EN-NODE

INSTALLATION GUIDE

(JAVA VERSION)

Version: 1.3

October 2, 2012

enfoTech

1368 How Lane
North Brunswick, New Jersey 08902
www.enfotech.com

Revision History

Version	Date	Created By	Reviewed By	Description
1.3	10/02/2012	Charlie Tsai	Charlie Tsai	Updated to company contact information

Tables of Contents

1 Purpose of Document

This document was created to provide instructions for installing and configuring the EN-Node software. EN-Node was built for the National Environmental Information Exchange Network (NEIEN) for the exchange of environmental and health data between State, Tribal, and Federal Environmental agencies.

This document covers the following topics:

- **Hardware/Software Requirements:** identifies the hardware and software needed to run EN-Node
- **Deployment Options:** Identifies different strategies for deploying the EN-Node software within your organization's existing network and security topology
- **Installation Instructions:** Provides step-by-step instructions for installing the EN-Node software
- **Initial Application Configuration:** Provides step-by-step instructions for Node configuration that must be performed after initially installing the Node
- **Installation Confirmation:** Provides step-by-step instructions on how to confirm that the Node is successfully installed and ready to communicate with other Exchange Network partners.

2 EN-Node System Overview

The EN-Node consists of the several sub-applications, each of which is described briefly below:

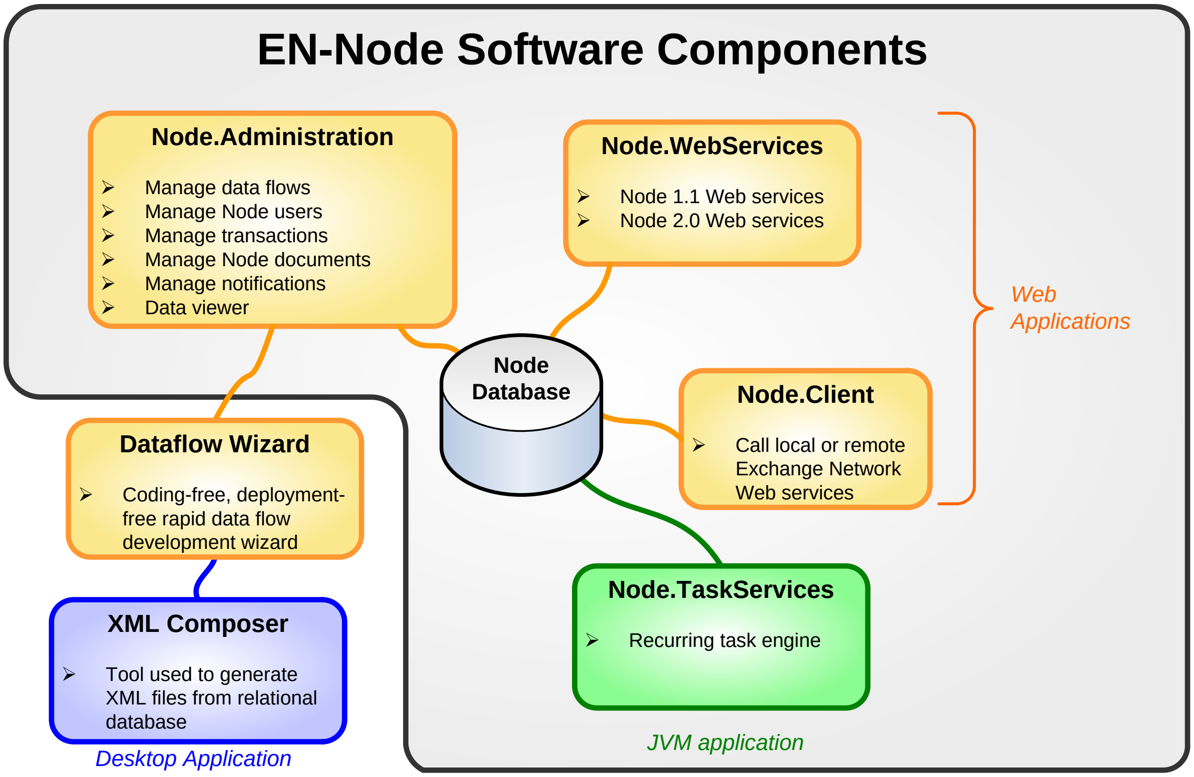


Figure 1: EN-Node Components

- **Node.Administration:** A web-based graphical interface that allows Node and Data Flow administrators to configure the Node and manage data flows. The Node.Administration Console allows administrators to: create and maintain dataflow plug-ins, manage security access to dataflows, review node activity logs, initiate inbound or outbound data transfers (either on-demand or recurring schedules), and browse inbound or outbound data (in raw or parsed format). This is the primary application that the Node's host agency will interact with.
- **Node.WebServices:** The core Web Services engine that controls the logic for responding to Web Service requests on the Node, providing the web services outlined in the Exchange Network Node 1.1 and 2.0 Specifications. When responding to a Web service request, Node.WebServices will execute logic plugged in for a particular data flow.
- **Node.Client:** A simple Web interface that allows individuals to invoke Node 1.1 or 2.0 Web Services on any Node, including your own node. This application can be useful for either testing your Node functionality, or can serve as a simple Node client to invoke Web services on other Nodes.
- **Node.TaskServices:** Provides the capability to execute tasks on a scheduled basis, which allows you to schedule and initiate Web service exchanges. These scheduled tasks typically involve the invocation of Web Services on other Nodes, such as EPA's Node. The scheduled tasks are defined by the task plug-in and are configured by a Dataflow Administrator for a particular data flow.

3 Hardware / Software Requirements

This section identifies the hardware and software required to run EN-Node. The specific requirements for memory, disk space, and load balancing are simply guidelines, as the exact requirements may vary widely depending on a number of factors such as:

- The number of data flows expected to be supported
- The volume, frequency, and/or complexity of data processing
- Server uptime requirements (e.g. 24/7)
- Mission critical nature of node data flows
- Whether Node is dedicated or shares server resources
- Network latency

With those caveats, here are the hardware requirements and recommendations:

Web / App Server(s):

Table 1: Web/Application Server Hardware Requirements and Recommendations

	Minimum Requirements	Recommended System
Load Balancing	Not Required	Network Load Balancing Recommended
Processor	Pentium IV 2.4 GHz or higher	Pentium IV 2.8 GHz or higher
Memory	1 GB of RAM	2 to 4 GB of RAM or higher
Disk Space	20 GB free hard disk space	40 GB free hard disk space or higher

Node Database Server:

Table 2: Node Database Server Hardware Requirements and Recommendations

	Minimum Requirements	Recommended System
Cluster Server	Not Required	Cluster Server with RAID 5 Storage Array
Processor	Pentium IV 2.4 GHz or higher	Pentium IV 2.8 GHz or higher
Memory	1 GB of RAM	4 GB of RAM or higher
Disk Space	40 GB free hard disk space	80 GB free hard disk space or higher

Server Software / Operating System Recommendations:

- **Operating System:**
  - Windows Server 2003 R2 or 2008 (64 bit editions are supported) or Sun Solaris 10
- **Java Environment:**
  - Java Runtime Environment (JRE) version of 1.4.2 (Need to compile source code) or 5.0 (The default deployment package is compiled by 1.5.0_18).
- **Web / Application Server:**
  - Apache Tomcat 5.0.x / 5.5.x
  - JBoss 4.2.3
  - Sun ONE Web Server 6.1
  - The server JVM memory should be more than: "-Xms256m -Xmx512m -XX:PermSize=256m -XX:MaxPermSize=512m"
  - Server Certificate/SSL 128 bit encryption (Optional)
- **Database Server:**
  - Oracle Database Server 9i or higher

The following diagram shows a general deployment diagram:

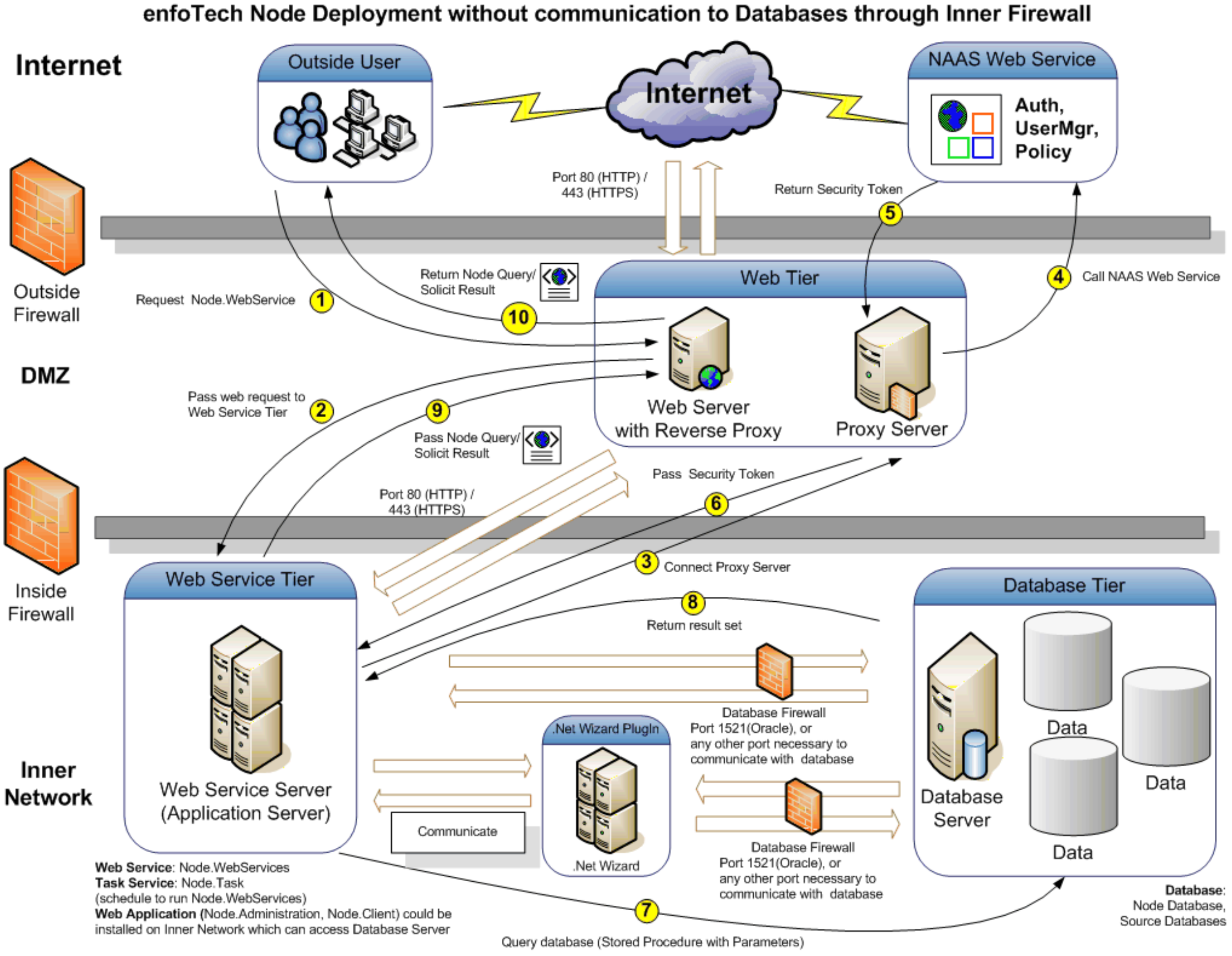

enfoTech Node Deployment without communication to Databases through Inner Firewall
Internet
Outside User
Internet
NAAS Web Service
Auth, UserMgr, Policy
Outside Firewall
DMZ
Port 80 (HTTP) / 443 (HTTPS)
Return Security Token
5
Return Node Query/ Solicit Result
Request Node.WebService
1
10
Web Tier
4
Call NAAS Web Service
Web Server with Reverse Proxy
Proxy Server
Pass web request to Web Service Tier
2
9
Pass Node Query/ Solicit Result
Pass Security Token
Port 80 (HTTP) / 443 (HTTPS)
6
Inside Firewall
3
Connect Proxy Server
Web Service Tier
Database Tier
8
Return result set
Database Firewall
Port 1521(Oracle), or any other port necessary to communicate with database
.Net Wizard PlugIn
Inner Network
Web Service Server (Application Server)
Communicate
.Net Wizard
Database Firewall
Port 1521(Oracle), or any other port necessary to communicate with database
Database Server
Data
Data
Data
Web Service: Node.WebServices
Task Service: Node.Task
(schedule to run Node.WebServices)
Web Application (Node.Administration, Node.Client) could be installed on Inner Network which can access Database Server
7
Query database (Stored Procedure with Parameters)
Database: Node Database, Source Databases

Figure 2: Generic System Architecture

4 EN-Node System Installation Instructions

Note: these installation instructions assume that the reader is experienced in deploying web applications on application servers and also experienced in managing databases in Oracle.

Note: Although the EN-Node architecture is designed for deploying the applications across multiple physical machines in an Enterprise environment, for clarity of this document, this guideline assume that all modules will be installed on a single machine.

4.1 Preparation for Software Installation

The following are conditions that need to be checked before deployment of the EN-Node:

- The server on which the EN-Node is deployed is accessible via the public Internet over a defined URL, which either has a registered domain name address, or is defined by an IP address and port number.
- The Node Server has access to the data source(s) that houses any data to be accessed by the Node.
- The Node should be hosted in a Demilitarized Zone (DMZ) or secured layer.
- It is strongly recommended to use 128-bit Security Socket Layer (SSL) on the web server.

4.2 Deployment Structure

- Application modules contain:
  - Node.Administratrion.war
  - Node.Client.war
  - Node.WebServices.war
  - Node.Task.war
- Database scripts contain:
  - NodeTablespaces.sql
  - eNode2008.sql
  - eNode2008Upgrade.sql
- Documents contain:
  - EN-Node Installation Guide.doc
  - EN-Node Administration Guide.doc

4.3 Node Database Installation

4.3.1 New Node Database Installation

Note: This section is for initial Node database installation only. For upgrading from enfoTech e-Node1.1 please follow instructions in the following section.

1. Log into Oracle as a user with sufficient rights to execute the supplies scripts. This will require the following rights: CREATE USER, CREATE TABLESPACE, CREATE TABLE and GRANT privileges.
2. Execute the following sql statements to create the Oracle tablespace NODESPACE:

```
CREATE TABLESPACE NODESPACE
  DATAFILE '/oradata/DEP/NODESPACE.dbf' SIZE 3000M REUSE
  AUTOEXTEND
    ON NEXT 1040K MAXSIZE  32767M EXTENT MANAGEMENT LOCAL;
```

Note: The datafile directory and size value can be changed based on the actual system requirements designated by User. The values provided above are not mandatory.

3. Execute the following Oracle sql statements to create the User "NODE":

```
CREATE TABLESPACE NODESPACE

   DATAFILE '/oradata/ DEP/NODESPACE.dbf' SIZE 3000M REUSE

   AUTOEXTEND

ON NEXT  1040K MAXSIZE  32767M EXTENT MANAGEMENT LOCAL;

CREATE USER NODE IDENTIFIED BY NODE123

DEFAULT TABLESPACE NODESPACE

TEMPORARY TABLESPACE TEMP

QUOTA UNLIMITED ON NODESPACE;

GRANT CONNECT TO NODE;

GRANT RESOURCE TO NODE;
```

   Note: The password for the Node user does not need to be "Node123". It can follow the existing standard or convention of User's IT environment. This information, however, will be needed in the Node application for database connection. The above SQL Statements for item 2 & 3 are prepared in the script, "**NodeTablespaces.sql**" provided by enfoTech as part of the installation files.

4. Execute the script eNODE2008.sql using SQL*PLUS to create the tables and insert initial data.

4.3.2 Upgrade from enfoTech Node 1.1 Database

Note: Before upgrading the database, please backup all existing database structures and data.

1. Login to Node Database as administrator or same level user, run "eNODE2008Upgrade.sql" to change table structure and create new tables.

4.4 Node Application Installation

Note: This section is for initial Node application installation only. For information on changes to the Node deploy files from Node1.1, please see section 5.

1. **Stop the Java application server** and (if this isn't the first time you are deploying an enfoTech Node), clean all old Node application directories and temporary work directory. This will ensure a clean install of the application.

2. **Deploy 4 WAR files:** Copy the 4 war files (Node.Administration, Node.Client, Node.TaskService, and Node.WebServices) into the deploy directory of the application server. For example:

   - Tomcat: Tomcat 5.0\webapps
   - JBoss: jboss-4.2.3.GA\server\default\deploy

   So for Tomcat 5.0, Steps 1-2 will be like this:

   a. Delete four application directories (Node.Administration, Node.Task,Node.Client, Node.WebServices) under “Tomcat 5.0\webapps”.

   b. Delete four temporary application directories (Node.Administration, Node.Task,Node.Client, Node.WebServices) under “Tomcat 5.0\ \work\Catalina\localhost”.

   c. Drop four new war files into deploy folder “Tomcat 5.0\webapps” of Tomcat server.

3. Create node system log directories, for example : c:\NodeLog

4. **Update Configuration Files:** Settings in several configuration files will need to be established as shown in the table starting on the next page.

   - Note: if you are upgrading from a previous enfoTech Node (1.1), make sure you copy in any previous dataflow specific entries, in addition to those listed in the table below.

5. **Start the Java application server:** Once all configuration settings have been setup, starts the server.

6. **Confirm application has started:** Navigate to http://yourserver:yourport/Node.Administration/Page/Entry/Login.do in your browser to confirm that the EN-Node core system has properly started.

File and Path	Section in File	Instructions

Node.Administration
\WEB-INF\Web.xml
	1.
	<env-entry>
	    <env-entry-name>node/dbFullString</env-entry-name>
	    <env-entry-value>
	    (description=(address_list=(address=(host=yourdbserver)(protocol=tcp)(port=1521))(address=(host= yourdbserver)(protocol=tcp)(port=1521)))(source_route=yes)(connect_data=(sid=yoursidname)))
	    </env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbUser</env-entry-name>
	    <env-entry-value>enode</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbPassword</env-entry-name>
	    <env-entry-value>enode</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	
	2.
	<env-entry>
	    <env-entry-name>node/dbServer</env-entry-name>
	    <env-entry-value> yourdbserver </env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbPort</env-entry-name>
	    <env-entry-value>1521</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbSID</env-entry-name>
	    <env-entry-value>yoursidname</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
		There are three ways to set up database connection.
		1. Use node/dbFullString element to set up fully connection string. (See red statement.)
		2. Use several connection elements to build a connection. (See blue statement.)
		3. Use JNDI connection string (See black statement.)
		4. node/dbType (See green statement) is necessary for all connection ways. For oracle9i please fill: “oracle9i”, for oracle10g or higher please fill “oracle10i”

File and Path	Section in File	Instructions
	<env-entry-name>node/dbUser</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbPassword</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 3. <resource-ref> <description>Oracle Development Datasource</description> <res-ref-name>jdbc/Node</res-ref-name> <res-type>javax.sql.DataSource</res-type> <res-auth>Container</res-auth> </resource-ref> <env-entry> <env-entry-name>node/dbJNDIName</env-entry-name> <env-entry-value>jdbc/Node</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 4. <env-entry> <env-entry-name>node/dbType</env-entry-name> <env-entry-value>oracle9i</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	
Node.Administration \WEB-INF \applicationContext.xml	1. <bean id="myDataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource"> <property name="driverClassName" value="oracle.jdbc.driver.OracleDriver" /> <property name="url" value="jdbc:oracle:thin://@yourserver:1521:yoursid"/> <property name="username" value="enode" /> <property name="password" value="enode" /> </bean>	Set database connection for Hibernate. Please change green statement to match local database. For JNDI set please change the red part to match the local database.

File and Path	Section in File	Instructions

	2.
	`<bean id="myDataSource"`
	`class="org.springframework.jndi.JndiObjectFactoryBean">`
	`        <property name="jndiName">`
	`            <value>java:comp/env/jdbc/yourdbname</value>`
	`        </property>`
	`</bean>`	

Node. Administration
\WEB-INF\Web.xml
	<env-entry>
	  <env-entry-name>http/PROTOCOL_VERSION</env-entry-name>
	  <env-entry-value>1.1</env-entry-value>
	  <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
		Add new protocol

Node.Administration
\WEB-INF\Web.xml
	`<env-entry>`
	`      <env-entry-name>adminLog</env-entry-name>`
	`      <env-entry-value>C:\YourDirectory\NodeLog </env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
		Configure log file output folder. This folder should be created first before you put here.

Node.Administration
\WEB-INF\Web.xml
	`<env-entry>`
	`      <env-entry-name>tempFilePath</env-entry-name>`
	`      <env-entry-value>C:\YourDirectory\NodeTemp</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
		Configure temporary file output folder. This folder should be created first before you put here.

Node.Administration
\WEB-INF\Web.xml
	`<env-entry>`
	`      <env-entry-name>sessionTimeOut</env-entry-name>`
	`      <env-entry-value>1200</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
		Changing this parameter will set the application session time out interval in seconds. Default is 1200 (i.e. 20 minutes).

Node.Administration
\WEB-INF\Web.xml
	`<env-entry>`
	`      <env-entry-name>node/dotNetHost</env-entry-name>`
	`      <env-entry-value>localhost</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`      <env-entry-name>node/dotNetHostPort</env-entry-name>`
	`      <env-entry-value>8080</env-entry-value>`
		Configure .Net Wizard PlugIn server Host name and port.

File and Path	Section in File	Instructions
	<env-entry-type>java.lang.String</env-entry-type> </env-entry>	
Node.Client \WEB-INF\Web.xml	1. <env-entry> <env-entry-name>node/dbFullString</env-entry-name> <env-entry-value> (description=(address_list=(address=(host=yourdbserver)(protocol=tcp)(port=1521))(address=(host= yourdbserver)(protocol=tcp)(port=1521)))(source_route=yes)(connect_data=(sid=yoursidname))) </env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbUser</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbPassword</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 2. <env-entry> <env-entry-name>node/dbServer</env-entry-name> <env-entry-value> yourdbserver </env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbPort</env-entry-name> <env-entry-value>1521</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbSID</env-entry-name> <env-entry-value>yoursidname</env-entry-value> <env-entry-type>java.lang.String</env-entry-type>	Database connection uses the same configuration like Node.Administration.

File and Path	Section in File	Instructions

	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbUser</env-entry-name>
	    <env-entry-value>enode</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	<env-entry>
	    <env-entry-name>node/dbPassword</env-entry-name>
	    <env-entry-value>enode</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
	3.
	 <resource-ref>
	   <description>Oracle Development Datasource</description>
	   <res-ref-name>jdbc/Node</res-ref-name>
	   <res-type>javax.sql.DataSource</res-type>
	   <res-auth>Container</res-auth>
	 </resource-ref>
	 <env-entry>
	    <env-entry-name>node/dbJNDIName</env-entry-name>
	    <env-entry-value>jdbc/Node</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	 </env-entry>
	
	4. <env-entry>
	        <env-entry-name>node/dbType</env-entry-name>
	        <env-entry-value>oracle9i</env-entry-value>
	        <env-entry-type>java.lang.String</env-entry-type>
	    </env-entry>	

Node.Client
\WEB-INF\Web.xml
	<env-entry>
	  <env-entry-name>clientLog</env-entry-name>
	  <env-entry-value> C:\YourDirectory\NodeLog </env-entry-value>
	  <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
		Set logging for Node.Client application (Same as Node.Administration above)

Node. Client
\WEB-INF\Web.xml
	<env-entry>
	    <env-entry-name>tempFilePath</env-entry-name>
	    <env-entry-value>C:\YourDirectory\NodeTemp</env-entry-value>
	    <env-entry-type>java.lang.String</env-entry-type>
	</env-entry>
		Configure temporary file output folder. (Same as Node.Administration above)

File and Path	Section in File	Instructions

Node.Task
\WEB-INF\Web.xml
	1.
	`<env-entry>`
	`      <env-entry-name>node/dbFullString</env-entry-name>`
	`      <env-entry-value>`
	`      (description=(address_list=(address=(host=yourdbserver)(protocol=tcp)(port=1521))(address=(host=yourdbserver )(protocol=tcp)(port=1521)))(source_route=yes)(connect_data=(sid=yoursidname)))`
	`      </env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`      <env-entry-name>node/dbUser</env-entry-name>`
	`      <env-entry-value>enode</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`      <env-entry-name>node/dbPassword</env-entry-name>`
	`      <env-entry-value>enode</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	
	2.
	`<env-entry>`
	`      <env-entry-name>node/dbServer</env-entry-name>`
	`      <env-entry-value> yourdbserver </env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`      <env-entry-name>node/dbPort</env-entry-name>`
	`      <env-entry-value>1521</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`      <env-entry-name>node/dbSID</env-entry-name>`
	`      <env-entry-value>yoursidname</env-entry-value>`
	`      <env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
		Database connection uses the same configuration like Node.Administration.

File and Path	Section in File	Instructions
	<env-entry-name>node/dbUser</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbPassword</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 3. <resource-ref> <description>Oracle Development Datasource</description> <res-ref-name>jdbc/Node</res-ref-name> <res-type>javax.sql.DataSource</res-type> <res-auth>Container</res-auth> </resource-ref> <env-entry> <env-entry-name>node/dbJNDIName</env-entry-name> <env-entry-value>jdbc/Node</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 4. <env-entry> <env-entry-name>node/dbType</env-entry-name> <env-entry-value>oracle9i</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	
Node.Task \WEB-INF\Web.xml	<env-entry> <env-entry-name>taskLog</env-entry-name> <env-entry-value> C:\YourDirectory\NodeLog </env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	Same like Node.Administration.
Node. Task \WEB-INF\Web.xml	<env-entry> <env-entry-name>tempFilePath</env-entry-name> <env-entry-value>C:\YourDirectory\NodeTemp</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	Configure temporary file output folder. (Same as Node.Administration above)

File and Path	Section in File	Instructions

Node.WebServices
\WEB-INF\Web.xml
	1.
	`<env-entry>`
	`<env-entry-name>node/dbFullString</env-entry-name>`
	`<env-entry-value>`
	`(description=(address_list=(address=(host=yourdbserver)(protocol=tcp)(port=1521))(address=(host= yourdbserver )(protocol=tcp)(port=1521)))(source_route=yes)(connect_data=(sid=yoursidname)))`
	`</env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`<env-entry-name>node/dbUser</env-entry-name>`
	`<env-entry-value>enode</env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`<env-entry-name>node/dbPassword</env-entry-name>`
	`<env-entry-value>enode</env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	
	2.
	`<env-entry>`
	`<env-entry-name>node/dbServer</env-entry-name>`
	`<env-entry-value> yourdbserver </env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`<env-entry-name>node/dbPort</env-entry-name>`
	`<env-entry-value>1521</env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
	`<env-entry-name>node/dbSID</env-entry-name>`
	`<env-entry-value>yoursidname</env-entry-value>`
	`<env-entry-type>java.lang.String</env-entry-type>`
	`</env-entry>`
	`<env-entry>`
		Database connection uses the same configuration like Node.Administration.

File and Path	Section in File	Instructions
	<env-entry-name>node/dbUser</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> <env-entry> <env-entry-name>node/dbPassword</env-entry-name> <env-entry-value>enode</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 3. <resource-ref> <description>Oracle Development Datasource</description> <res-ref-name>jdbc/Node</res-ref-name> <res-type>javax.sql.DataSource</res-type> <res-auth>Container</res-auth> </resource-ref> <env-entry> <env-entry-name>node/dbJNDIName</env-entry-name> <env-entry-value>jdbc/Node</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry> 4. <env-entry> <env-entry-name>node/dbType</env-entry-name> <env-entry-value>oracle9i</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	
Node.WebServices \WEB-INF\Web.xml	<env-entry> <env-entry-name>webServicesLog</env-entry-name> <env-entry-value> C:\YourDirectory\NodeLog</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	Same like Node.Administration.
Node. WebServices \WEB-INF\Web.xml	<env-entry> <env-entry-name>tempFilePath</env-entry-name> <env-entry-value>C:\YourDirectory\NodeTemp</env-entry-value> <env-entry-type>java.lang.String</env-entry-type> </env-entry>	Configure temporary file output folder. (Same as Node.Administration above)

File and Path	Section in File	Instructions
Node.WebServices \WEB-INF\services \NetworkNode2\META-INF\ NetworkNode2.wsdl	`<soap12:address location="http://yourhost:your port/Node.WebServices/services/NetworkNode2"/>`	The endpoint of webservice should be changed to match local host address. Do not change the “*/Node.WebServices/services /NetworkNode2”*

Node.WebServices
\WEB-INF\services
\NetworkNode2\META-INF\services.xml
	`<parameter name="useOriginalwsdl">true</parameter>`
	`<parameter name="modifyUserWSDLPortAddress">true</parameter>`
		Need to be changed if user want to generate different style of the endpoint of web service. We recommend use default value.
		“useOriginalwsdl”
		Is true means Axis2 will keep the original endpoint in wsdl, otherwise false means Axis2 will generate new endpoint automatically. Default is true
		“modifyUserWSDLPortAddress” is “true “means Axis2 will change the endpoint based on the server ip address and port. “false” forbid Axis2 change the wsdl. Default is “true”.

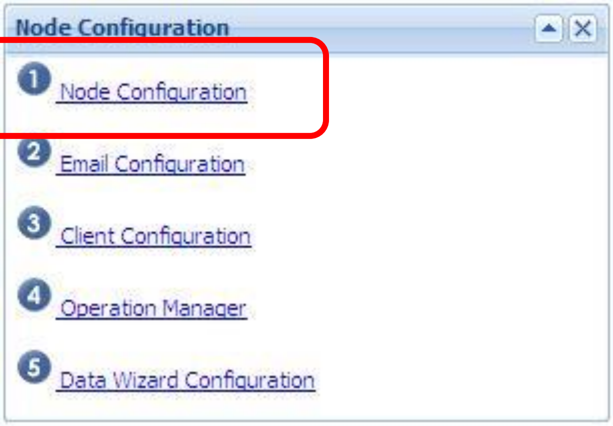
Node.WebServices
\WEB-INF \axis2.xml
	`<transportReceiver name="http"`
	`class="org.apache.axis2.transport.http.SimpleHTTPServer">`
	`<parameter name="port"> your port </parameter>`
		The port must match the endpoint address. For example:
		The endpoint is:” `http://localhost:8080/Node.WebServices/services/NetworkNode2"`
		This port is “8080”.

5 Additional Configuration If Upgrading from enfoTech Node 1.1

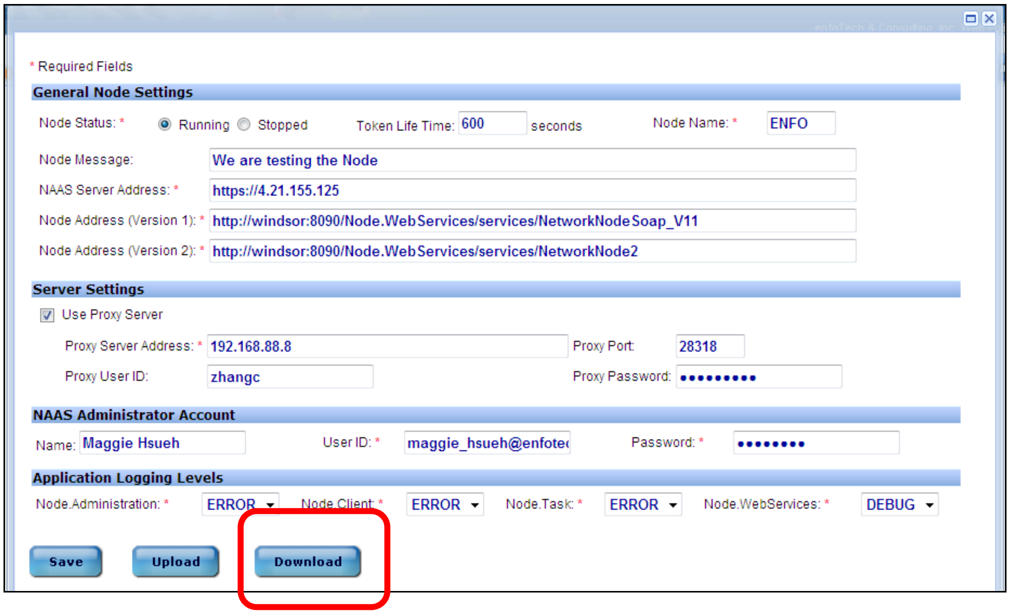
Note: skip this section if you are NOT upgrading from enfoTech Node 1.1.

Some special configuration updates need to be performed if you are upgrading from a previous enfoTech Node 1.1 installation. This section describes these upgrade steps:

1. Log into the Node Administration Console and click on the Node Configuration Link in the Node Configuration webpart, as shown here:



2. Click the **Download** button to download the current Node configuration settings file, as shown here:



* Required Fields

General Node Settings

Node Status: * ● Running ○ Stopped Token Life Time: 600 seconds Node Name: * ENFO

Node Message: We are testing the Node

NAAS Server Address: * https://4.21.155.125

Node Address (Version 1): * http://windsor:8090/Node.WebServices/services/NetworkNodeSoap_V11

Node Address (Version 2): * http://windsor:8090/Node.WebServices/services/NetworkNode2

Server Settings

☑ Use Proxy Server

Proxy Server Address: * 192.168.88.8 Proxy Port: 28318

Proxy User ID: zhangc Proxy Password: ●●●●●●●●●

NAAS Administrator Account

Name: Maggie Hsueh User ID: * maggie_hsueh@enfotec Password: * ●●●●●●●●

Application Logging Levels

Node.Administration: * ERROR Node.Client: * ERROR Node.Task: * ERROR Node.WebServices: * DEBUG

3. Use a text editor to add the following sections.
   a. Below elements between </ClientSettings> and <AutoMail>

```
<ClientSettings2>
   <WebServicesURL>
http://yourserver:yourport/Node.WebServices/services/NetworkNode2
   </WebServicesURL>
</ClientSettings2>
```

b. Below elements between </AutoMail> and </Configuration>

```
 <NodeSettings2>
   <NodeStatus>
     <Status>Running</Status>
     <Message>The Node 2.0 is Currently Running</Message>
   </NodeStatus>
<NodeURL>http://yourserver:yourport/Node.WebServices/services/NetworkNode2</NodeURL>
   <TokenLifeTime Enabled="true" time="600"/>
   <NAAS>
     <URL name="authentication">https://naas.epacdxnode.net </URL>
     <URL name="user">https://naas.epacdxnode.net/xml/usermgr.wsdl</URL>
     <URL name="policy">https://naas.epacdxnode.net/xml/policy.wsdl</URL>
   </NAAS>
   <NodeAdministrator>
     <Name>Charlie</Name>
     <Credentials>
       <UserID>charlie@enfotech.com</UserID>
       <Password>askAdRTfjiasdfk=-3^78</Password>
     </Credentials>
   </NodeAdministrator>
 </NodeSettings2>
```

c. Below elements between </ProxySettings> and <NodeSettings>

```
<FTPSettings status="A or leave blank " host="yourhost or leave blank " port="22 or leave blank ">
        <Credentials>
                <UserID>your username or leave blank</UserID>
                <Password>your password or leave blank </Password>
        </Credentials>
</FTPSettings>
```

4. Save file then click “Upload” button to upload back to Node database.

6 Initial Application Configuration

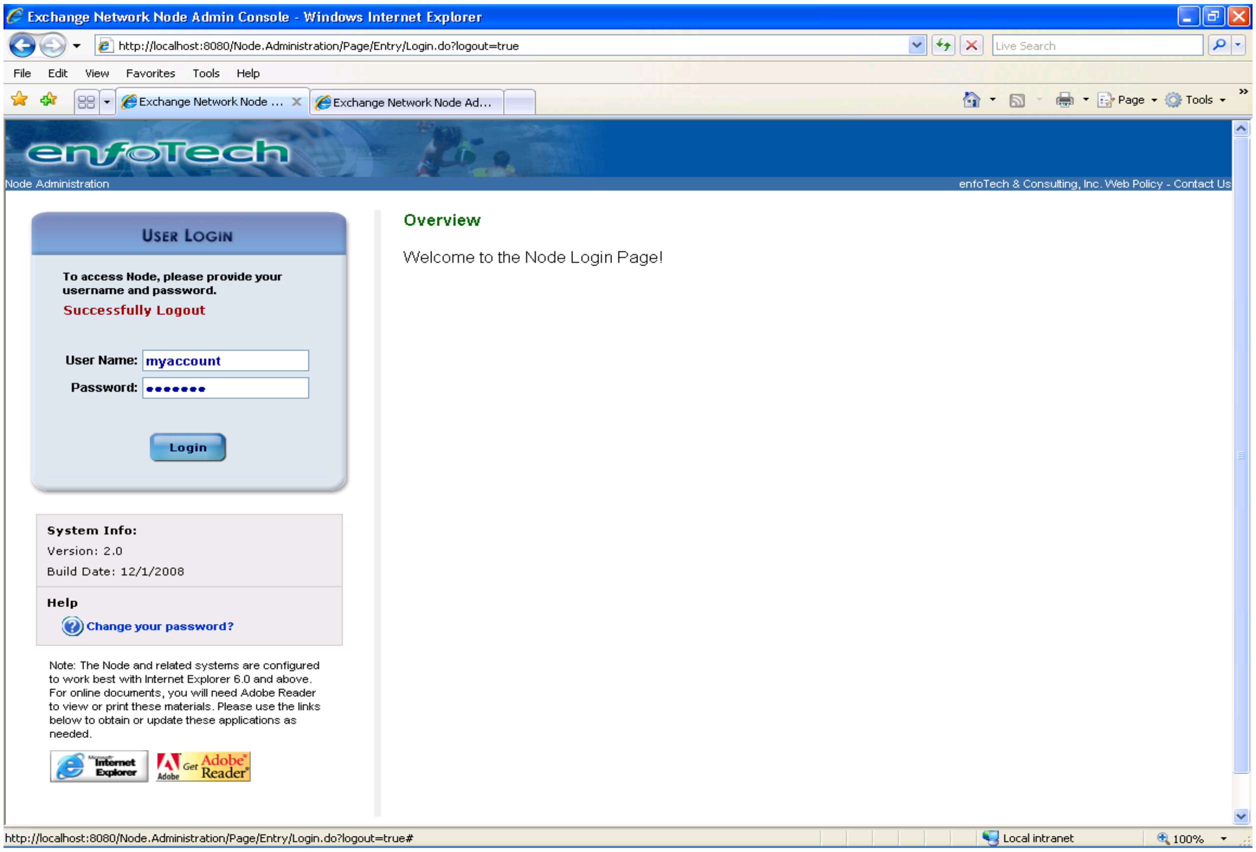
After the Node is installed, there are some initial configuration settings that must be set before the Node can interact with other Nodes on the Exchange Network. This section provides step-by-step instructions for this initial application configuration.

1.) Proceed to the Node.Administration website (the URL should be http://<install directory>/Node.Administration.

Login using the default Node Administration Console username and password, which is:

username: admin
password: password

You will be prompted to change this temporary password to a more secure password during your first successful login attempt.



Exchange Network Node Admin Console - Windows Internet Explorer
http://localhost:8080/Node.Administration/Page/Entry/Login.do?logout=true
Live Search
File
Edit
View
Favorites
Tools
Help
Exchange Network Node ...
Exchange Network Node Ad...
Page
Tools
enfoTech
Node Administration
enfoTech & Consulting, Inc. Web Policy - Contact Us
USER LOGIN
Login
Internet Explorer
Get Adobe Reader
Local intranet
100%

To access Node, please provide your username and password.

Successfully Logout

User Name: myaccount

Password: •••••••

System Info:
Version: 2.0
Build Date: 12/1/2008

Help

Change your password?

Note: The Node and related systems are configured to work best with Internet Explorer 6.0 and above. For online documents, you will need Adobe Reader to view or print these materials. Please use the links below to obtain or update these applications as needed.

Overview

Welcome to the Node Login Page!

http://localhost:8080/Node.Administration/Page/Entry/Login.do?logout=true#

Figure 3: Node Administration Login Page

2.) Proceed to the appropriate Node specification version (1.1 or 2.0) by selecting the appropriate tab located at the top of the dashboard (this will identify the Node specification under which all subsequent actions will be performed for).

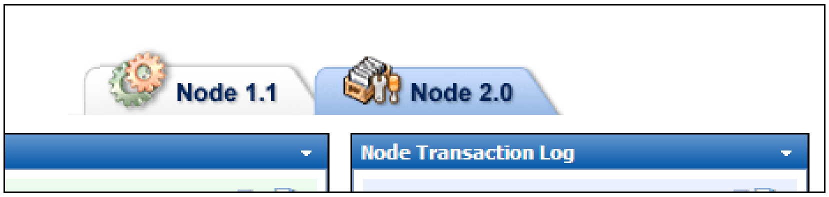
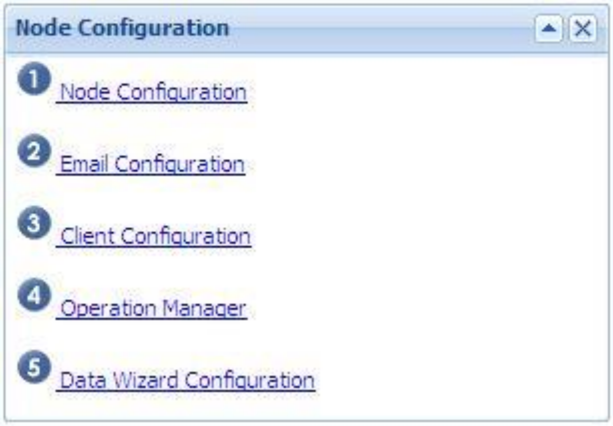


Figure 4: Node.Administration Dashboard

3.) Locate the ‘Node Configuration’ web part on the dashboard, which will look like the following.



Node Configuration

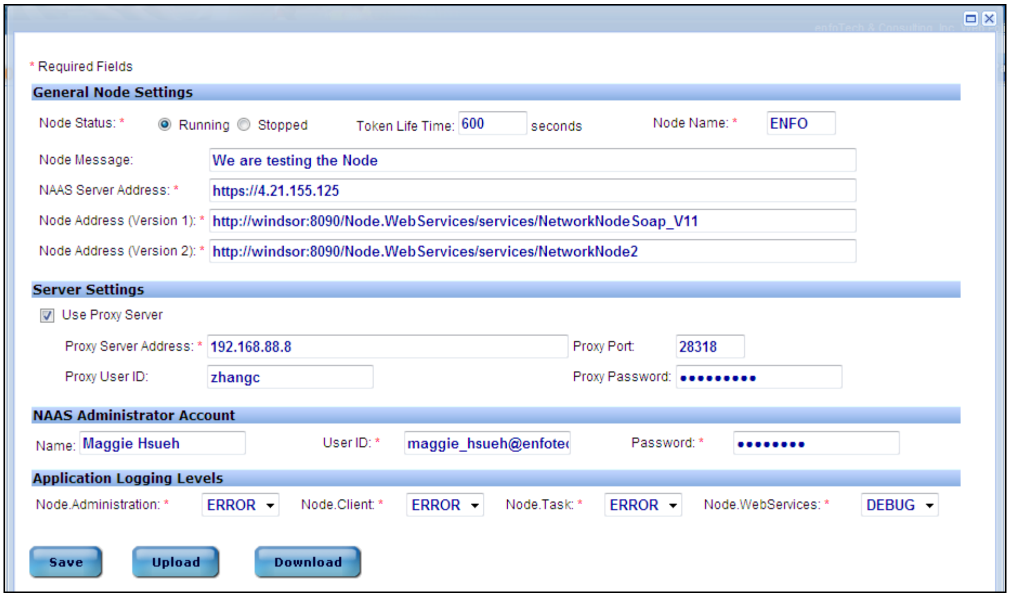
1. Node Configuration
2. Email Configuration
3. Client Configuration
4. Operation Manager
5. Data Wizard Configuration

Figure 5: ***Node Configuration Web Part***

On this screen set up all necessary Node configuration information.

4.) Proceed to link #1 called “Node Configuration”, and set the following configuration settings:

- **Node Name:** EPA-assigned Node identifier for your organization’s node (for states, the name is generally the 2 letter State Code). Please contact the CDX Node Helpdesk if you do not know your EPA-assigned Node ID.
- **Node Address:** Local address of where the NodeServices.asmx file is installed. Please make sure the Node endpoint is different for different version, as shown in the following diagram:
  - Node1.1: http://yourhost:yourport/Node.WebServices/services/NetworkNodeSoap_V11
  - Node2.0: http://yourhost:yourport/Node.WebServices/services/NetworkNode2
- **NAAS Node Administrator Name:** Node Administrator’s Full Name
- **NAAS Node Administrator User ID:** NAAS assigned User ID Please contact the Node Helpdesk if you do not know your EPA-assigned Node NAAS Administrator username/password.
- **NAAS Node Administrator Password:** NAAS assigned Password
- **NAAS Server Address:**
  i. **Test Server:** https://naas.epacdxnode.net
  ii. **Production Server:** https://cdxnodenaas.epa.gov
- **Proxy Settings:** Only use these if the Node server must access a proxy to access the internet.



* Required Fields

General Node Settings

Node Status: * ● Running ○ Stopped Token Life Time: 600 seconds Node Name: * ENFO

Node Message: We are testing the Node

NAAS Server Address: * https://4.21.155.125

Node Address (Version 1): * http://windsor:8090/Node.WebServices/services/NetworkNodeSoap_V11

Node Address (Version 2): * http://windsor:8090/Node.WebServices/services/NetworkNode2

Server Settings

☑ Use Proxy Server

Proxy Server Address: * 192.168.88.8 Proxy Port: 28318

Proxy User ID: zhangc Proxy Password: •••••••••

NAAS Administrator Account

Name: Maggie Hsueh User ID: * maggie_hsueh@enfotec Password: * ••••••••

Application Logging Levels

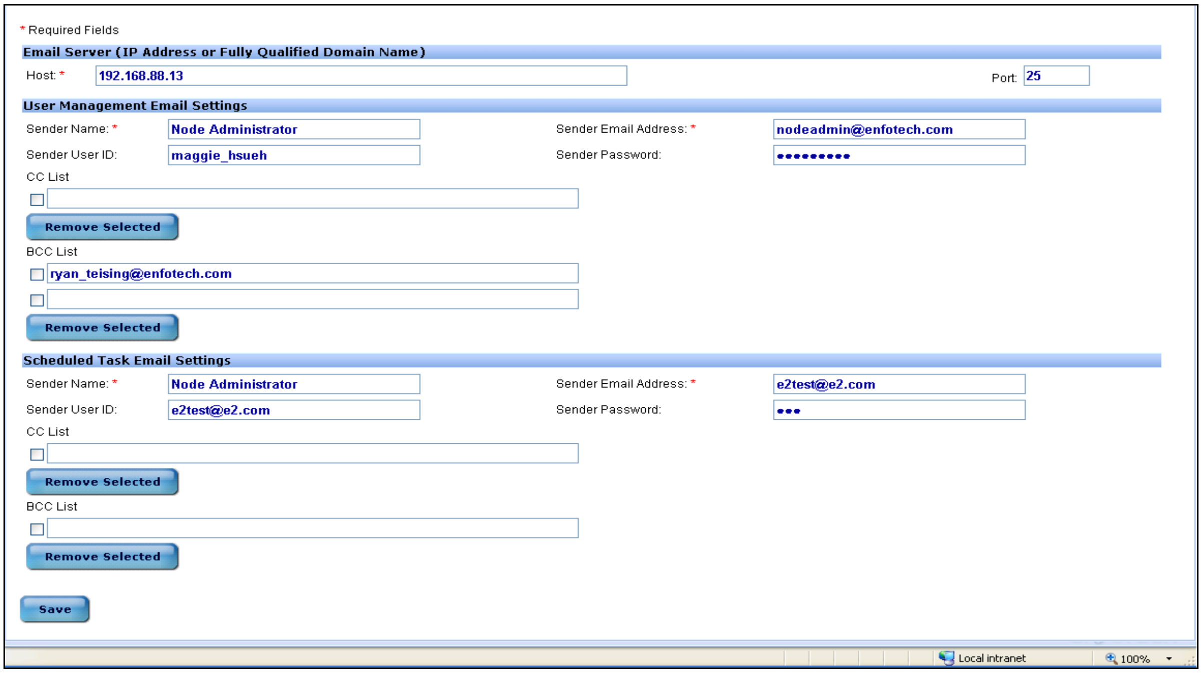
Node.Administration: * ERROR Node.Client: * ERROR Node.Task: * ERROR Node.WebServices: * DEBUG

Figure 6: ***Node Configuration Screen***

Click the Save button when you are done.

Note: you can also click the Download button to download the configuration settings in XM L format, and then reupload using the Upload button. This provides a quicker way to update configuration settings, but please use at your own risk, as this may be easier to make mistakes if you are unfamiliar with the configuration XML structure.

5.) Proceed to option 3, *Email Configuration*, and modify the email server and account information with the email server, port and credentials for sending emails. You can also change the email addresses the emails are coming from, the cc lists and bcc lists. Click the **Save** button when you are done.



* Required Fields

Email Server (IP Address or Fully Qualified Domain Name)

Host: * 192.168.88.13 Port: 25

User Management Email Settings

Sender Name: * Node Administrator Sender Email Address: * nodeadmin@enfotech.com

Sender User ID: maggie_hsueh Sender Password: •••••••••

CC List

□

BCC List

□ ryan_teising@enfotech.com

□

Scheduled Task Email Settings

Sender Name: * Node Administrator Sender Email Address: * e2test@e2.com

Sender User ID: e2test@e2.com Sender Password: •••

CC List

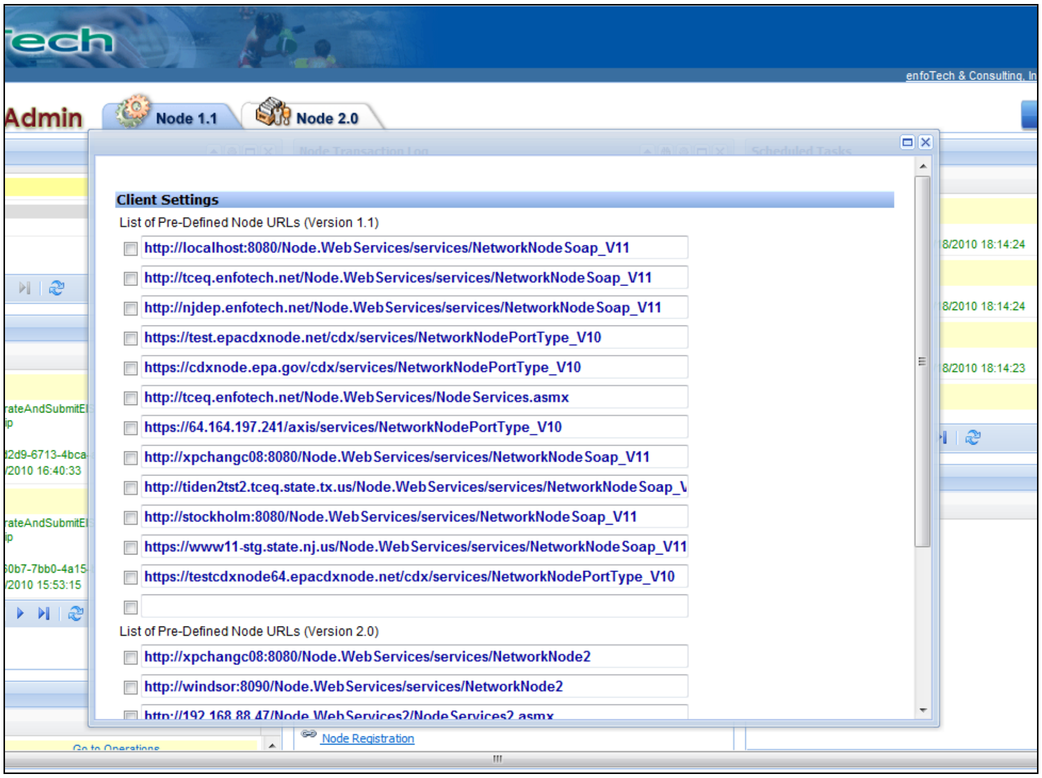
□

BCC List

□

Figure 7: ***Email Configuration Screen***

6.) On the Web Client Configuration screen, enter in any URLs that you want to appear in the drop down of your Node.Client application. Click the **Save** button when you are done.



Client Settings

List of Pre-Defined Node URLs (Version 1.1)

- □ http://localhost:8080/Node.WebServices/services/NetworkNodeSoap_V11
- □ http://tceq.enfotech.net/Node.WebServices/services/NetworkNodeSoap_V11
- □ http://njdep.enfotech.net/Node.WebServices/services/NetworkNodeSoap_V11
- □ https://test.epacdxnode.net/cdx/services/NetworkNodePortType_V10
- □ https://cdxnode.epa.gov/cdx/services/NetworkNodePortType_V10
- □ http://tceq.enfotech.net/Node.WebServices/NodeServices.asmx
- □ https://64.164.197.241/axis/services/NetworkNodePortType_V10
- □ http://xpchangc08:8080/Node.WebServices/services/NetworkNodeSoap_V11
- □ http://tiden2tst2.tceq.state.tx.us/Node.WebServices/services/NetworkNodeSoap_V
- □ http://stockholm:8080/Node.WebServices/services/NetworkNodeSoap_V11
- □ https://www11-stg.state.nj.us/Node.WebServices/services/NetworkNodeSoap_V11
- □ https://testcdxnode64.epacdxnode.net/cdx/services/NetworkNodePortType_V10
- □

List of Pre-Defined Node URLs (Version 2.0)

- □ http://xpchangc08:8080/Node.WebServices/services/NetworkNode2
- □ http://windsor:8090/Node.WebServices/services/NetworkNode2
- □ http://192.168.88.47/Node.WebServices2/NodeServices2.asmx

Figure 8: ***Node Client Configuration Page***

6.1 Installation of Data Flow Plug-Ins

The EN-Node fully supports all PlugIn modules of enfoTech e-Node1.1. Please follow the PlugIn installation guides which are shipped with the Plugin to install the PlugIn exactly like the enfoTech e-Node1.1. A typical installation of PlugIn modules follows these steps:

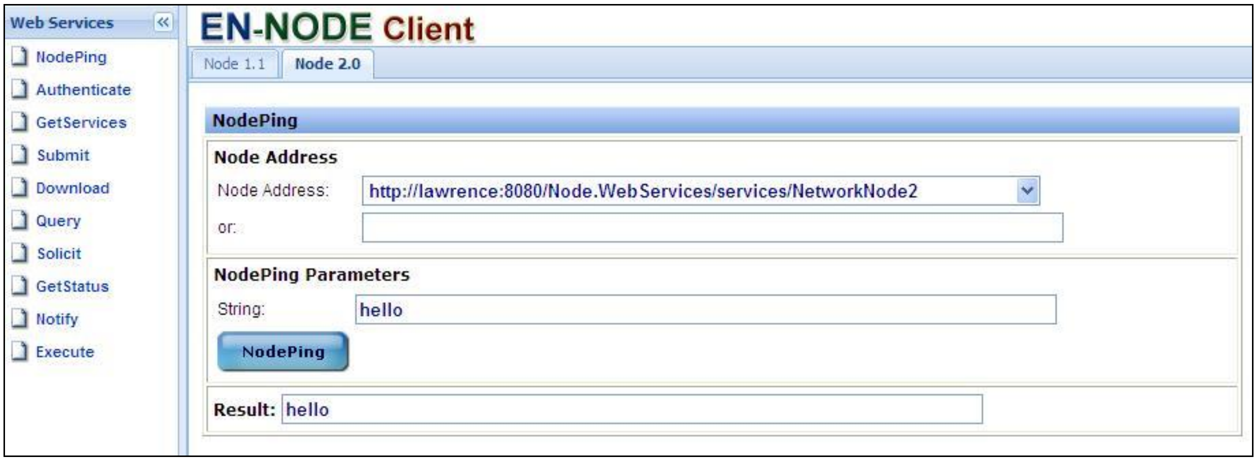
1. Stop server
2. Copy PlugIn jar files into all web modules lib folder. For example:
   Tomcat 5.0\webapps\Node.Administration\WEB-INF\lib
   Tomcat 5.0\webapps\NodeTask\WEB-INF\lib
   Tomcat 5.0\webapps\Node.Client\WEB-INF\lib
   Tomcat 5.0\webapps\Node.WebServices\WEB-INF\lib
3. Add PlugIn special parameters in Web.xml which are in \ WEB-INF\ folder. For example:
   Tomcat 5.0\webapps\Node.Administration\WEB-INF\ Web.xml
   Tomcat 5.0\webapps\NodeTask\WEB-INF\ Web.xml
   Tomcat 5.0\webapps\Node.Client\WEB-INF\ Web.xml
   Tomcat 5.0\webapps\Node.WebServices\WEB-INF\ Web.xml
4. Start server.

7 Installation Confirmation Testing

After the node is completely installed and configured, you can perform some tests to confirm that it is working properly.

a). **Conduct Node Ping test:** The first test is a “Node Ping” test to confirm that the Node Web Services are operational.

- Proceed to the Node.Client website (http://<install-directory>>/Node.Client). The Node Ping page will be displayed by default. Navigate to the Node 1.1 tab or Node 2.0 tab, depending on which version you are testing.
- Select the Node Web Service address from the **Node Address** dropdown list that corresponds to your Node’s Web Service endpoint. It is important to only test your Node 2.0 endpoint using the Node 2.0 tab and Node 1.1 endpoint using the Node 1.1 tab. They are not cross-compatible (i.e. you cannot use the Node 1.1 client tab to test your Node 2.0 web services, and vice versa.)
- Enter “hello” (or any other message) in the **String** textbox and click the **NodePing** button. The screen should refresh and the Node Ping Result should be “hello”, as shown here:



Node Address: http://lawrence:8080/Node.WebServices/services/NetworkNode2

or:

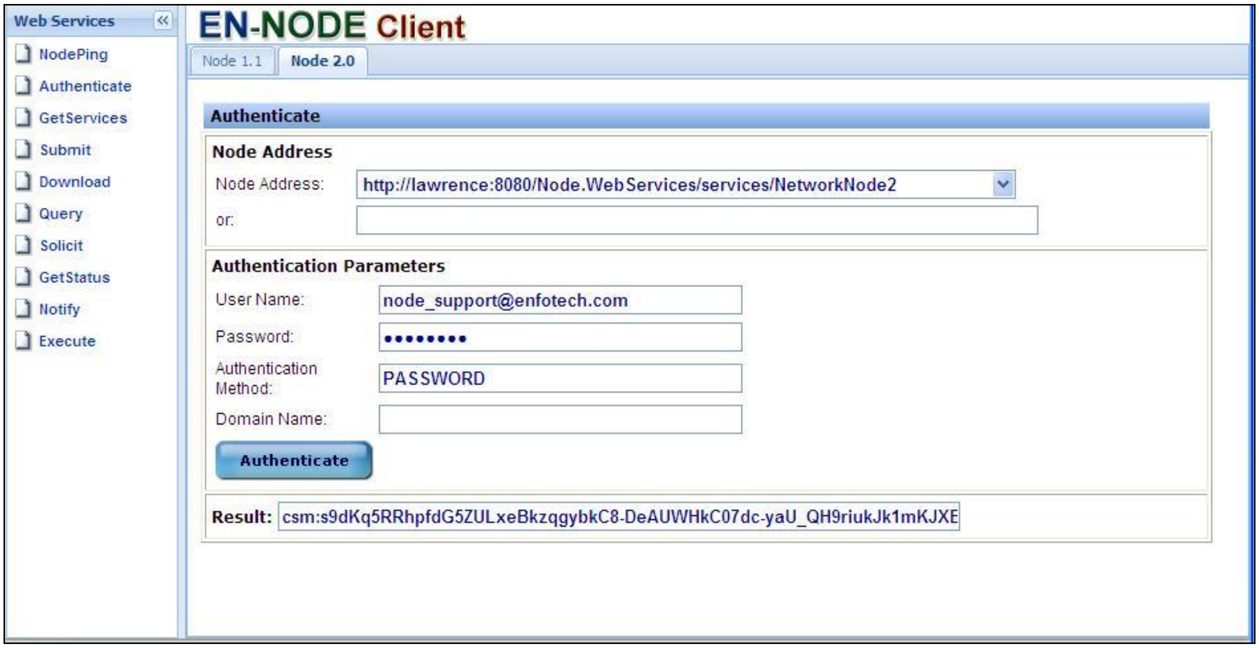
String: hello

Result: hello

Figure 9: *Node Ping Using Node Client*

b). **Conduct Authenticate Test:** The purpose of the authenticate test is to test whether your Node can communicate with EPA’s NAAS security service.

- Click the **Authenticate** link from the left menu
- Enter in your NAAS username and password, make sure **Authentication Method** = PASSWORD, and then click the **Authenticate** button.
Note: If you do not have a NAAS username/password, please contact the CDX Node Helpdesk.
- The screen should refresh and you should be presented with a long string which is a security token, as shown here:



Authenticate

Node Address

Node Address: http://lawrence:8080/Node.WebServices/services/NetworkNode2

or:

Authentication Parameters

User Name: node_support@enfotech.com

Password: ••••••••

Authentication Method: PASSWORD

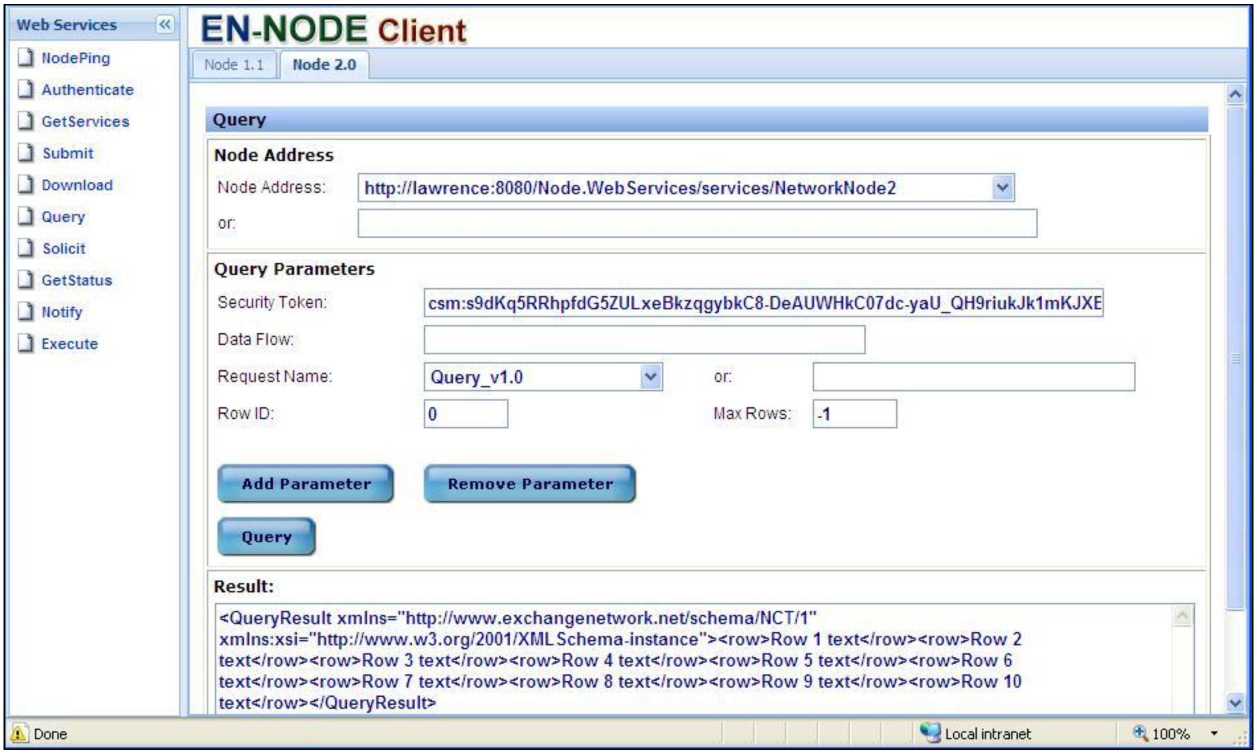
Domain Name:

Result: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Figure 10: ***Authenticate to the local Node***

c). **Conduct Query Test:** You can only perform this test if the Node has a Query data service provided. By default, one testing Query service is included in the Node installation called “NCT” (which stands for Node Certification Testing). This will test the Node’s ability to respond to a query request.

- Click the Query link from the left menu
- Select the request called “Query_v1.0” from **Request Name** dropdown list
- Click the **Query** button to obtain the query result. The screen should refresh and display the query results as shown in the figure below:



Node Address

Node Address: http://lawrence:8080/Node.WebServices/services/NetworkNode2

or:

Query Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Data Flow:

Request Name: Query_v1.0 or:

Row ID: 0 Max Rows: -1

Result:

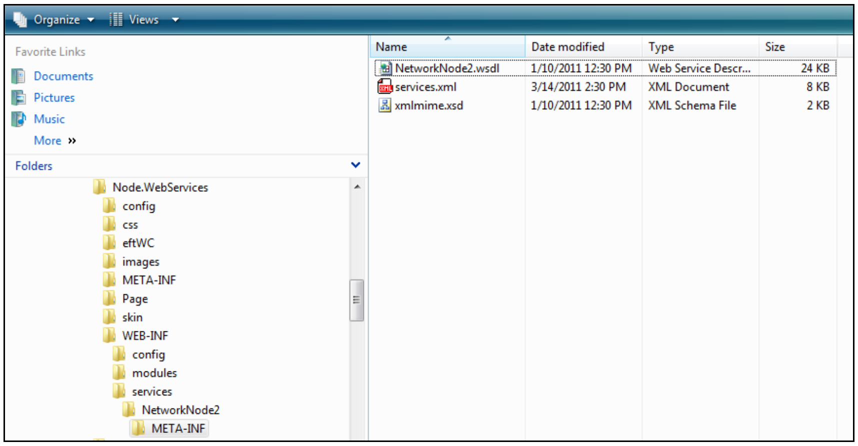
```
<QueryResult xmlns="http://www.exchangenetwork.net/schema/NCT/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><row>Row 1 text</row><row>Row 2
text</row><row>Row 3 text</row><row>Row 4 text</row><row>Row 5 text</row><row>Row 6
text</row><row>Row 7 text</row><row>Row 8 text</row><row>Row 9 text</row><row>Row 10
text</row></QueryResult>
```

Figure 11: *Query Test*

8 MTOM Configuration

The EN-Node supports MTOM transmission format. Sometimes you may need to change the MTOM configuration setting based on incoming requests. (For example, Node 2.0 specs require dynamic MTOM, whereas Node 2.1 specs require MTOM always on.) In this situation, you must change the services.xml of Axis2 to implement the configuration change.

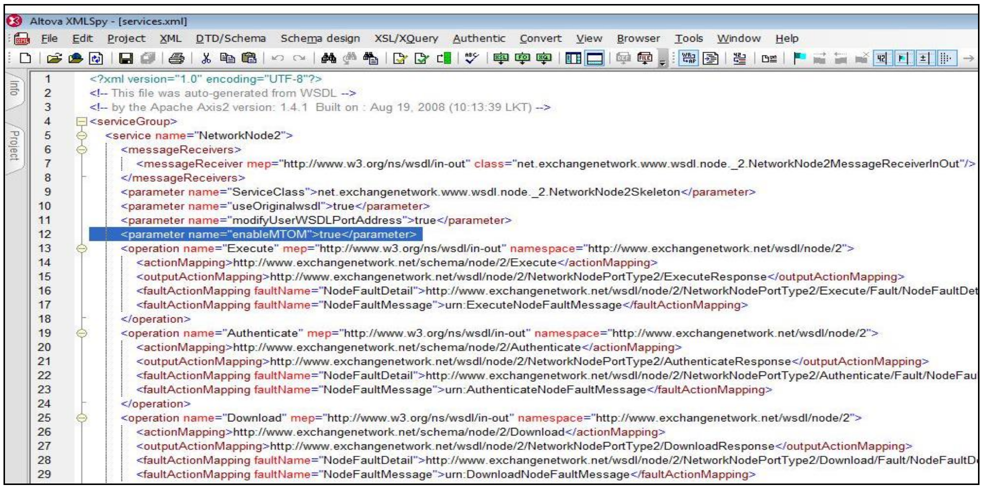
1. Unzip Node.WebService.war to any folder like below:



Name	Date modified	Type	Size
NetworkNode2.wsdl	1/10/2011 12:30 PM	Web Service Descr...	24 KB
services.xml	3/14/2011 2:30 PM	XML Document	8 KB
xmlmime.xsd	1/10/2011 12:30 PM	XML Schema File	2 KB

2. .Go to Node.WebServices\WEB-INF\services\NetworkNode2\META-INF and find the **services.xml**. Open this file to find the statement:

   `<parameter name="enableMTOM">true</parameter>`



```xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- This file was auto-generated from WSDL -->
<!-- by the Apache Axis2 version: 1.4.1  Built on : Aug 19, 2008 (10:13:39 LKT) -->
<serviceGroup>
    <service name="NetworkNode2">
        <messageReceivers>
            <messageReceiver mep="http://www.w3.org/ns/wsdl/in-out" class="net.exchangenetwork.www.wsdl.node._2.NetworkNode2MessageReceiverInOut"/>
        </messageReceivers>
        <parameter name="ServiceClass">net.exchangenetwork.www.wsdl.node._2.NetworkNode2Skeleton</parameter>
        <parameter name="useOriginalwsdl">true</parameter>
        <parameter name="modifyUserWSDLPortAddress">true</parameter>
        <parameter name="enableMTOM">true</parameter>
        <operation name="Execute" mep="http://www.w3.org/ns/wsdl/in-out" namespace="http://www.exchangenetwork.net/wsdl/node/2">
            <actionMapping>http://www.exchangenetwork.net/schema/node/2/Execute</actionMapping>
            <outputActionMapping>http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/ExecuteResponse</outputActionMapping>
            <faultActionMapping faultName="NodeFaultDetail">http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/Execute/Fault/NodeFaultDet
            <faultActionMapping faultName="NodeFaultMessage">urn:ExecuteNodeFaultMessage</faultActionMapping>
        </operation>
        <operation name="Authenticate" mep="http://www.w3.org/ns/wsdl/in-out" namespace="http://www.exchangenetwork.net/wsdl/node/2">
            <actionMapping>http://www.exchangenetwork.net/schema/node/2/Authenticate</actionMapping>
            <outputActionMapping>http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/AuthenticateResponse</outputActionMapping>
            <faultActionMapping faultName="NodeFaultDetail">http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/Authenticate/Fault/NodeFau
            <faultActionMapping faultName="NodeFaultMessage">urn:AuthenticateNodeFaultMessage</faultActionMapping>
        </operation>
        <operation name="Download" mep="http://www.w3.org/ns/wsdl/in-out" namespace="http://www.exchangenetwork.net/wsdl/node/2">
            <actionMapping>http://www.exchangenetwork.net/schema/node/2/Download</actionMapping>
            <outputActionMapping>http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/DownloadResponse</outputActionMapping>
            <faultActionMapping faultName="NodeFaultDetail">http://www.exchangenetwork.net/wsdl/node/2/NetworkNodePortType2/Download/Fault/NodeFaultD
            <faultActionMapping faultName="NodeFaultMessage">urn:DownloadNodeFaultMessage</faultActionMapping>
```

   Set this element to one of the following statuses:

   - **true**: Force to enable MTOM.
   - **false**: Force to disable MTOM.
   - **optional**: Let Axis 2 to select MTOM automatically based on incoming request format. For example, if the incoming request is general soap format; Axis 2 will respond with general soap format; if the incoming request is MTOM, Axis 2 will return a MTOM formatted response.

3. Zip the Node.WebServices folder back to Node.WebServices.war
4. Deploy the new Node.WebServices.war package.