed to identify any existing migration script.

Once the migration task is ready to be executed, add it to a directory determined by the `contentRoot` properties. Typically, the directory is as follows:

```
modeln/<module>/content/base/Default/Importers/MigrationTask/
```

where the `ProductName` is the same as the `<module>` and can be `platform`, `app`, `compliance`, and so on.

13.9 Migrating Sales Validations to the Direct Loader Framework

Prior to the Model N 5.2 release, an FGO was created when the sales lines were uploaded. This, in turn, prompted the Java plug-in to validate the lines. Once the lines were validated, they were inserted into the table.

As of 5.2.1, the sales lines are uploaded using the Direct Loader Channel and a temporary table is created for the incoming data. This temporary table content is mapped directly to the `mn_sale` table. The validation process is kicked off after the data flow completes in order to increase performance and speed, and for memory usage.

For more information about the Direct Loader Channel, see Configuring Direct Loader on page 79.

13.9.1 Migration Process

This section explains how to migrate sales validations. The examples use `<projectName>` to refer to the specific name that is given to your project, for example, acmepharma.

To migrate sales validations:

1. Set up a development environment.
    a. Create the database, the database schema, and import the database.
    b. Set up the Direct Loader by following the instructions in Direct Loader Channel Configuration on page 79.

2. Identify custom validations.

   a. Create a relevant directory structure to configure the data flows and validation configurations.

   For example: `xxx\xxx\content\xxx\Default\Importers\xxx`

   b. Identify the custom validations in the project based on the project requirement.

   c. Determine which validations must be turned on or turned off for direct and indirect sales.

3. Update sales validations for the new framework. Design the validation configuration file according to the project requirements.

   - Indirect Sale ValidationConf

     The ValidationConf, `BothSaleValidation`, is the starting point for configuration/customization when lines are loaded as open lines. `BothSaleValidation` extends the `SaleValidation.xml` file, which means that it inherits all the validation tasks and validation messages from the `SaleValidation.xml` and all of these validation tasks and validation messages are enabled by default. While designing the ValidationConf file for your project, check the validations which are used and not used in the current installation, map the validations with the validation tasks and validation messages, and turn them off them depending on the out-of-the-box installation.

Note: If Indirect Sales are loaded as closed lines (as in GPSolo), `HistoricalSaleValidation` is the starting point for configuration/ customization as it already has the price resolution tasks turned off.

   - Direct Sale ValidationConf

     The ValidationConf, `HistoricalDirectSaleValidation`, is the starting point for configuration/customization when lines are loaded as closed lines. `HistoricalDirectSaleValidation` uses the `SaleValidation.xml` file, which means that it inherits all the validation tasks and validation messages from the `SaleValidaiton.xml` and all of these validation tasks and messages are disabled by default. While designing the ValidationConf file for your project, check the validations which are used and not used in the current installation, map the validations with the validation tasks and validation messages, and turn them off them depending on the out-of-the-box installation.

4. Split the Direct Sales data flow. Add entry conf to the data flow to list the fields and their corresponding order. For example:

```
<EntryConf ConfName="<projectName>BothSaleFields"
 EntryType="IndirectSale"
 OrderNum="10"
 ImportAction="addModify">
 <FieldConf FieldName="ExternalRefId" OrderNum="1" />
 <FieldConf FieldName="ExternalLineType" OrderNum="2" />
 <FieldConf FieldName="ShipToCustomerNum" OrderNum="3" />
 <FieldConf FieldName="SoldToCustomerNum" OrderNum="4" />
 <FieldConf FieldName="ExternalItemId" OrderNum="5" />
 <FieldConf FieldName="ExternalContractId" OrderNum="6" />
 <FieldConf FieldName="ExternalInvoiceDate" OrderNum="7" />
 <FieldConf FieldName="ExternalInvoiceAmount.Value" OrderNum="8" />
 <FieldConf FieldName="ExternalInvoiceQuantityAmount" OrderNum="9" />
 <FieldConf FieldName="ExternalInvoiceUom" OrderNum="10" />
 <FieldConf FieldName="ExternalRequestDate" OrderNum="11" />
 <FieldConf FieldName="ExternalParentDistrBranchId" OrderNum="12" />
 <FieldConf FieldName="ExternalParentDistrName" OrderNum="13" />
 <FieldConf FieldName="ExternalParentDistrState" OrderNum="14" />
 <FieldConf FieldName="ExternalParentDistrZip" OrderNum="15" />
 <FieldConf FieldName="ExternalParentDistrCountry" OrderNum="16" />
 <FieldConf FieldName="ExternalParentDistrDebitMemo" OrderNum="17" />
 <FieldConf FieldName="ExtendedGrossAmt.Value" OrderNum="18" />
 <FieldConf FieldName="ExternalDistrCostAmount.Value" OrderNum="19" />
 <FieldConf FieldName="ExternalExtendedDistrCostAmount.Value"
OrderNum="20" />
 <FieldConf FieldName="ExternalExtendedDistrRebateAmount.Value"
OrderNum="21" />
 </EntryConf>
```

5. Develop custom validation tasks.

   For new custom validations:

   a. Create a customized PL/SQL package header and body to define the new validation so that all of your other customized validations can simply be added to this package. In this package, create a stored procedure for the custom validation.

   b. Create a property file to register the SQL scripts for the PL/SQL packages.

   c. Place all three files created in previous steps into your property path as defined in `<projectName>_sale_validation.properties`.

   For example:

```
…/props/ sale_validation/
<projectName>_sale_validation.sql
…/props/sale_validation/
<projectName>_sale_validation_body.sql
…/props/sale_validation/
<projectName>_sale_validation.properties
```

   d. Include the property created in Step b in your application property file.

   e. Run the DBPostImport ant-based tool to compile the PL/SQL package created in Step a.

   f. Check with your SQL tool to make sure that the package is compiled without any warnings or errors. If there are any warnings or errors, debug your PL/SQL code until the compile is clean.

   g. Add a Validation Task to the ValidationConf for designated data flows. To add a validation task, add the task and register the error message. For example:

Adding a task:

```
        <Task TaskName="ValidateCot"
            Order="9001"
            Enabled="true"
            Required="true"
            Type="PROCEDURE"

Operation="<projectName>_sale_validation_pkg.valid
ate_cot" />
```

Make sure that Order number is proper. It is recommended that you start with 9000 as currently all out-of- the-box validation task order numbers are before 4000.

Registering the message and severity level (here it is registered as FATAL):

```
<Message MessageName="CotExists"
            MsgType="root.msg.sale.CotExists"
             Index="143"
            DefaultSeverity="FATAL"
             TaskName="ValidateCot"
            Enabled="true"/>
```

The Index number must be unique across all sale validation Message registrations. Currently, the out-of- the-box application has fewer than 100.

h. Follow the same procedure for `HistoricalDirectSale`.

To override existing functionality for custom validations:

a. Create a stored procedure to override the custom validations.

b. Register the validation task and message as follows:

```
        <Task TaskName="ValidateDistrPrice"
            Operation=”<projectName>_sale_validation_pkg.
            validate_distributor_price”/>
```

c. Create a directory structure for the stored procedure. For example:

`\xxx\xxx\props\xxx`

d. Include the custom validations files in the property file.

e. Register the validation tasks with the property file as explained in the preceding example in step b. For example:

```
com.modeln.db.ora.sqlScript.<projectName>_SALE_VALIDATION_HEADER=
    sql<sale_validation/<projectName>_sale_validation.sql>
com.modeln.db.ora.sqlScript.<projectName>_SALE_VALIDATION_BODY=
    sql<sale_validation/<projectName>_sale_validation_body.sql>
```

f. Follow the same procedure for `HistoricalDirectSale`.

6. Load the data flows:

   a. Load the validation configurations.

   Load `<projectName>BothSaleValidation.xml` and `<projectName>HistoricalDirectSaleValidation.xml` using the `ValidationConfFromXml` data flow.

   b. Load the data flows:

   Use `DataFlowFromXml` to load `BothSaleFromDirectLoader.xml` and `HistoricalDirectSaleFromDirectLoader.xml`.

   Use the appropriate data flow to load the sale line.

13.9.2 Troubleshooting

While loading the sale line, if you get any errors even after the data flow and validation configurations are properly loaded, determine whether all of the indexes in the database are enabled. If the indexes are not enabled, run the following SQL script:

```
alter index MN_SALE_BIT0_IDX ENABLE;
```

Check the DirectLoader setup. If the DirectLoader is not set up:

1. Follow the instructions in Direct Loader Channel Configuration on page 79.
2. Run the DBPostImport ant-based tool.
3. Recompile all stored procedures.

Make sure that the sale line is formatted as follows:

- The date format in the sale line must be in mm/dd/yyyy format.
- Make sure that the amount field (for example, invoice amount) is not associated with any currency type.
- Make sure that the Quantity field in the sales lines is not associated with `UOM.UOM` and that quantities are treated as one field in any Model N version that is prior to the 5.2 version. Starting in the Model N 5.2 version, UOM and quantity are treated as two separate fields.

13.10 Rebates Migration

This section describes how to migrate customized bucketing queries to allow them to work with the old bucketing queries framework.

The `com.modeln.AppSwitch.rebates.useOldBucketingQueries.xml` application switch, which is located in the `rebates/content/base/Global/Importers/Configuration/RebatesSwitches.xml` file, allows you to switch between using the bucketing queries framework that existed in the 5.0 and earlier releases to the bucketing queries framework that was introduced in the 5.2 release. To load the application switch, use the `GlobalConfigurationFromXml` data flow. If you want to use the framework that existed in 5.0 and earlier, set the value of the application switch to `true`. To use the framework that was introduced in the 5.2 release, set it to `false`.

13.10.1 Migrating Bucketing Queries

To migrate bucketing queries, first identify the customized queries that exist in your project and determine whether these queries have changed in the core product between the release you are on currently and the release to which you are migrating.

13.10.1.1 Customized Queries in Properties Files

To determine whether you have any customized queries in property files:

1. Find all of the customized queries in the project properties files.
2. Determine whether any of the customized queries were changed in the product releases between the versions using the results from the DiffReport tool.

13.10.1.2 Customized Queries in Class Files

1. Find all of the customized queries in the project class files.
2. Determine whether any of the customized queries were changed in the product releases between the versions using the results from the DiffReport tool.

13.11 Report Migration

Starting with Model N release 5.2, reporting differs from earlier Model N releases as follows:

- The Model N 5.2 application uses Cognos 8.
- Prior to the 5.2 release, Model N used ReportNet.
- Most of the out-of-the-box reports in the Model N 5.2 release use the Cognos models rather than SQL-based methods.
- The Ad-hoc Cognos models have been scaled down to include only the essential data elements.

The following are reasons to migrate your reports if you are migrating from a pre-5.2 release to Model N 5.2 or greater:

- You have any custom SQL-based reports.
- Although there were not many changes made to the database schemas under some circumstances, custom fields or previous fields no longer exist.
- Cognos 8 brings some new XML standards that may affect the customized reports.

Note: You should take the out-of-the-box reports from the 5.2. deployment, but you should migrate the custom reports. Cognos 8 uses a new XML format so custom reports will not work if you simply copy or import them into the 5.2 environment.

To migrate your custom reports:

1. Place the ZIP file, which contains a copy of all the reports to be migrated, in the deployment folder of Cognos.

   ```
   C:\cognos\c8\deployment
   ```

2. Import the ZIP file by navigating to **Cognos Connection > Tools > Content Administration**.
3. In the New Import Wizard, select the folder that you want to import and follow the wizard.
4. You can rename the folder if needed, and then click **Next**.
5. Select the folders that you want to import, and then click **Next**. You can rename the folders that will appear in the Public Folder.
6. Select the options that apply, and then click **Next**.
7. Verify the accuracy of all of the options you have chosen and make changes if needed.
8. Under Action, click **Save** and run once.
9. Click **Finish**.
10. Select the appropriate options, and then click **Run**.

    You can see the required Report folders in the Public Folder tab.

If you have copied the reports to a `/customized/` folder:

1. Convert ReportNet report specifications to Cognos 8 report specifications using the `RSupgrade.bat` tool in the `Cognos 8\bin` directory.

   For example:

   ```
   RSupgrade -i CRN_reportSpec -o C8_reportSpec
   
   RSupgrade -i iff.xml -o iff_cognos8.xml
   ```

2. Migrate the reports by exporting them, importing them, or using the Cognos `RSupgrade.bat` tool.
3. Open and validate each report.
4. Fix any errors.
5. Run the report via Cognos.
6. Copy the report to the `/standard/`folder.
7. Run the report via the Model N application (Standard Templates)
8. Verify the results.

Note: You can export or import the custom reports after the 5.2 Cognos deployment has been completed, but make sure to replace the 5.2 out-of-the-box reports with the 5.0 out-of-the-box reports.

13.11.1 Database Schema Changes

When you migrate your reports, you must be careful to note any schema changes between the database from which you are migrating and the database to which you are migrating. You can use the TOAD tool to compare the schemas between the two databases.

To use TOAD:

1. Start TOAD.
2. Open the database connections for the databases that you want to compare.

   Select DBA > Compare Schemas.
3. Select Connects and Schemas.
4. Select Options/Object Types to choose which components you want to compare.
5. Select Options and change the "Stop when # of differences …" to 9999

   Unselect indexes because they are named numerically and will cause too many differences. The number of differences is limited.
6. Save the output by clicking the Disk button in the Results tab.

13.11.2 Troubleshooting

The following is a list of issues you may encounter while migrating your reports as well as explanations of how to solve them.

- If you encounter an error while validating a report, replace the # and the $ that are used in SQL alias names with a value such as NO and AMT.
- If all of the prompts are not displaying in the application, create separate blocks for each parameter.
- If SQL and report are not validating, check to see whether the table names are qualified in the SQL. If they are, update them accordingly or remove them.
- If you cannot select the report properly because of the date parameters, convert the Cognos date and date/time prompt values to SQL formats. For example, convert them from:

  `#prompt(‘Date Parameter’,’date’)#`

  to:

  `TO_DATE(#sq(prompt(‘Date Parameter’,’date’))#,’YYYY-MM-DD’)`
- If the report headers are missing, manually recreate them.
- If an error occurs while you are validating a query for query-based reports, remove the schema name from the table naming convention. For example, change:

  `modeln.mn_sale_validation`

  to

  `mn_sale_validation`

13.12 Price Master Configuration Migration

In 5.3, to support direct and indirect prices, `direct_indirect_type` was added to the price master config, which controls the type of price returned by Price Master. This column is a bit flag column, with the following values:

Code 13-2: direct_indirect_type Column

```
public final static CMnPriceMasterDirectIndirectType DIRECT =
```

Code 13-2: direct_indirect_type Column (Continued)

```
new CMnPriceMasterDirectIndirectType("DIRECT", 1, "Direct");

public final static CMnPriceMasterDirectIndirectType INDIRECT =
new CMnPriceMasterDirectIndirectType("INDIRECT", 2, "Indirect");

public final static CMnPriceMasterDirectIndirectType DIRECT_INDIRECT=
new CMnPriceMasterDirectIndirectType("DIRIND", 4, "Direct/Indirect");
```

These bit values can be combined. For example, to create a price master configuration that returns a price from both Direct and Direct/Indirect contracts, the `direct_indirect_type` value for the price master configuration would be set to 1(Direct) + 4(Direct/Indirect) = 5.

In 5.4, price master configurations needed be to created such that the following criteria was met:

> To support multiple organizations, from any organization, price master must be able to find all three configurations (IndirectType, DirectType and AllDirectIndirectType) either associated with the organization, or associated with one of its ascendent organizations.
>
> - IndirectType - returns price from indirect and direct/indirect contracts
> - DirectType - returns prices from direct and direct/indirect contracts
> - AllDirectIndirectType - returns prices from direct, indirect and direct/indirect contracts

Examples of price master configurations are available under /pharmaconfig/content/product/Global/Importers/PriceMasterConfig/CotPriceMasterC onfig.xml.

13.13 Custom Price Type Migration

Changes in the database tables for performance improvement and time zone-awareness in the 5.3 and 5.4 releases resulted in several changes to the Price Type content. As a result, several manual migration tasks are needed.

13.13.1 Fix Price Type Formulas

The following steps are required to get the price type formulas to run. These still go against the mn_sale view, so performance will be an issue, but the calculation will not encounter exceptions.

13.13.1.1 Changing Data Types

In Model N release 5.4, the JAXB jar was upgraded from Java-1.4 compatible to Java 1.5-compatible. This third-party JAXB tool is used as part of the price type builder tool. As a result, some of the inner workings of the price type builder have changed.

The `type` attribute on Columns (such as CalcColumn and DBColumn) now comes from a specific list of:

- index
- integer
- decimal
- string
- ndc
- ndc9
- resultkind
- customerid
- externallinelinkid
- worksheetname
- itemid
- extcontractid
- netprice

Note: Using a different value, such as `result_kind` instead of `resultkind`, will cause a null pointer exception to occur since the `type` attribute will be set to null, and this is a required field for most Columns (such as CalcColumns).

Solution

To resolve this issue, enter only values specified in the list.

13.13.1.2 Non-existent Columns

In Model N release 5.4, changed were made to the `mn_date_dim` table to remove incorrect data now that the time zone awareness was added. Some of the external queries relied on these columns.

The removed columns are:

- start_of_day
- end_of_day
- date_sql

Some sample usages of these columns are:

```
start_of_day = to_timestamp( ''{workbook.periodYearStartDate, time,
yyyy-MM-dd HH:mm:ss.SSS}'', ''YYYY-MM-DD HH24:MI:SS.FF'')
```

```
start_of_day = mn_struct_line_item_ver.start_date
```

```
start_of_day = ADD_MONTHS( to_date(
''{calculationBasis.dataSourceFilterMap.DIR.startDate, time, yyyy-MM-
dd}'' , ''YYYY-MM-DD'' ), -11 )
```

Solution

There are many possible solutions, depending on the specific situation.

- In the first example, switch `mn_date_dim.start_of_day` to `mn_date_dim.date_num`, and format the `workbook.periodYearStartDate` as `yyyyMMdd` instead of as `timestamp`.

  For example, the replacement for above would be:

```
date_num = {workbook.periodYearStartDate, time, yyyyMMdd}
```

- In the second and third example, switch from `mn_date_dim.start_of_day` to `mn_date_dim.date_num`, and format the right hand side as `yyyyMMdd`.

  For example, the second example would be replaced with:

```
date_num = to_char( mn_struct_line_item_ver.start_date, ''yyyyMMdd'' )
```

  The third example would be replaced with:

```
date_num = to_char( ADD_MONTHS( to_date(
''{calculationBasis.dataSourceFilterMap.DIR.startDate, time, yyyy-MM-
dd}'' , ''YYYY-MM-DD'' ), -11 ), ''yyyyMMdd'' )
```

13.13.1.3 Data Source Enumeration Refactoring

New data sources must be able to be created while the application is running. This cannot be achieved if the data source is an enumeration. In release 5.4, the data source was moved from an enumeration to a string field.

Note: This only affects historical price type content files. The existing content in the database will be migrated by way of a migration script.

Solution

The following changes to the price type content .xml files are needed:

- `DataSourceConfig` has a new attribute, `DataSourceName`, which is the user friendly name of the data source (This can be copied from the `DisplayName` of the old `CMnDataSource` enum).
- `DataSourceConfig` and `DataSourcesFilter` have a new attribute, `DataSourceType`, which is one of the following values (which can be copied from the `DataSourceType` of the old `CMnDataSource` enum):
  - DIRECT_SALES
  - INDIRECT_SALES
  - REBATES
  - CHARGEBACKS
  - ADJUSTMENTS
- `DataSourceConfig` has an attribute `DataSource` and `DataSourcesFilter`

has an attribute `DataSources`.

- If the `DataSourcesFilter` and `DataSourceConfig` are referring to the same data source, then the values of these must match. Previously, the enumeration was able to know that DIR=DIRECT_SALES, INDIR=INDIRECT_SALES.
- The .xml file used by the PriceTypeBuilder tool refers to `calculationBasis.dataSourceFilterMap.ABC` in several places. The *ABC* string must also match identically the value of `DataSourceConfig.DataSource`.
- The .xml file used by the PriceTypeBuilder tool has a `DatasourceFilter` inside each DBWorksheet. This `DatasourceFilter` has a `name` attribute, which also must match identically the value of `DataSourceConfig.DataSource`.

13.13.2 Correctness Changes

The following changes are needed to increase correctness.

13.13.2.1 Replace References to Date Dim FK with Specific Date Columns

In Model N release 5.4, the Model N suite became “time zone-aware”. This means that the date dim IDs are now deprecated, as the information they contain is potentially incorrect. The corresponding timestamp column must be used instead:

Table 13-1: date_dim to timestamp Mapping

date_dim FK column	timestamp column
inv_date_id	inv_date_sql
paid_date_dim_id	inv_date_sql
transaction_date_dim_id	external_transcation_date
receive_date_dim_id	received_date
load_date_dim_id	load_date
request_date_dim_id	external_request_date
process_date_dim_id	external_process_date
approved_date_dim_id	external_approved_date
sub_close_date_dim_id	sub_close_date
chgbk_close_date_dim_id	chgbk_close_date

Solution

To resolve this issue:

1. Replace references of “dataSelectionType.dateDimFKColumn” with “dataSelectionType.saleDateColumn”.
2. Replace the StartDim and EndDim parameters with timestamps. For example, replace:

```
<Parameter name="DS.StartDim">
<Value><![CDATA[
{calculationBasis.dataSourceFilterMap.DIR.startDate, time, yyyyMMdd}
]]></Value>
</Parameter>
```

with

```
<Parameter name="DS.StartDate">
<Value><![CDATA[
to_timestamp(
‘{calculationBasis.dataSourceFilterMap.DIR.startDate, time, yyyy-MM-dd
HH:mm:ss}’,
‘YYYY-MM-DD HH24:MI:SS’
)
]]></Value>
</Parameter>
```

3. Since the “dateDimFKColumn” was a number and the “saleDateColumn” is a timestamp, you also need to update the portion of the query that was being compared against. For example, the SQL snippet

```
AND DSQ.[$DS.DateCol] <=
(select date_dim_id from mn_date_dim where date_num = [$DS.EndDim])
```

must be replaced with

```
AND DSQ.[$DS.DateCol] <= [$DS.EndDate]
```

4. In DBWorksheets, replace the sub_close_date_dim_id (and chgbk_close_date_dim_id) comparison against the data cutoff date with a comparison between sub_close_date (and chgbk_close_date) against a timestamp of the data cutoff date. For example, replace

```
AND DSQ.sub_close_date_dim_id <=
(select date_dim_id from mn_date_dim where date_num = [$DataCutoffDim] )
```

with

```
AND DSQ.sub_close_date <= [$DataCutoffDate]
```

where [$DataCutoffDate] is defined as

```
to_timestamp(
'{calculationBasis.dataCutoffDate,time,yyyy-MM-dd HH:mi:ss}' ,
'YYYY-MM-DD HH24:MI:SS'
)
```

13.13.2.2 Persist Required Intermediate Results

Starting in Model N release 5.3, the table where the intermediate steps of the GP calculation were stored is no longer used as a long term storage of calculated values. From that point on, at the end of the calculation, the `mn_workbook_int_result` is truncated.

Solution

Any information in `mn_workbook_int_result` that needs to be persisted after the calculation must be moved to `mn_wb_pers_int_result`. Worksheets now have a "persisted" attribute which is *false* by default. Set this attribute to true for any worksheets which need to be persisted after the calculation.

13.13.3 Performance Changes

The following sections cover the changes to stop using the `mn_sale` view.

13.13.3.1 Replace mn_sale View with Sales Tables

In Model N release 5.3 the `mn_sale` table was deleted and eight new sales tables were created: four for each type of sale, and within those four divisions, further divided into one table for open transactions and one for closed transactions. The `mn_sale` table became an `mn_sale` view, which has performance issues.

At the same time, GP introduced "sales subset cache" tables. These tables have fewer rows and fewer columns than the real sale tables. Fewer rows and fewer columns greatly improves performance.

Solution

These two causes can be implemented at the same time to improve performance.

Each DBWorksheet has an attribute `dBTableName` and `dBSubsetTableName`. These values need to be updated based on the `sale_type` value of the data source:

Table 13-2: Sales Type Mappings

Type	sale_type	dBTableName	dBSubsetTableName
Direct	1	mn_dir_sale_closed	mn_gp_dir_sale
Indirect / Chargeback	2, 4, 6	mn_indir_sale_closed	mn_gp_indir_sale
Rebates	8	mn_rbtpmt_sale_closed	mn_rbtpmt_sale
Adjustments	16	mn_custom_sale_closed	mn_gp_custom_sale

Note: The PriceType must have a `cacheSaleSubsets` attribute set to *true* in order to use the subset tables during the calculation. Otherwise, the regular sale tables will be used. By default the value of this property is *false*.

13.13.3.2 Add Custom Column to Sales Subset Tables

In Model N release 5.3, GP introduced the “sales subset cache” tables. These tables only have columns which are used by the calculation. The list of columns used is defined in the properties as:

- com.modeln.gp.saleSubsetAttrList.base.date_ids
- com.modeln.gp.saleSubsetAttrList.base.filterCriteria
- com.modeln.gp.saleSubsetAttrList.base.selected
- com.modeln.gp.saleSubsetAttrList.mn_gp_dir_sale
- com.modeln.gp.saleSubsetAttrList.mn_gp_indir_sale
- com.modeln.gp.saleSubsetAttrList.mn_gp_rbtpmt_sale
- com.modeln.gp.saleSubsetAttrList.mn_gp_custom_sale

Solution

Customized price types must append these lists to add needed columns into the list, and then run bootstrap to update the sales subset cache tables.

13.13.3.3 Unknown Sale Columns

In Model N release 5.3, the names of some sales columns were changed. For example:

Table 13-3: Renamed Sales Columns

Former name	New name
ext_inv_amt	total_inv_amt
ext_contract_amt	total_contract_amt

Note: mn_custom_sale_closed no longer has a “ppd” column.

Solution

Replace all references to old columns with references to new columns.

Financial adjustments are no longer are discounted by “(1-ppd)”. All the formulas which use this must be updated to remove references to “ppd” from financial adjustment DBWorksheets.

13.13.3.4 More Changes to DBWorksheet’s SQLFilters

With the switch to using the GP sales subset cache tables, changes are needed to the SQLFilters. One advantage of the GP sales subset cache tables is fewer columns in these tables than the regular sales tables. This means some columns referenced in the SQLFilters are no longer in the GP sales subset cache tables because that data is filtered by the mechanism which populates these sales subset cache tables.

Solution

- Remove conditions on:
    - Realm Criterion (realm_num in the generated property, [RC:Sale:<alias>] in the DBWorksheet's SQLFilter)
    - Lifecycle Status (lifecycle_status)
    - Line Validation Status (status_num_gp)
- For direct sales, rebates, and financial adjustments, remove the condition on sale_type.
- For indirect sales and chargebacks, replace the BITAND logic on sale_type with an OR clause, such as:

```
(ISQ.sale_type = 4 OR ISQ.sale_type = 6)
```

- For Financial adjustments, add a new where clause against custom_sale_type:

```
custom_sale_type = 'CS.FINADJ'
```

Note: The value being filtered for custom_sale_type is "CS.FINADJ" in 5.4, but "FINANC_ADJ" in 5.3. See Fix Financial Adjustment Enum on page 214 for details.

13.13.3.5 Fix Financial Adjustment Enum

Starting in Model N release 5.3, the table `mn_custom_sale_closed` has a column `custom_sale_type`. The value of this column for financial adjustments was FINANC_ADJ. In the GP queries, a condition was added to filter only the sales lines in `mn_custom_closed` that have a `custom_sale_type` of FINANC_ADJ (refer to More Changes to DBWorksheet's SQLFilters on page 213).

In Model N release 5.4, the value of the enumeration was changed in the sales application from FINANC_ADJ to CS.FINADJ. A migration script is used to migrate existing content for the sales application, but you need to manually change the GP external queries.

Solution

Do a search and replace to replace FINANC_ADJ with CS.FINAD". Limit the files being searched to just the property files used by the GP calculation queries.

13.13.3.6 Improve DatasourceFilters

The old queries against the sale table looked something like:

```
accuracy = ...
AND sale_id IN ( <data source filter sub query> )
AND inv_date_id <= ...
AND inv_date_id >= ...
```

This turns out to have performance issues since the data source filter is also querying the sales table. As of Model N release 5.3 there is now an attribute on the DBWorksheet’s DatasourceFilter named `generateQuery` which controls how the query is generated and how the code in the calc plug-in class (`CMnMAMPCalcPluginV522`) is generated.

The new generated query would contain the data source filter criterion directly in the where clause, such as:

```
accuracy = ...
AND <data source filter criterion>
AND inv_date_id <= ...
AND inv_date_id >= ...
```

Solution

Add generateQuery=”false” to the DatasourceFilters on DBWorksheets, and remove the “sale_id IN ([SQ:Filter])” from the SQLFilter. By default, generateQuery is true.

13.13.4 “Model-Aware” Changes to Utilize 5.4 GP Features

In Model N release 5.4, the GP price type formulas are stored in the database. This means:

- many of the customer requests to Model N that are currently classified as "customizations" can be re-classified as "configurations", which the users themselves can do through the UI (without having to bring down the application and redeploy a new build)
- the Model N platform can handle versioning of price type formulas (and generated queries) better than current approach of versioning property files

Following are the actions needed to move a price type into this “model-aware” price type, by merging the .xml file used by the PriceTypeBuilder tool with the .xml file that contains the price type content.

13.13.4.1 Change Data Cutoff Date from Date to Timestamp

In Model N release 5.4, GP allows the Data Cutoff Date of a workbook to be set to the granularity of the hour and minute, instead of just to a given day. This feature is not used by the price type unless the price type formula is modified.

Solution

Most price types reference the data cutoff as a date dim, such as:

```
<Parameter name="DataCutoffDim">
<Value><![CDATA[
{calculationBasis.dataCutoffDate,time,yyyyMMdd}
]]></Value>
</Parameter>
```

This can be replaced with:

```
<Parameter name="DataCutoffDate">
<Value><![CDATA[
   to_timestamp(
   ‘{calculationBasis.dataCutoffDate, time, yyyy-MM-dd HH:mm:ss}’ ,
    'YYYY-MM-DD HH24:MI:SS'
    )
]]></Value>
</Parameter>
```

Now the full timestamp is part of the parameter, instead of just the year, month, and day.

The usages of the old parameter were usually of the form:

```
... < to_date('[$DataCutoffDim]', 'YYYYMMDD')+1
```

The usages of the new parameter are of the form:

```
... < [$DataCutoffDate]
```

13.13.4.2 Merge PTB Input XML and PriceTypeFromXML Data Flow Input .xml Files

As explained previously, in Model N release 5.4, the GP price type formulas were moved from .xml files stored on the file system into the database. The generated calculation logic is moved from external queries (in property files) and Java classes into the database.

This allows for leveraging the platform framework to version the formulas and for some changes to the formula through the user interface.

However, this requires merging the file that used to be an input into the PriceTypeBuilder tool with the file that was loaded into the application via the PriceTypeFromXml data flow.

Note: The PriceTypeBuilder tool is still backwards-compatible, and can still be used from the command line to generate property files and Java classes.

The PriceTypeFromXml data flow is still backwards-compatible, and can be used to load both the old price type content, as well as the new “model-aware” price type content.

13.14 Transaction Amount Data Source Filters

Some customers may have created data source filters against `Transaction Amount`. These filters were treating `Transaction Amount` as a money object, but as of Model N release 5.4.1, need to treat it as a decimal object.

The following two queries must be run to find out whether you have any filters defined against the `Transaction Amount`:

Query the Most Recent Versions

```
SELECT filter_id, ver_num, to_char(expression)
FROM mn_filter
WHERE expression LIKE '%TX!ExternalInvoiceAmount%' ;
```

Query the Version Table

```
SELECT filter_id, ver_num, to_char(expression)
FROM mn_filter_ver
WHERE expression LIKE '%TX!ExternalInvoiceAmount%' ;
```

For every result by the first `SELECT` query, the following `UPDATE` query must be run:

```
UPDATE mn_filter
SET expression = ?1
WHERE filter_id = ?2 AND ver_num = ?3 ;
```

For every result by the second `SELECT` query, the following `UPDATE` query must be run:

```
UPDATE mn_filter_ver
SET expression = ?1
WHERE filter_id = ?2 AND ver_num = ?3 ;
```

In the previous UPDATE queries, the arguments `?2` and `?3` must be replaced with the `filter_id` and the `ver_num` returned by the `SELECT` query previously run.

The argument `?1` needs to be manually created. The initial `SELECT` query will have returned something like the following:

```
<logicalOp name="and">
  <logicalOp name="and">

    ... other filters ...

# a filter such as "Transaction Amount Is Less Than ___", "Transaction
Amount Is Equal To ____", etc.
    <comparisonOp name="lt">
      <attribute direction="left" name="TX!ExternalInvoiceAmount"/>
      <embedded direction="right">
        <constant sourceType="Money">
          <Value scale="2">2.00</Value>
          <Currency>USD</Currency>
        </constant>
      </embedded>
      <property name="valueType"><value
sourceType="Class">com.modeln.util.unit.CMnMoney</value></property>
    </comparisonOp>

# a filter such as "Transaction Amount Is Not Defined"
    <isNull name="TX!ExternalInvoiceAmount">
      <property name="valueType"><value
sourceType="Class">com.modeln.util.unit.CMnMoney</value></property>
    </isNull>

# a filter such as "Transaction Amount Is Defined"
    <isNotNull name="TX!ExternalInvoiceAmount">
      <property name="valueType"><value
sourceType="Class">com.modeln.util.unit.CMnMoney</value></property>
    </isNotNull>

    ... other properties ...

  </logicalOp>
</logicalOp>
```

Note: The previous example has been formatted for better viewing. It is actually stored as one long text without spaces and newlines.

The filters against `Transaction Amount` need to be updated in the following ways to change from type money to decimal:

1. For `comparisonOp`, `isNull`, and `isNotNull` elements, the text:

```
      <property name="valueType"><value
sourceType="Class">com.modeln.util.unit.CMnMoney</value></property>
```

must be replaced with:

```
      <property name="valueType"><value
sourceType="Class">java.lang.Double</value></property>
```

2. For `comparisonOp` elements, the embedded value needs to be modified from:

```
      <embedded direction="right">
        <constant sourceType="Money">
          <Value scale="2">2.00</Value>
          <Currency>USD</Currency>
        </constant>
      </embedded>
```

to:

```
      <embedded direction="right">
        <constant sourceType="double">2.0</constant>
      </embedded>
```

Important: The value "2.0" is just an example. Be sure to use the same value in your `UPDATE` query as was returned by the `SELECT` query. So the argument `?1` to the `UPDATE` query would be the value returned as "expression" in the `SELECT` query, transformed according to the two rules defined earlier.

13.15 Migrating Data Export Configurations

As of Model N release 5.4.0, significant changes were made to the data export portion of the Government Pricing Application. Since this is usually a highly customized area, refer to the Data Export Configurations chapter of the *Government Pricing Developer's Guide* for information on the data export functionality and adjust your customizations as needed.

14 Configuration Options

The following sections describe general application functionality that can be enabled or configured using the application properties.

- Packaging
- Time Zones
- Thread Pools

For information about logging levels, see the chapter on logging levels in the *Technical Release Notes*.

14.1 Packaging

This section provides information on the following:

- Packaging Structure
- Creating a Custom Configuration
- Application Modules

14.1.1 Packaging Structure

The packaging structure consists of a definition of the allowable configurations and a specification of which configuration to use for a given deployment.

14.1.1.1 Allowable Configurations

This section provides the design for how the allowable configurations are defined in the Model N system. An allowable configuration is simply a configuration that does not violate any constraints.

14.1.1.2 Configuration Definition

The allowable configurations are defined in a .xml configuration file. There are three types of configuration files that are loaded, in order, during startup:

- Base Configuration

  This file will provide any vertical-agnostic features or feature groups. This should provide a definition of all vertical agnostic application modules that can be deployed. This is a required configuration file and there can be only a single base configuration file.

- Vertical Configuration

  This file will provide any vertical-specific features or feature groups. In addition, any extensions to the base configuration are specified here. This is a required configuration file and there is expected to be a single configuration file per vertical.

- Custom Configuration

  This file will provide any custom configurations that may be added during a deployment. This is an optional configuration file and there may be any number of custom configuration files.

See Application Modules for example base, Medical Device and Pharmaceutical configurations.

14.1.1.3 XML Configuration Format

The root element of the XML configuration file is `Configuration` and may contain any one of the following common elements:

- Description Element
- Dependency Element
- Feature Element
- Feature Group Element
- Feature Overlay Element
- Feature Group Overlay Element

Description Element

The description element defines a human-readable string that provides a textual definition.

XML Schema Definition

```
<xs:simpleType name="description">
  <xs:restriction base="xs:string">
    <xs:minLength>1</xs:minLength>
  </xs:restriction>
</xs:simpleType>
```

XML Example

```
<Description><![CDATA[Verbose & useful description!]]></Description>
```

Dependency Element

The dependency element defines a single dependency using a single name identifier.

XML Schema Definition

```
<xs:complexType name="dependency">
  <xs:attribute name="Name" type="xs:string" use="required"/>
</xs:complexType>
```

XML Example

```
<Dependency Name="aFeature"/>
```

Feature Element

The feature element defines a specific feature that is available. The name of the feature must correspond to the property file include name that defines the feature to be loaded with the `_includes` suffix. For example, if a feature is named medicaid, there must exist a property file that can be located using the specification `medicaid_includes`.

You must define any dependent feature or feature group prior to defining this feature definition - no forward references are supported.

XML Schema Definition

```
<xs:element name="Feature"> (1)
  <xs:complexType>
    <xs:any>
      <xs:element name="Description" type="description" (2)
                  minOccurs="1" maxOccurs="1"/>
      <xs:element name="Dependency" type="dependency" (3)
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:any>
    <xs:attribute name="Name" type="xs:string" use="required"/>
    <xs:attribute name="External" type="xs:boolean" use="optional"/>
(4)
  </xs:complexType>
</xs:element>
```

Code Section Notes:

(1) The feature name is an identifier used to uniquely identify this feature.

(2) The description is a human-readable string that can be shown to the user to provide information about what this feature is.

(3) The dependency elements define which features or feature groups are required for this feature. Therefore, if this feature is configured in a deployment, any dependent features or feature groups are automatically configured in the deployment.

(4) The external flag is an optional boolean flag that indicates whether the feature can be configured directly in a deployment using the `com.modeln.packaging.configuration` property. The default value is `false`. This flag cannot be overridden in an overlay.

XML Example

```
<Feature Name="aFeature" External="true">
  <Description>My Feature Definition</Description>
  <Dependency Name="OtherFeature1"/>
  <Dependency Name="OtherFeature2"/>
</Feature>
```

Feature Group Element

The feature group element aggregates a collection of features and/or feature groups into a single unit. Any dependent feature or feature group must be defined prior to this feature definition. For example, no forward references are supported.

XML Schema Definition

```
<xs:element name="FeatureGroup"> (1)
  <xs:complexType>
    <xs:any>
      <xs:element name="Description" type="description" (2)
                  minOccurs="1" maxOccurs="1"/>
      <xs:element name="Dependency" type="dependency" (3)
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:any>
    <xs:attribute name="Name" type="xs:string" use="required"/>
    <xs:attribute name="External" type="xs:boolean"
use="optional"/> (4)
  </xs:complexType>
</xs:element>
```

Code Section Notes:

(1) The name is an identifier used to uniquely identify this feature group. Note that features and feature groups may not use the same names.

(2) The description is a human-readable string that can be shown to the user to provide information about what this feature group is.

(3) The dependency elements define which features or feature groups are included in this group. Therefore, if this feature is configured in a deployment, any dependent features or feature groups are automatically configured in the deployment.

(4) The external flag is an optional boolean flag that indicates whether the feature can be configured directly in a deployment using the `com.modeln.packaging.configuration` property. The default value is `false`. This flag cannot be overridden in an overlay.

XML Example

```
<FeatureGroup Name="aFeatureGroup">
  <Description>My Feature Group Definition</Description>
  <Dependency Name="OtherFeature1"/>
  <Dependency Name="OtherFeature2"/>
  <Dependency Name="OtherFeatureGroup1"/>
</FeatureGroup>
```

Feature Overlay Element

The feature overlay element extends a feature to either modify the description or extend the set of dependencies. The overlay name is used to look up the feature being overlaid. The feature being overlaid must already exist. For example, forward references are not allowed.

Any number of overlays can be supported and will be applied in order. In practice, this means that there may be at most two overlays of a base configuration (for example, base vertical custom) and at most one overlay of a vertical specific configuration (for example, vertical custom).

XML Schema Definition

```
<xs:element name="FeatureOverlay"> (1)
  <xs:complexType>
    <xs:any>
      <xs:element name="Description" type="description" (2)
                  minOccurs="0" maxOccurs="1"/>
      <xs:element name="Dependency" type="dependency" (3)
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:any>
    <xs:attribute name="Name" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
```

Code Section Notes:

(1) The feature name is an identifier used to uniquely identify the feature being overlaid. It is an error to overlay a feature that does not exist.

(2) If an optional description is provided, the description replaces the description that exists in the overlaid feature.

(3) Any dependency elements augment the existing dependency list in the overlaid feature. As with the feature element, the features or feature groups listed in the dependency list must already exist. Any duplicates are ignored.

XML Example

```
<FeatureOverlay Name="aFeature">
  <Dependency Name="AdditionalFeature1"/>
</FeatureOverlay>
```

Feature Group Overlay Element

The feature group overlay element extends a feature group to either modify the description or extend the set of dependencies. The overlay name is used to look up the feature group being overlaid. The feature group being overlaid must already exist. For example, forward references are not allowed.

As with feature overlays, any number of overlays can be supported and will be applied in order. In practice, this means that there may be at most two overlays of a base configuration (for example, base vertical custom) and at most one overlay of a vertical specific configuration (for example, vertical custom).

XML Schema Definition

```
<xs:element name="FeatureGroupOverlay">
  <xs:complexType>
    <xs:any>
      <xs:element name="Description" type="description"
                  minOccurs="0" maxOccurs="1"/>
      <xs:element name="Dependency" type="dependency"
                  minOccurs="0" maxOccurs="unbounded"/>
    </xs:any>
    <xs:attribute name="Name" type="xs:string" use="optional"/>
  </xs:complexType>
</xs:element>
```

Code Section Notes:

(1) The feature name is an identifier used to uniquely identify the feature group being overlaid. It is an error to overlay a feature group that does not exist.

(2) If an optional description is provided, the description replaces the description that exists in the overlaid feature group.

(3) Any dependency elements augment the existing dependency list in the overlaid feature group. As with the feature element, the features or feature groups listed in the dependency list must already exist. Any duplicates are ignored.

XML Example

```
<FeatureGroupOverlay Name="aFeature">
  <Description>Modified Feature Group Description</Description>
  <Dependency Name="AdditionalFeature1"/>
</FeatureGroupOverlay>
```

14.1.1.4 Deployment Configuration

This section provides the design for how deployments specify the configuration they are using.

The deployment specification consists of two properties that must reside in the property file that is used to run the application. These two properties cannot be inherited through property file includes because these will be parsed and used to determine the set of files to include.

The `com.modeln.packaging.vertical` property defines the vertical configuration for this deployment. This will be used to locate the proper set of deployable features from the set of all allowable features.

The `com.modeln.packaging.configuration` property specifies a list of features for this deployment. This is a comma-separated list. Each entry in the list must be an allowed feature for the specified vertical and must be marked as an external.

The `com.modeln.packaging.customConfigNames` property specifies a list of custom configurations in this deployment. This is a comma-separated list, where each entry of the list defines a custom configuration .xml file that can be found in the classpath as a resource. Each name listed is used to generate a filename by adding the `PkgConfig.xml` suffix. The configuration files are located using the classpath. If any of the custom configuration files are not found, then a fatal exception is thrown.

14.1.2 Application Modules

The Model N application is composed of various application modules. Each module contains functional code and user interface definitions that determine application behavior. The application functionality and behavior depends upon the combination of modules used to build and run the application. The application modules are combined into specific application configurations. These configurations ensure that the proper dependencies are satisfied. The deployed configuration is determined by specifying one of the allowed configurations and is based on your requirements and licensing. For the list of certified configurations, see the Certified Configurations chapter in the *Technical Release Notes*.

To configure for a particular configuration, set the `com.modeln.packaging.vertical` property to either `pharma` or `meddev` and set the `com.modeln.packaging.configuration` property to `FullSuitePackage`.

The following table lists the certified configurations:

Configuration Name	Vertical Specification	Configuration Specification
Full Medical Device Application Suite	meddev	FullSuitePackage
Full Pharmaceutical Application Suite	pharma	FullSuitePackage

14.1.3 Creating a Custom Configuration

If this is the first custom configuration you are creating, you must create a custom packaging configuration .xml file. Otherwise, you may define custom configurations in an existing configuration .xml file.

To create and use a custom configuration:

1. Create an .xml file.
2. Put the .xml file in the classpath.

   For example, create `customPkgConfig.xml` and put it in the classpath.
3. Customize the configuration.

   For example, create a new feature group, `MyCustomPackage`, and include two new features, feature-A and feature-B. In this example, the code and properties associated with the features have already been developed.

Following is the file structure you would use to create this feature group with the feature-A and feature-B features:

```
<Configuration>

    <Feature Name="feature-A">
        <Description>My Feature A</Description>
    </Feature>

    <Feature Name="feature-B">
        <Description>My Feature B</Description>
        <Dependency Name="feature-A"/>
    </Feature>

    <FeatureGroup Name="FooPackage" External="true">
        <Description>My Custom Package</Description>
        <Dependency Name="FullSuitePackage"/>
        <Dependency Name="feature-A"/>
        <Dependency Name="feature-B"/>
    </FeatureGroup>

</Configuration>
```

In this example, feature-B depends on feature-A and this is explicitly specified in this file to ensure that the feature includes property files are included in the correct order. For example, `feature-A_includes.properties` is included before `feature-B_includes.properties`.

The `FooPackage` definition is marked as external, which means that it can be used directly in a deployment.

4. To specify that the deployment use a given custom configuration, you must specify the configuration in the deployment property file, as in the following example:

```
com.modeln.packaging.customConfigNames=custom

com.modeln.packaging.configuration=FooPackage
```

The `customConfigName` property indicates to the environment that you must load the `customPkgConfig.xml` file in addition to the out-of-the-box packaging configuration files. The `configuration` property indicates which configuration to use.

14.2 Time Zones

The time zone setting affects the use of dates within the application. If set incorrectly, the dates may appear incorrectly in the log file and may result in abnormal application behavior. Ensure that the time and locale settings are correct at the operating system level as well, and that the application server and database server clocks are synchronized.

Use the following syntax to set the time zone within the application property file:

```
com.modeln._default.timeZone=America/Los_Angeles
```

Model N supports a subset of the time zones supported by the`java.util.TimeZone` Java class. The unsupported time zones included:

- the deprecated three-letter time zones such as PST and EST
- the time zones not supported by the Oracle JDK being used with the database instance.

14.3 Thread Pools

Thread pools provide a mechanism for managing application processing. Threads are "light-weight sub-processes" that can separately execute operations in an application. Applications that run more than one operation at a time use threads to minimize overhead, as opposed to using "heavy-weight" processes. Applications use thread pools to provide a configurable number of pre-allocated threads for simultaneous operations. This further minimizes overhead, as opposed to allocating threads only when operations need them. Thread pools also allow for setting limits on how much processing an application can do by defining the maximum number of threads in the pool. When a thread is needed for an operation, but all previously allocated threads are in use, the thread pool allocates another thread and adds it to the pool, up to the maximum number of threads.

14.3.1 Standard Model N Thread Pools

Model N has several standard thread pools that are used by applications and by types of operation (for example, by asynchronous or background operations). These thread pools are named and each is configured with default values for the number of pre-allocated threads and the maximum number of threads. These default values are generally not changed for custom implementations. The standard Model N thread pools include:

- `pool.pricemon.pricecheck`
- `pool.membership.publish`
- `pool.gp.calculation`
- `pool.medicaid.uracalc`
- `pool.bidaward`
- `pool.cancel`
- `asynchronous`
- `synchronous`

For detailed information on each thread pool, see Standard Commands Executed in Standard Thread Pools.

14.3.2 Enforcing a Single-Threaded Thread in a Cluster

Some of the standard Model N thread pools must be configured to be single-threaded in a cluster – that is, to have a maximum of only one thread in the pool, and to use only one instance of the thread pool on one application server in the cluster. These pools are used to run standard operations that must be single-threaded. These pools include:

- `pool.medicaid.uracalc`
- `pool.bidaward`

Note: As of the 5.2.2 release, the `membership.publish` pool is no longer limited to a single thread.

Configurations of standard single-threaded pools – including the maximum number of threads - should not be changed for custom implementations. However, you must configure *commands* to enforce single-threading in a cluster. You do this by excluding commands from executing on applications servers *other than* the application server you designate as the one on which the thread pool must be used, effectively making the thread pool instance on that application server a singleton.

To configure standard thread pools:

1. Log into the Model N application as an IT administrator.
2. Go to the Nodes Administration page by clicking the **Administration > Nodes**.
3. Choose a node name from the list and click the link.

   The Logging page appears.
4. In the left navigation pane, under Node, click Thread Pools.
5. Populate the following fields by entering the information:
   - Minimum Pool Size
   - Maximum Pool Size

If you have existing thread pools that are not standard Model N thread pools, add a file to or modify the following XML to include the new thread pool configurations:

`<module>/base/Global/Importers/Configuration/ThreadConfig.xml`

Following is an example of how to modify the XML:

Code 14-1: Enforcing a Single-Thread in a Cluster

```
<?xml version="1.0" encoding="UTF-8" ?>
<Entries>
<Configuration Name="com.modeln.ThreadSvc.pool.audit.threadName"
Value_type="STRING" Description="Thread pool with threads called
audit">
  <ConfigValue Value="AuditThread" />
  </Configuration>
<Configuration
Name="com.modeln.ThreadSvc.pool.audit.threadKeepAliveTime"
Value_type="INTEGER" Description="Audit thread pool keep alive.">
  <ConfigValue Value="-1" />
  </Configuration>
```

Code 14-1: Enforcing a Single-Thread in a Cluster (Continued)

```
<Configuration Name="com.modeln.ThreadSvc.pool.audit.minimumPoolSize"
Value_type="INTEGER" Description="Audit thread pool minimum size">
  <ConfigValue Value="1" />
  </Configuration>
<Configuration Name="com.modeln.ThreadSvc.pool.audit.maximumPoolSize"
Value_type="INTEGER" Description="Audit thread pool maximum size">
  <ConfigValue Value="1" />
  </Configuration>
<Configuration Name="com.modeln.ThreadSvc.pool.audit.initialPoolSize"
Value_type="INTEGER" Description="audit thread pool initial size">
  <ConfigValue Value="1" />
  </Configuration>
<Configuration Name="com.modeln.ThreadSvc.pool.audit.includedTypes"
Value_type="STRING" Description="audit thread pool included resource
types">
  <ConfigValue Value="root.command.audit" />
  </Configuration>
<Configuration Name="com.modeln.ThreadSvc.namedPools" Value_type="LIST"
Description="custom named pools">
  <ConfigValue Value="pool.audit" />
  </Configuration>
  </Entries>
```

14.3.3 Standard Commands Executed in Standard Thread Pools

The following tables indicate standard commands executed in standard thread pools. For thread pools that are single-threaded in a cluster, these commands must be specifically included on the application server with the effective singleton thread pool – and excluded on other application servers - as in the examples above.

Note: As of the 5.2.2 release, the `publishMembershipCommand` no longer needs to run in a single-thread thread pool. The default size of its thread pool has been changed to 2 and you can set it to any value.

Table 14-1: pool.pricemon.pricecheck

pool.pricemon.pricecheck	
Thread Pools	`pool.pricemon.pricecheck`
Included Command Types	`root.command.pricemon.pricecheck.singleContractPriceCheckCommand` `root.command.pricemon.pricecheck.batchActiveContractsPriceCheckCommand`
Build Module	app
Single-Threaded in a Cluster?	No

Table 14-1: pool.pricemon.pricecheck (Continued)

pool.pricemon.pricecheck	
Default Size	1
Comments	Runs price monitoring single and batch contract checks.

Table 14-2: gp.calculation

gp.calculation	
Thread Pools	`pool.gp.calculation`
Included Command Types	`root.command.gp.calculation`
Build Module	gp
Single-Threaded in a Cluster?	No
Default Size	1
Comments	Runs GP calculations.

Table 14-3: medicaid

medicaid	
Thread Pools	`pool.medicaid.uracalc`

Included Command Types	`root.command.medicaid.uracalc.nightly`
	`root.command.medicaid.uracalc.user`
	`root.command.medicaid.uracalc.program`
	`root.command.medicaid.uracalc.master`
	`root.command.medicaid.uracalc.programList`
	`root.command.medicaid.autoadjust.ura`
	`root.command.medicaid.autoadjust.user`
	`root.command.medicaid.autoadjust.claimpool`

Build Module	medicaid
Single-Threaded in a Cluster?	Yes

Table 14-3: medicaid (Continued)

medicaid	
Default Size	1
Comments	Runs Medicaid URA calculations.

Table 14-4: bidaward

bidaward	
Thread Pools	`pool.bidaward`

Included Command Types	`root.command.bidaward.gen.contract`
	`root.command.bidaward.gen.nightly`
	`root.command.bidaward.gen.change`
	`root.command.bidaward.pub.specific`
	`root.command.bidaward.pub.full`

Build Module	pharmaba
Single-Threaded in a Cluster?	Yes
Default Size	1
Comments	Runs Bid Award generations.

Table 14-5: Cancel Thread

CancelThread	
Thread Pools	`pool.cancel`
Included Command Types	`root.command.cancel.*`
Build Module	platform
Single-Threaded in a Cluster?	No
Default Size	2
Comments	For cancel commands, ensures that there is always a thread to use.

Table 14-6: Asynchronous Thread

AsynchronousThread	
Thread Pools	`asynchronous`
Included Command Types	*
Build Module	platform
Single-Threaded in a Cluster?	No
Default Size	4
Comments	For scheduled commands.

Table 14-7: Synchronous Thread

SynchronousThread	
Thread Pools	`synchronous`
Included Command Types	*
Build Module	platform
Single-Threaded in a Cluster?	No
Default Size	15
Comments	For commands that execute immediately.

A Recommended Internet Explorer Settings

This appendix documents the Internet Explorer settings that Model N recommends.

A.1 Application-specific Settings

A.1.1 Performance Analytics

Security warnings may appear on each page of Performance Analytics unless the server is added as a trusted site on the Security Tab of the Internet Options dialog box from Internet Explorer. The host would be entered as `*.domain.com.`

Alternately, in local_common.properties file, for the com.modeln.AppSwitch.analytics.performance.qv.host and com.modeln.AppSwitch.analytics.performance.qv.server.domain property values, do not use a fully qualified host name. For example, instead of serverr1.domain.com use server1.

A.2 Internet Explorer 8

The following settings were set based on version 8.0.6001.18702 of Internet Explorer 8.

Recommendation: Do not use closing the browser window as a way to log out of the application. If multiple windows are sharing a session to the Model Nsystem, the session will not be cleared.

Default Security Zones Settings

- Internet: Medium
- Local Intranet: Medium-Low
- Trusted Sites: Custom
- Restricted Sites: Custom

Custom Security Zones

Table A-1: Settings for Internet Explorer 8 Custom Security Zones

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
.NET Framework				
Loose XAML	Enable	Enable	Enable	Disable
XAML browser applications	Enable	Enable	Enable	Disable
XPS documents	Enable	Enable	Enable	Disable
.NET Framework-reliant components				
Run components not signed with Authenticode	Enable	Enable	Enable	Disable
Run components signed with Authenticode	Enable	Enable	Enable	Disable
Permissions for components with manifests	High Safety	High Safety	High Safety	Disable
ActiveX controls and plug-ins				
Allow Previously unused ActiveX controls to run without prompt	Enable	Enable	Enable	Disable
Only allow approved domains to use ActiveX without prompt	Enable	Disable	Disable	Enable
Allow Scriptlets	Disable	Enable	Disable	Disable
Automatic prompting for ActiveX controls	Disable	Enable	Disable	Disable

Table A-1: Settings for Internet Explorer 8 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
Binary and script behaviors	Enable	Enable	Enable	Disable
Display video and animation on a webpage that does not use external video player	Disable	Disable	Disable	Disable
Download signed ActiveX controls	Prompt	Prompt	Enable	Disable
Download unsigned ActiveX controls	Disable	Disable	Prompt	Disable
Initialize and script ActiveX controls not marked as safe for scripting	Disable	Disable	Disable	Disable
Run ActiveX controls and plug-ins	Enable	Enable	Enable	Disable
Script ActiveX controls marked safe for scripting	Enable	Enable	Enable	Disable
Downloads				
Automatic prompting for file downloads	Disable	Enable	Disable	Disable
File download	Enable	Enable	Enable	Disable
Font download	Enable	Enable	Enable	Prompt
Enable .NET Framework setup	Enable	Enable	Enable	Disable
Java VM				
Java Permissions	High Safety	Medium Safety	High Safety	Disable Java
Miscellaneous				
Access data sources across domains	Disable	Prompt	Disable	Disable
Allow META REFRESH	Enable	Enable	Enable	Disable
Allow scripting of Internet Explorer web browser control	Disable	Enable	Disable	Disable
Allow script-initiated windows without size or position constraints	Disable	Enable	Disable	Disable

Table A-1: Settings for Internet Explorer 8 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
Allow webpages to use restricted protocols for active content	Prompt	Prompt	Prompt	Disable
Allow websites to open windows without address or status bar	Enable	Enable	Enable	Disable
Display mixed content	Prompt	Prompt	Prompt	Prompt
Don't prompt for client certificate selection when no cert. or only one cert. exists	Disable	Enable	Disable	Disable
Drag and drop or copy and paste files	Enable	Enable	Enable	Prompt
Include local directory path when uploading files to a server	Enable	Enable	Enable	Disable
Installation of desktop items	Prompt	Prompt	Prompt	Disable
Launching Launching applications and unsafe files	Prompt	Enable	Prompt	Disable
Launching programs and files in an IFRAME	Prompt	Prompt	Prompt	Disable
Navigate sub-frames across different domains	Disable	Enable	Disable	Disable
Open files based on content, not file extension	Enable	Enable	Enable	Disable
Submit nonencrypted form data	Prompt	Enable	Enable	Prompt
Use Pop-up Blocker	Enable	Disable	Enable	Enable
Use SmartScreen Filter	Enable	Disable	Enable	Enable
Userdata persistence	Enable	Enable	Enable	Disable
Web sites in less privileged web content zone can navigate into this zone	Enable	Enable	Prompt	Disable
Scripting				
Active scripting	Enable	Enable	Enable	Disable

Table A-1: Settings for Internet Explorer 8 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
Allow programmatic clipboard access	Prompt	Enable	Prompt	Disable
Allow status bar updates via script	Enable	Enable	Enable	Disable
Allow websites to prompt for information using scripted windows	Enable	Enable	Enable	Disable
Scripting of Java applets	Enable	Enable	Enable	Disable
Enable XSS Filter	Enable	Disable	Enable	Enable
User Authentication				
Logon				
Anonymous logon	Unchecked	Unchecked	Unchecked	Unchecked
Automatic logon only in Intranet zone	Checked	Checked	Checked	Unchecked
Automatic logon with current username and password	Unchecked	Unchecked	Unchecked	Unchecked
Prompt for user name and password	Unchecked	Unchecked	Unchecked	Checked

Advanced Tab

Accessibility

- Always expand ALT text for images: Checked
- Move system caret with focus/selection changes: Unchecked
- Reset text size to medium for new windows and tabs: Unchecked
- Reset text size to medium while zooming: Checked
- Reset zoom level to 100% for new windows and tabs: Checked
- Enable Caret Browsing for new windows and tabs: Unchecked

Browsing

- Automatically check for Internet Explorer updates: Unchecked
- Automatically recover from page layout errors with Compatibility View: Checked
- Close unused folders in History and Favorites: Unchecked
- Disable script debugging (Internet Explorer): Unchecked
- Disable script debugging (Other): Unchecked
- Display a notification about every script error: Unchecked
- Display Accelarator button on selection: Unchecked
- Enable folder view for FTP sites: Checked
- Enable page transitions: Checked
- Enable third-party browser extensions: Checked
- Enable visual styles on buttons and controls in web pages: Checked
- Enable websites to use the search pane: Unchecked
- Enable Suggested Sites: Unchecked
- Force offscreen compositing even under Terminal Services: Unchecked
- Notify when downloads complete: Checked
- Reuse windows for launching shortcuts: Checked
- Show friendly HTTP error messages: Checked
- Underline links: Always
- Use inline AutoComplete: Unchecked
- Use most recent order when switching tabs with Ctrl + Tab: Unchecked
- Use Passive FTP (for firewall and DSL modem compatibility): Checked
- Use smooth scrolling: Checked

Java (Sun)

- Use JRE <version> for <applet>: Checked

Microsoft VM

- Java console enabled: Unchecked
- Java logging enabled: Unchecked
- JIT compiler for virtual machine enabled: Checked

HTTP 1.1 Settings

- Use HTTP 1.1: Checked
- Use HTTP 1.1 through proxy connections: Checked

International

- Always show encoded addresses: Unchecked
- Send IDN server names: Checked
- Send IDN server names for Intranet addresses: Unchecked
- Send UTF-8 URLs: Checked
- Show Information Bar for encoded addresses: Checked
- Use UTF-8 for mailto links: Unchecked

Multimedia

- Always use ClearType for HTML: Checked
- Enable Automatic Image Resizing: Checked
- Play animations in web pages: Checked
- Play sounds in web pages: Checked
- Show image download placeholders: Unchecked
- Show pictures: Checked
- Smart image dithering: Checked

Printing

- Print background colors and images: Unchecked

Search from the Address Bar

- Do not submit unknown addresses to your auto-search provider: Unchecked
- Just display the results in the main window: Checked

Security

- Allow active content from CDs to run on My Computer: Unchecked
- Allow active content to run in files on My Computer: Unchecked
- Allow software to run or install even if the signature is invalid: Unchecked
- Check for publisher's certificate revocation: Checked
- Check for server certificate revocation: Unchecked
- Check for signatures on downloaded programs: Checked
- Do not save encrypted pages to disk: Unchecked
- Empty Temporary Internet Files folder when browser is closed: Unchecked
- Enable Integrated Windows Authentication: Checked
- Enable native XMLHTTP support: Checked
- Enable DOM Storage: Checked
- Enable memory protection to help mitigate online attacks: Checked
- Enable Smartscreen Filter: Unchecked
- Use SSL 2.0: Unchecked
- Use SSL 3.0: Checked
- Use TLS 1.0: Checked
- Warn about certificate address mismatch: Checked
- Warn if changing between secure and not secure mode: Unchecked
- Warn if POST submittal is redirected to a zone that does not permit posts: Checked

A.3 Internet Explorer 7

The following settings were set based on version 7.0.5730.11 of Internet Explorer 7.

Default Security Zones Settings

- Internet: Medium
- Local Intranet: Medium-Low
- Trusted Sites: Custom
- Restricted Sites: Custom

Custom Security Zones

Table A-2: Settings for Internet Explorer 7 Custom Security Zones

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
.NET Framework				

Table A-2: Settings for Internet Explorer 7 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
Loose XAML	Enable	Enable	Enable	Disable
XAML browser applications	Enable	Enable	Enable	Disable
XPS documents	Enable	Enable	Enable	Disable
.NET Framework-reliant components				
Run components not signed with Authenticode	Enable	Enable	Enable	Disable
Run components signed with Authenticode	Enable	Enable	Enable	Disable
ActiveX controls and plug-ins				
Allow Previously unused ActiveX controls to run without prompt	Enable	Enable	Enable	Disable
Allow Scriptlets	Disable	Enable	Disable	Disable
Automatic prompting for ActiveX controls	Disable	Enable	Disable	Disable
Binary and script behaviors	Enable	Enable	Enable	Disable
Display video and animation on a webpage that does not use external video player	Disable	Disable	Disable	Disable
Download signed ActiveX controls	Prompt	Prompt	Enable	Disable
Download unsigned ActiveX controls	Disable	Disable	Prompt	Disable
Initialize and script ActiveX controls not marked as safe for scripting	Disable	Disable	Disable	Disable
Run ActiveX controls and plug-ins	Enable	Enable	Enable	Disable
Script ActiveX controls marked safe for scripting	Enable	Enable	Enable	Disable
Downloads				
Automatic prompting for file downloads	Disable	Enable	Disable	Disable

Table A-2: Settings for Internet Explorer 7 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
File download	Enable	Enable	Enable	Disable
Font download	Enable	Enable	Enable	Prompt
Enable .NET Framework setup				
	Enable	Enable	Enable	Disable
Miscellaneous				
Access data sources across domains	Disable	Prompt	Disable	Disable
Allow META REFRESH	Enable	Enable	Enable	Disable
Allow scripting of Internet Explorer web browser control	Disable	Enable	Disable	Disable
Allow script-initiated windows without size or position constraints	Disable	Enable	Disable	Disable
Allow webpages to use restricted protocols for active content	Prompt	Prompt	Prompt	Disable
Allow websites to open windows without address or status bar	Enable	Enable	Enable	Disable
Display mixed content	Prompt	Prompt	Prompt	Prompt
Don't prompt for client certificate selection when no cert. or only one cert. exists	Disable	Enable	Disable	Disable
Drag and drop or copy and paste files	Enable	Enable	Enable	Prompt
Include local directory path when uploading files to a server	Enable	Enable	Enable	Disable
Installation of desktop items	Prompt	Prompt	Prompt	Disable
Launching Launching applications and unsafe files	Prompt	Enable	Prompt	Disable
Launching programs and files in an IFRAME	Prompt	Prompt	Prompt	Disable
Navigate sub-frames across different domains	Disable	Enable	Disable	Disable

Table A-2: Settings for Internet Explorer 7 Custom Security Zones (Continued)

Security Zone Settings	Internet	Local Intranet	Trusted	Restricted
Open files based on content, not file extension	Enable	Enable	Enable	Disable
Software channel permissions	Medium	Medium	Medium	High
Submit nonencrypted form data	Prompt	Enable	Enable	Prompt
Use Phishing Filter	Enable	Disable	Enable	Enable
Use Pop-up Blocker	Enable	Disable	Enable	Enable
Userdata persistence	Enable	Enable	Enable	Disable
Web sites in less privileged web content zone can navigate into this zone	Enable	Enable	Prompt	Disable
Scripting				
Active scripting	Enable	Enable	Enable	Disable
Allow programmatic clipboard access	Prompt	Enable	Prompt	Disable
Allow status bar updates via script	Enable	Enable	Enable	Disable
Allow websites to prompt for information using scripted windows	Enable	Enable	Enable	Disable
Scripting of Java applets	Enable	Enable	Enable	Disable
User Authentication				
Logon				
Anonymous logon	Unchecked	Unchecked	Unchecked	Unchecked
Automatic logon only in Intranet zone	Checked	Checked	Checked	Unchecked
Automatic logon with current username and password	Unchecked	Unchecked	Unchecked	Unchecked
Prompt for user name and password	Unchecked	Unchecked	Unchecked	Checked

Advanced Tab

Accessibility

- Always expand ALT text for images: Checked
- Move system caret with focus/selection changes: Unchecked
- Reset text size to medium for new windows and tabs: Unchecked
- Reset text size to medium while zooming: Checked
- Reset zoom level to 100% for new windows and tabs: Checked

Browsing

- Automatically check for Internet Explorer updates: Unchecked
- Close unused folders in History and Favorites: Unchecked
- Disable script debugging (Internet Explorer): Unchecked
- Disable script debugging (Other): Unchecked
- Display a notification about every script error: Unchecked
- Enable folder view for FTP sites: Checked
- Enable Install On Demand (Internet Explorer): Unchecked
- Enable page transitions: Checked
- Enable Personalized Favorites Menu: Unchecked
- Enable third-party browser extensions: Checked
- Enable visual styles on buttons and controls in web pages: Checked
- Enable websites to use the search pane: Unchecked
- Force offscreen compositing even under Terminal Services: Unchecked
- Notify when downloads complete: Checked
- Reuse windows for launching shortcuts: Checked
- Show friendly HTTP error messages: Checked
- Underline links: Always
- Use inline AutoComplete: Unchecked
- Use most recent order when switching tabs with Ctrl + Tab: Unchecked
- Use Passive FTP (for firewall and DSL modem compatibility): Checked
- Use smooth scrolling: Checked

HTTP 1.1 Settings

- Use HTTP 1.1: Checked
- Use HTTP 1.1 through proxy connections: Checked

International

- Always show encoded addresses: Unchecked
- Send IDN server names: Checked
- Send IDN server names for Intranet addresses: Unchecked
- Send UTF-8 URLs: Checked
- Show Information Bar for encoded addresses: Checked
- Use UTF-8 for mailto links: Unchecked

Multimedia

- Always use ClearType for HTML: Checked
- Enable Automatic Image Resizing: Checked
- Play animations in web pages: Checked
- Play sounds in web pages: Checked
- Show image download placeholders: Unchecked
- Show pictures: Checked
- Smart image dithering: Checked

Printing

- Print background colors and images: Unchecked

Search from the Address Bar

- Do not search from the Address bar: Unchecked
- Just display the results in the main window: Checked

Security

- Allow active content from CDs to run on My Computer: Unchecked
- Allow active content to run in files on My Computer: Unchecked
- Allow software to run or install even if the signature is invalid: Unchecked
- Check for publisher's certificate revocation: Checked
- Check for server certificate revocation: Unchecked
- Check for signatures on downloaded programs: Checked
- Do not save encrypted pages to disk: Unchecked
- Empty Temporary Internet Files folder when browser is closed: Unchecked
- Enable Integrated Windows Authentication: Checked
- Enable native XMLHTTP support: Checked
- **Phishing Filter**
  - Disable Phishing Filter: Unchecked
  - Turn off automatic website checking: Checked
  - Turn on automatic website checking: Unchecked
- Use SSL 2.0: Unchecked
- Use SSL 3.0: Checked
- Use TLS 1.0: Checked
- Warn about certificate address mismatch: Checked
- Warn if changing between secure and not secure mode: Unchecked
- Warn if POST submittal is redirected to a zone that does not permit posts: Checked

B

Packaging XML

This appendix contains the following .xml files that are used by the Configuration Console:

- Base Configuration File
- Med Dev Configuration File
- Pharma Configuration File

Code B-1: Base Configuration File

```
<Configuration>
<!-- The base configuration file contains all features and feature
groups that are applicable to all verticals.-->
 <!-- Features-->
 <Feature Name="app">
  <Description>Base Features</Description>
  </Feature>
 <Feature Name="pricemaster">
  <Description>Price Master Feature</Description>
  <Dependency Name="app" />
  </Feature>
 <Feature Name="oqd">
  <Description>Order Quantity Discounts Feature</Description>
  <Dependency Name="app" />
```

Code B-1: Base Configuration File (Continued)

```
 </Feature>
<Feature Name="sales">
 <Description>Sales Submission Feature</Description>
 <Dependency Name="app" />
 </Feature>
<Feature Name="mt">
 <Description>Media Tracking Feature</Description>
 <Dependency Name="sales" />
 </Feature>
<Feature Name="compliance">
 <Description>Compliance Feature</Description>
 <Dependency Name="app" />
 </Feature>
<Feature Name="rebates">
 <Description>Rebates/Fees Feature</Description>
 <Dependency Name="app" />
 <Dependency Name="sales" />
 </Feature>
<Feature Name="distrebates">
 <Description>Distributor Rebates Features</Description>
 <Dependency Name="app" />
 <Dependency Name="sales" />
 </Feature>
<!-- Packages-->
<FeatureGroup Name="ContractsAndPricingPackage">
 <Description>Contracts and Pricing</Description>
 <Dependency Name="app" />
 <Dependency Name="pricemaster" />
 <Dependency Name="oqd" />
 </FeatureGroup>
<FeatureGroup Name="SalesPackage">
 <Description>Sales Submissions</Description>
 <Dependency Name="sales" />
 <Dependency Name="mt" />
 </FeatureGroup>
<FeatureGroup Name="RebatesPackage">
 <Description>Rebates/Fees</Description>
 <Dependency Name="rebates" />
 </FeatureGroup>
<FeatureGroup Name="CompliancePackage">
 <Description>Compliance</Description>
 <Dependency Name="compliance" />
 </FeatureGroup>
<FeatureGroup Name="DistributorRebatesPackage">
 <Description>Distributor Rebates</Description>
 <Dependency Name="distrebates" />
 </FeatureGroup>
 </Configuration>
```

Code B-2: Med Dev Configuration File

```
<Configuration>
<!-- The meddev configuration file contains all features and feature
groups that are applicable only to the Medical Device vertical. This
configuration file may reference features/groups in the base
configuration. -->
 <!-- Features -->
 <Feature Name="meddev">
  <Description>Medical Device Extensions</Description>
  <Dependency Name="app" />
  </Feature>
 <Feature Name="meddevrebates">
  <Description>Medical Device Rebates Extensions</Description>
  <Dependency Name="meddev" />
  <Dependency Name="rebates" />
  </Feature>
 <!-- Feature Group Overlays-->
 <FeatureGroupOverlay Name="ContractsAndPricingPackage">
  <Dependency Name="meddev" />
  </FeatureGroupOverlay>
 <FeatureGroupOverlay Name="RebatesPackage">
  <Dependency Name="meddevrebates" />
  </FeatureGroupOverlay>
 <!-- Deployable Packages -->
 <FeatureGroup Name="FullSuitePackage" External="true">
  <Description>Full Medical Device Application Suite</Description>
  <Dependency Name="ContractsAndPricingPackage" />
  <Dependency Name="SalesPackage" />
  <Dependency Name="RebatesPackage" />
  <Dependency Name="CompliancePackage" />
  <Dependency Name="DistributorRebatesPackage" />
  </FeatureGroup>
  </Configuration>
```

Code B-3: Pharma Configuration File

```
<Configuration>
 <!-- The pharma configuration file contains all features and feature
groups that are applicable only to the Pharmaceutical vertical. This
configuration file may reference features/groups in the base
configuration. -->
 <!-- Features -->
 <Feature Name="pharma">
  <Description>Pharmaceutical Extensions</Description>
  <Dependency Name="app" />
  </Feature>
 <Feature Name="usregpharma">
  <Description>US Pharmaceutical Regulatory Extensions</Description>
```

Code B-3: Pharma Configuration File (Continued)

```
  <Dependency Name="pharma" />
  </Feature>
 <Feature Name="gp">
  <Description>Government Pricing</Description>
  <Dependency Name="app" />
  <Dependency Name="pharma" />
  <Dependency Name="usregpharma" />
  </Feature>
 <Feature Name="gpsolo">
  <Description>Government Pricing Solo</Description>
  <Dependency Name="gp" />
  </Feature>
 <Feature Name="fsscompliance">
  <Description>FSS Compliance</Description>
  <Dependency Name="gp" />
  </Feature>
 <Feature Name="medicaid">
  <Description>Medicaid</Description>
  <Dependency Name="app" />
  <Dependency Name="pharma" />
  <Dependency Name="usregpharma" />
  </Feature>
 <Feature Name="medicaid_solo">
  <Description>Medicaid Solo</Description>
  <Dependency Name="medicaid" />
  </Feature>
 <Feature Name="pharmaba">
  <Description>Bid Awards</Description>
  <Dependency Name="pharma" />
  <Dependency Name="sales" />
  </Feature>
 <Feature Name="pharmarebates">
  <Description>Pharmaceutical Rebates Extensions</Description>
  <Dependency Name="pharma" />
  <Dependency Name="rebates" />
  </Feature>
 <Feature Name="pharmadistrebates">
  <Description>Pharmaceutical Distributor Rebates (Chargeback) 
Extensions</Description>
  <Dependency Name="sales" />
  <Dependency Name="pharma" />
  <Dependency Name="distrebates" />
  </Feature>
 <Feature Name="mco">
  <Description>Pharmaceutical Managed Care Application</Description>
  <Dependency Name="pharma" />
  <Dependency Name="pharmarebates" />
  </Feature>
```

Code B-3: Pharma Configuration File (Continued)

```
 <!-- Feature Group Overlays -->
 <FeatureGroupOverlay Name="ContractsAndPricingPackage">
  <Dependency Name="pharma" />
  <Dependency Name="pharmaba" />
  <Dependency Name="pharmadistrebates" />
  </FeatureGroupOverlay>
 <FeatureGroupOverlay Name="RebatesPackage">
  <Dependency Name="pharmarebates" />
  </FeatureGroupOverlay>
 <FeatureGroupOverlay Name="DistributorRebatesPackage">
  <Description>Chargebacks</Description>
  </FeatureGroupOverlay>
 <FeatureGroup Name="ManagedCarePackage">
  <Dependency Name="mco" />
  <Description>Managed Care</Description>
  </FeatureGroup>
 <!-- Deployable Packages -->
 <FeatureGroup Name="FullCommercialSuitePackage" External="false">
  <Description>Full Pharmaceutical Commercial Application Suite</
Description>
  <Dependency Name="ContractsAndPricingPackage" />
  <Dependency Name="SalesPackage" />
  <Dependency Name="RebatesPackage" />
  <Dependency Name="CompliancePackage" />
  <Dependency Name="DistributorRebatesPackage" />
  <Dependency Name="ManagedCarePackage" />
  </FeatureGroup>
 <FeatureGroup Name="ManagedCareOnlyPackage">
  <Dependency Name="mco" />
  </FeatureGroup>
 <FeatureGroup Name="GPOnlyPackage" External="false">
  <Description>Pharmaceutical Government Pricing Only</Description>
  <Dependency Name="gp" />
  <Dependency Name="gpsolo" />
  <Dependency Name="fsscompliance" />
  </FeatureGroup>
 <FeatureGroup Name="MedicaidOnlyPackage" External="false">
  <Description>Pharmaceutical Medicaid Only</Description>
  <Dependency Name="medicaid_solo" />
  </FeatureGroup>
 <FeatureGroup Name="FullRegulatorySuitePackage" External="false">
  <Description>Full Pharmaceutical Regulatory Application Suite</
Description>
  <Dependency Name="GPOnlyPackage" />
  <Dependency Name="MedicaidOnlyPackage" />
  </FeatureGroup>
 <FeatureGroup Name="FullSuitePackage" External="true">
  <Description>Full Pharmaceutical Application Suite</Description>
  <Dependency Name="FullCommercialSuitePackage" />
```

Code B-3: Pharma Configuration File (Continued)

```
<Dependency Name="gp" />
<Dependency Name="medicaid" />
<Dependency Name="fsscompliance" />
</FeatureGroup>
</Configuration>
```

C Installation Verification Checklists

This appendix provides information to help verify that your installation of the Model N application suite was successful.

Note: For information on the Installation Verification page available within the Administration menu, see the *Application Administrator's Guide*.

C.1 Steps for Verification

To verify your installation:

1. Populate or import the application database.

   Review bootstrap and population output to ensure now errors were reported

2. Deploy the Model N application
   a. Review the application server deployment logs to ensure there were no errors or exceptions.
   b. Review the Model N application logs to ensure there were no errors or exceptions.

   c. Check the copyright page to ensure that the correct Model N version number is displayed.
   d. Log in to the Model N application to perform a smoke test.
   e. Review the app server and Model N logs to ensure no exceptions were thrown during smoke testing.

3. Deploy the Cognos Reports (optional)
   a. Review the output of the Ops tool to ensure no errors were displayed during report template deployment.
   b. Log in to the Model N application and run sample reports.
   - SQL-based report verification (Analyst Workload report for example)
   - Model-based report verification (Contrct/Offer report for example)
   - PMA report verification to ensure both PMA is working and reports are working
4. Deploy the RPI application (optional – Pharma only)

   Log in to the RPI application and perform a quick smoke test.

On a clustered environment it's a little more complicated, but that only applies to IBM WebSphere customers.

C.1.1 Verification Checklist for New Installs

C.1.1.1 Database Server

Database

☐ **Database restarted.**

☐ **Performance data of previous cycle saved.**

Model N Database Deployment

☐ **Model N schema was refreshed and data was imported.**

☐ **Cognos schema was refreshed**

C.1.1.2 Application Server

General

☐ **Application deployed with logs exists**

☐ **Able to navigate within the Model N application**

☐ **Cognos reports run successfully**

   ☐ **SQL-based reports (Analyst Workload)**

   ☐ **Model-based reports (Contract/ Offer)**

   ☐ **PMA reports (PMA- Overview)**

- ☐ Admin scripts available
- ☐ Tool scripts available
- ☐ Data content available
- ☐ RPI Works (Pharma installations only)

Cognos

- ☐ Configurations are up-to-date

Model N Deployment

- ☐

C.1.2 Verification Checklist for Upgrades

C.1.2.1 Database Server

Database

- ☐ Database restarted.
- ☐ Performance data of previous cycle saved.

Model N Database Deployment

- ☐ Model N schema was refreshed and data was imported.
- ☐ Cognos schema was refreshed

C.1.2.2 Application Server

General

- ☐ Application deployed with logs exists
- ☐ Able to navigate within the Model N application
- ☐ Cognos reports run successfully
  - ☐ SQL-based reports (Analyst Workload)
  - ☐ Model-based reports (Contract/ Offer)
  - ☐ PMA reports (PMA- Overview)
- ☐ Admin scripts available
- ☐ Tool scripts available
- ☐ Data content available
- ☐ RPI Works (Pharma installations only)

Cognos

- ☐ Configurations are up-to-date

Model N Deployment

- ☐ **Application refreshed without error**
- ☐ **dbsync completed without error**
- ☐ **Cognos refreshed and restarted**
- ☐ **Cognos template deployment complete without error**
- ☐ **Server log has no errors**

D Revisions

This appendix covers the revisions made to this guide based on changes in the Model N application suite, or expansion or inclusion of the content. It does not address editorial changes.

D.1 Changes in Release 5.5.1

The following changes have been made to this guide for the 5.5.1 release:

- The direct migration paths from prior Model N releases to 5.5.1 have been added and previous migrations have been removed. See Supported Migrations for more information.
- Support for deployment under JBoss has been added. See Deploying with Red Hat JBoss Enterprise Application Platform for more information.
- Support for Oracle 10g is not supported in this release and has been removed from the documentation.
- The recommended initial settings for Oracle have been updated. See Initialization File on page 48 for information on the current recommendations.
- The value for the LANG variable under WebSpere has been updated to en_US.UTF-8 as noted in Administering WebSphere on page 109.

D.2 Changes in Release 5.5

The following changes have been made to this guide for the 5.5 release:

- The direct migration paths from prior Model N releases to 5.5 have been added and previous migrations have been removed. See Supported Migrations for more information.
- References to realm have been removed as realm functionality is now deprecated.
- Some values to the initial Oracle database settings have been updated. See Initialization File on page 48 for more information.
- To support the export and display of multi-byte characters in Windows applications such as Microsoft Excel, the JVM argument `-Dfile.encoding=UTF-8` was added for both WebSphere and WebLogic.

Index