EN-NODE
INSTALLATION GUIDE
(.NET VERSION)

Version: 2.3

October 2, 2012

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

Revision History

Version	Date	Created By	Reviewed By	Description
2.3	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:

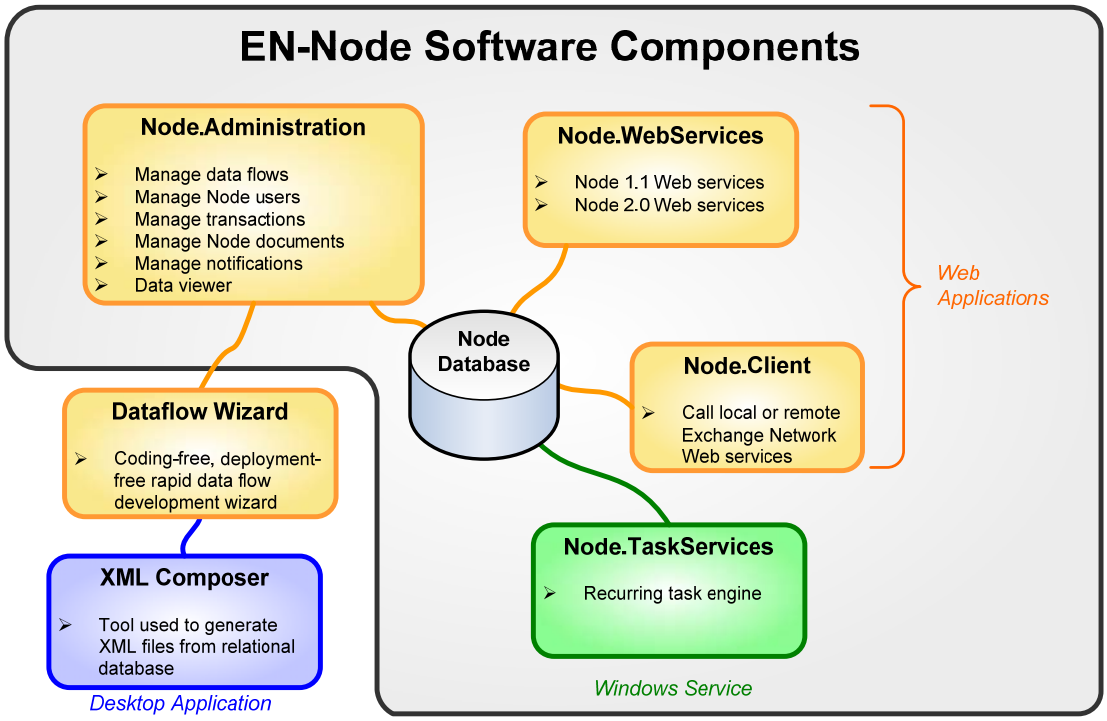

Figure 1: EN-Node Components

- **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:

- **All Servers:**
  - Windows Server 2003 R2 or Windows Server 2008 (64 bit editions are supported)
- **Web / Application Server:**
  - Internet Information Services 5, 6, 7, or 7.5
  - ASP.NET 2.0
  - Microsoft Web Services Enhancements (WSE) 2.0 SP3
  - Microsoft Web Services Enhancements (WSE) 3.0
  - Microsoft .NET Framework .NET framework 3.5
- **Database Server:**
  - Microsoft SQL Server 2000 or later (including Express Editions)
    -or-
    Oracle Database Server 9i or later

4 EN-Node Deployment Options

EN-Node supports multiple deployment options. The option that is best suited for your installation will likely depend on your organization's existing Network topology and security requirements or preferences. These various deployment options are described in this section.

4.1 Deployment Option A: Standard Deployment Architecture

The following diagram depicts the typical Node deployment approach, which has the following characteristics:

- All Node application components (Node.Administration, Node.Client, Node.Webservices, Node.WebServices2, and Node.Task) are installed on the same machine in the DMZ. This Web Server is optionally Load Balanced.
- The Node database is installed inside an inner firewall on SQL Server or Oracle

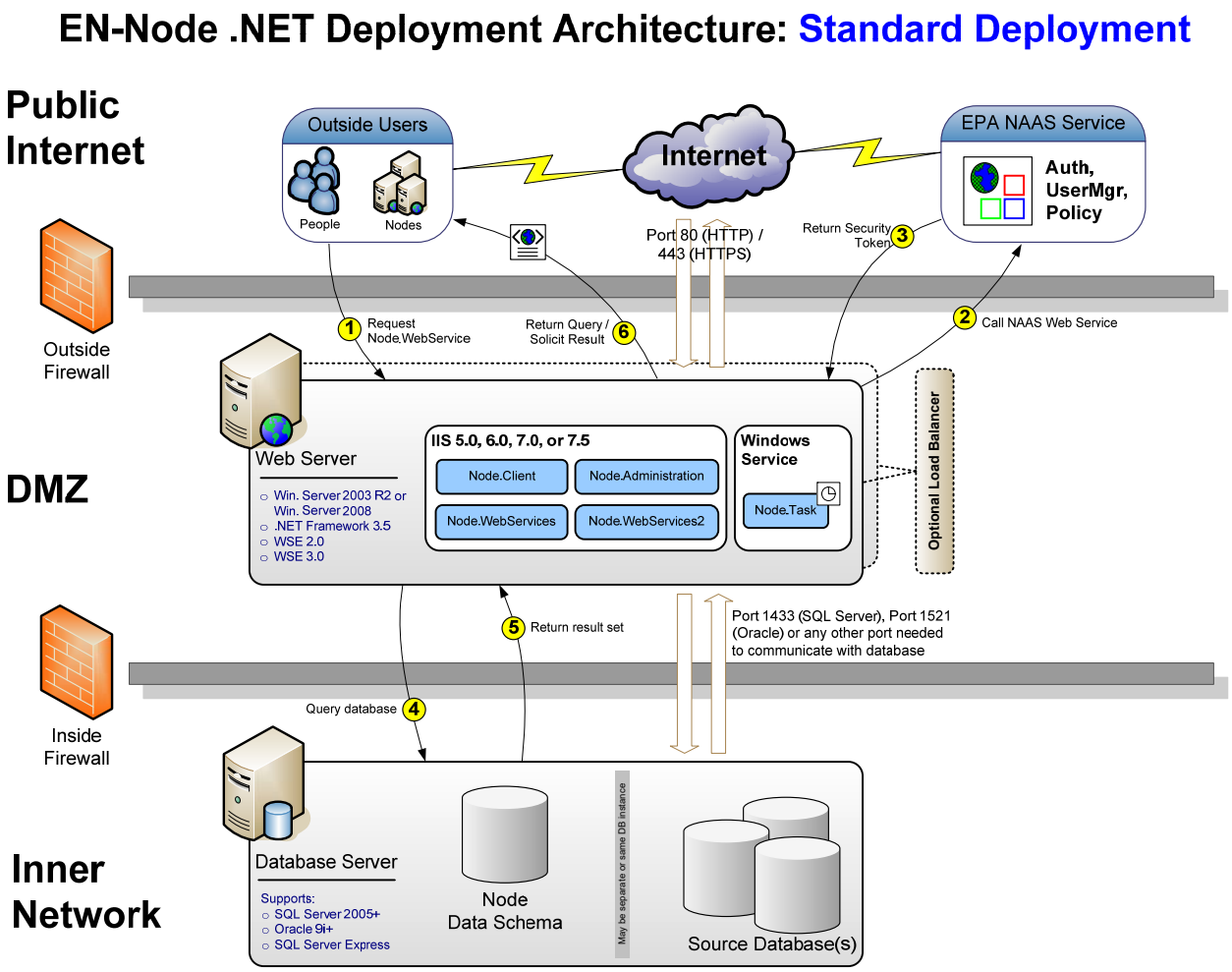

Figure 2: Typical Deployment Architecture

This diagram also indicates a typical web service interaction in steps 1 – 6. This illustrates an example of a query or solicit call and response on the node.

4.2 Deployment Option B: Simplified Deployment Architecture

The 2nd most common deployment approach is to install all components on the same server. This differs from the standard deployment by having the Node database installed on the same server as the Node applications.

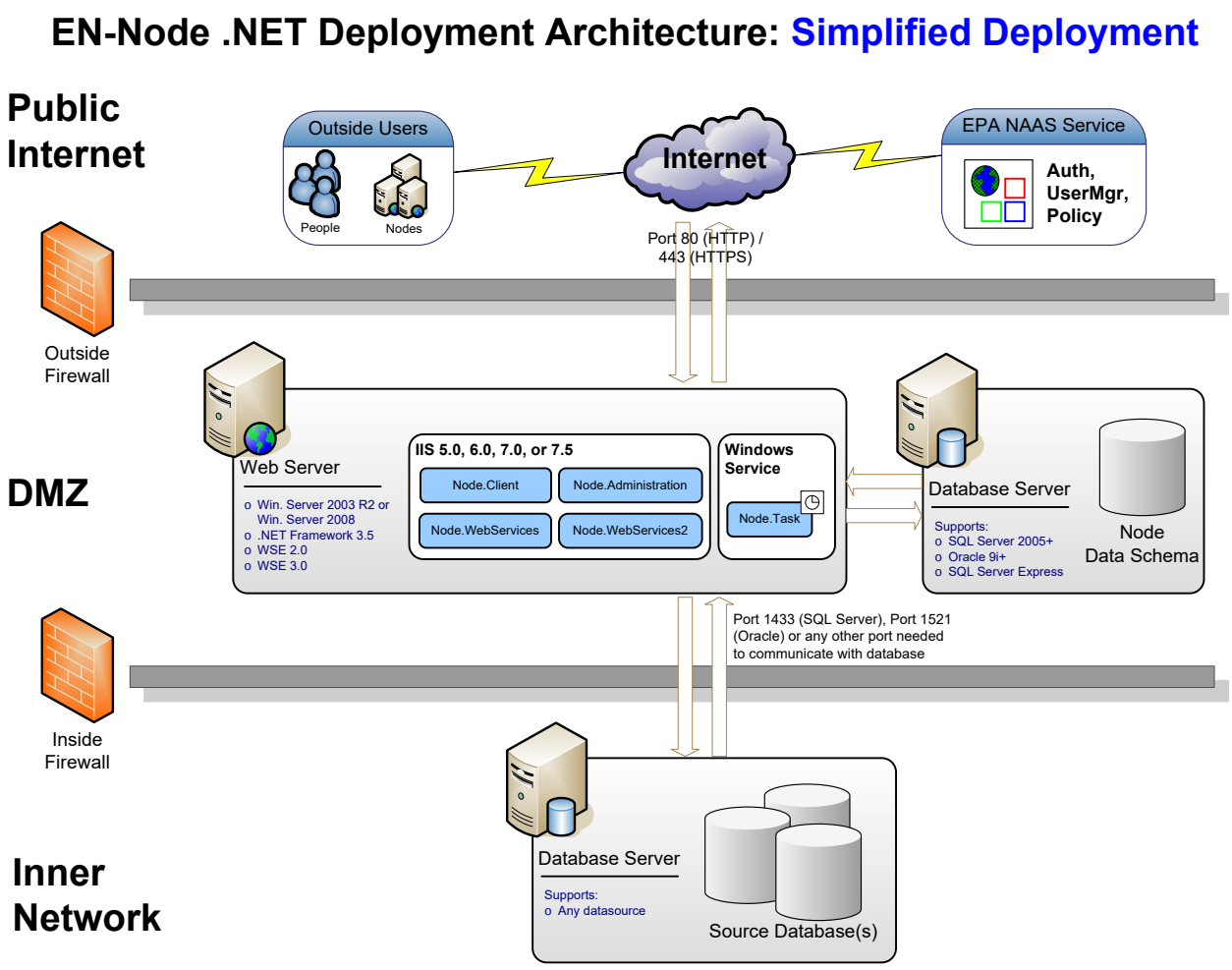

Figure 3: Simplified Deployment Architecture

This deployment approach has the following Pros and Cons:

Pros	Cons
• Less hardware requirements (1 server versus 2)	• Decreased security (increased opportunity for unauthorized access to Node Database)

This option is less secure, by increasing the exposure of the database to the public, and is only recommended in cases where the customer is limited to one server or the Node is processing non-sensitive information.

4.3 Deployment Option C: Secure Deployment Architecture

As an alternative, some customers prefer to install the Node.Administration and/or the Node.Task applications inside an inner firewall for increased security. This configuration is supported and shown in the following diagram:

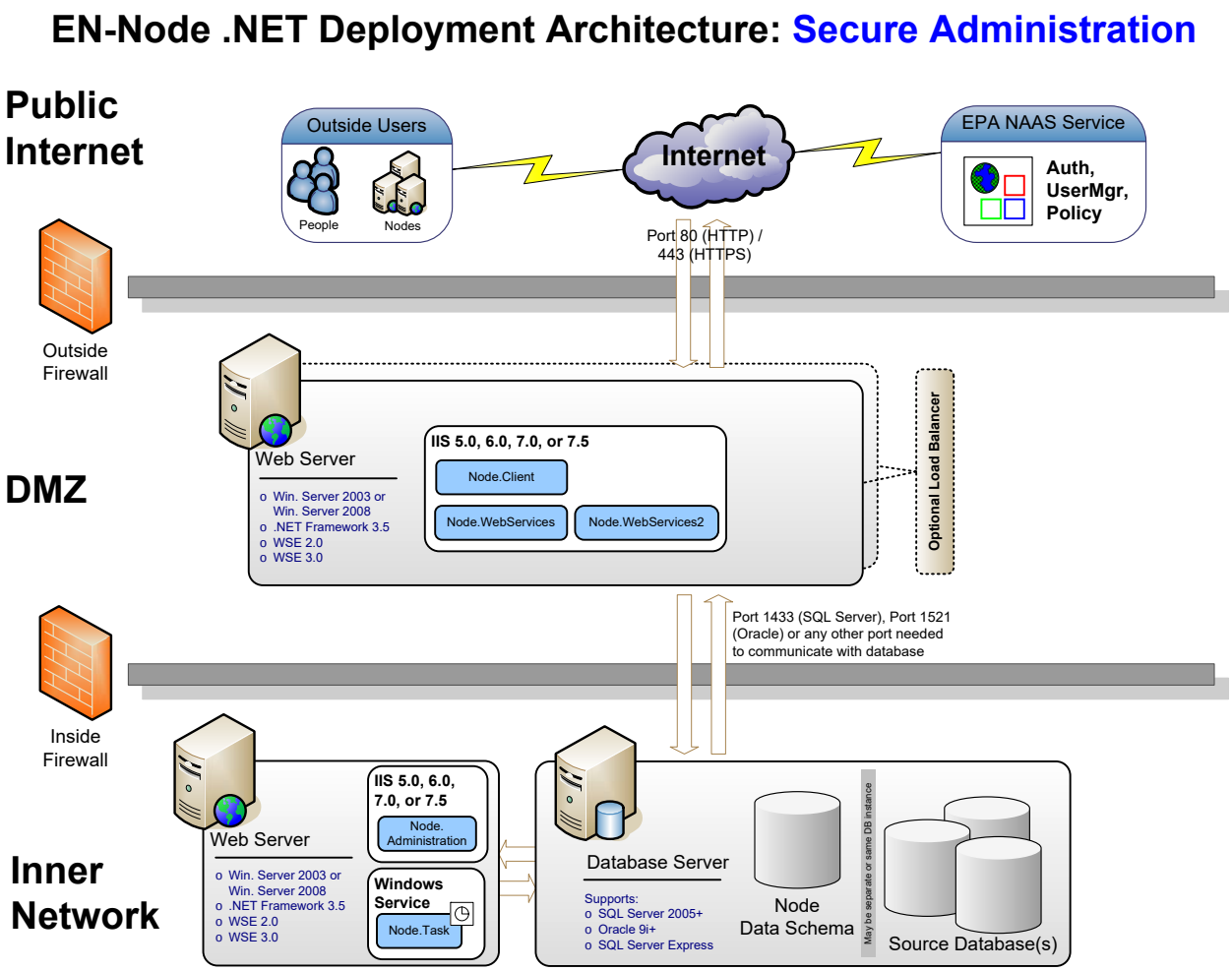

Figure 4: More Secure Application Deployment Architecture

This deployment approach has the following Pros and Cons:

Pros	Cons
• Increased security (reducing opportunity for unauthorized access to Node Administration Console or Task engine)	• Reduces ability for remote administration of Node • More complex deployment

5 EN-Node System Installation Instructions

Note: these installation instructions assume that the reader is experienced in deploying web applications on Microsoft IIS and also experienced in importing databases in either SQL Server or Oracle.

5.1 Deployment Package File Structure

A description of the Node deployment package files structure and directory contents are provided below. Please note that the deployment package should be downloaded to the local drive on the Node server (and/or Node database server) and accessible to the Administrator.

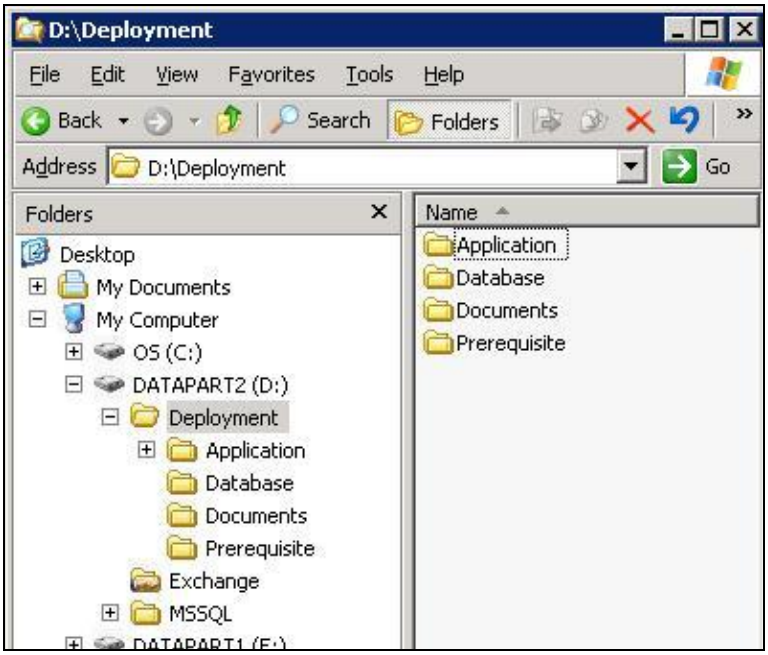

Figure 5: Deployment File Structure

Under the **Deployment** folder, the following sub-folders can be found:

- **Application** folder, which contains the following sub-folders:
  - *Node.Administratrion*
  - *Node.Client*
  - *Node.WebService* (used for Node 1.1 and Node 2.0 specifications)
  - *Node.TaskHandler*
  - *Node.TaskService*
  - *Logs*
- **Database** folder, which contains the following contents:
  - *Node2008.bak*
  - *RestoreNode2008.sql*
  - *aspnode.dmp*
  - *RestoreDB.sql*
- **Document** folder, which contains the following contents:
  - *EN-Node Installation Guide.doc*
- **Prerequisite** folder, which contains the following contents:
  - *Microsoft WSE 3.0.msi*
  - *Microsoft WSE 2.0 SP3.msi*
  - *dotNetFx35setup.exe*

The use of the folders or files identified above are explained in further detail in the remaining sections.

5.2 Node Database Installation

This section provides instructions on installing and configuring the Node database. The Node Database can be either Oracle 9i or higher or SQL Server 2000 or higher (including SQL Server Express editions). Please note that the instructions presented below must be performed from the Database Server.

For SQL Server:

1.) Under the *Deployment\Database* folder, locate the **RestoreNode2008.sql** file and open this file for editing in MS SQL Server Management Studio

2.) In the file, locate the following statement and modify the areas highlighted in yellow, as shown below:

```
RESTORE DATABASE [Node2008] FROM DISK = N'<Node Database dump file location>'
WITH FILE = 1,
MOVE N'Node2008' TO N'<Data file location>',
MOVE N'Node2008Log' TO N'<Log file location> ',
NOUNLOAD, STATS = 10
```

Example:

<Dump file location> = D:\Deployment\Database\Node2008.bak
<Data file location> = D:\MSSQL\DATA\Node2008.MDF
<Log file location> = D:\MSSQL\DATA\Node2008_log.LDF

3.) Save the modifications made to the file for future reference.

4.) Execute/run the file, which will install the databases on the appropriate Database server

For Oracle:

1.) From the *Deployment\Database* folder, execute the **RestoreDB.sql** script which will create a tablespace that will contain the Node database. You may wish to edit this file prior to executing it to update the location/name of the datafile.

2.) From the *Deployment\Database* folder, import the table data provided in the **aspnode.dmp** file into the Oracle schema that was created for the Node database

5.3 Node Application Installation

This section provides instructions on installing and configuring the Node applications. Please note that the instructions presented below must be performed on the Node Application Server. The Node Application Server must meet the following criteria:

a. Application Server must be accessible such that requests made from outside of the network are accepted.

b. Application Server must have permissions to execute queries, updates, and deletes on the Node database (i.e. a database user must be created with these rights).

c. Application Server may also need to have the ability to communicate with other databases, depending on the data flows available in the Node, such that the Application Server is able to execute any database calls on any potential databases that a data flow references

Note: If you plan on supporting both Node 1.1 and Node 2.0 web services, two Node.WebServices folders will need to be created (one folder called *Node.WebService* for Node 1.1 and the other called *Node.WebService2* for Node 2.0). The web.config file under each Node.WebService folder will also need to be configured such that it references the appropriate specification. These steps are covered in more detail in the subsequent instructions.

Step-by-Step Node Application Installation Instructions:

1.) Install .NET Framework 3.5 (from the *Deployment\Prerequisite\Microsoft* folder directory).

2.) Install Microsoft WSE 2.0 SP3(Download from the *Microsoft*).

3.) Install Microsoft WSE 3.0 (Download from the *Microsoft*).

4.) Under the Application directory of the deployment package, there are five sub directories: Node.Administration, Node.Client, Node.TaskHandler, Node.TaskService, and Node.WebServices. Copy these directories onto the file system of the application server.

Note: *Another copy of the Node.WebServices folder (named Node.WebServices2) will also need to be created if you plan on supporting both Node 1.1 and Node 2.0 web services.*

5.) Launch Internet Information Services (IIS) Manager

6.) In IIS, create one or more application pools for the Node application. The most common approach is to create only one application pool for the all Node web applications: Node.Administration, Node.Client, Node.WebServices, and Node.WebServices2. Or, the user may wish to create a separate application pool for each of these applications.

- ***Note:*** *If using IIS7, the following additional settings are recommended:*
  - *Set as "Classic" for Managed Pipeline Mode*
  - *After creating the Application Pool, click on "Advanced Settings" and change Process Model Identity to "NetworkService"*

7.) Create 3-4 virtual directories (if using IIS 5-6) or 3-4 applications (if using IIS 7) for the 3 or 4 web applications, pointing the directories to the appropriate folder directory (e.g. the virtual directory for the Node.Administration application should point to the location where you have copied the Node.Administration directory, etc). Grant each web application *Read, Write, Run Scripts* permissions.

Note: *If both Node 1.1 and Node 2.0 specifications are to be used in the EN-Node, a fourth virtual directory (Node.WebServices2) must be created and pointed to the appropriate file directory.*

8.) **Setup Logging Directory:** Identify which folder directory you would like the log files to output to and create the folder and log. There can be up to four different directories, one for each of the applications. Alternatively, each application could share the same directory for the log files. Create the directories and folder structure, if needed. Make sure that the ASPNET and "Network Service" users have rights to read, write, and modify from this directory, both at the file system level and throughout IIS.

9.) All 3 (or 4, if supporting Node 1.1 and Node 2.0) virtual directories should be configured through IIS to use .the NET 2.0 Framework / ASP.NET 2.0 version. (This may require running aspnet_regiis if the services are previously configured as NET Framework 1.1)

10.) Settings in several configuration files will then need to be established as shown in the table below:

Table 3: Configuration Settings

File Path	Section in File	Instructions
***Node.Administration* directory:**		

File Path	Section in File	Instructions
Web.config	<appSettings> (key="Node.TaskHandler.Path")	Modify the "value" attribute to point to the Node.TaskHandler.exe file located in the Node.TaskHandler directory. **i.e.** `<add key="Node.TaskHandler.Path" value="F:\Web\Node.TaskHandler\Node.TaskHandler.exe"/>` **Note:** Node.TaskHandler, Node.TaskServices and Node.Administration should be installed on the same server
	<appSettings> (key="LOG_PATH")	Modify the "value" attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** `<add key="LOG_PATH" value="F:\Web\logs\AdminLog.txt"/>`
	<connectionStrings> (name="node")	Modify the "connectionString" attribute with the correct Node database connection information. The "providerName" attribute should be "System.Data.OracleClient" if the node database is Oracle or "System.Data.SqlClient" if the node database is SQL Sever. Reference Appendix 1 for information on setting the connection string. **i.e.** `<add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword" providerName="System.Data.SqlClient"/>`
***Node.Client* directory:**		
Web.config	<appSettings> (key="LOG_PATH")	Modify the "value" attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** `<add key="LOG_PATH" value="F:\Web\logs\ClientLog.txt"/>`
	<connectionStrings> (name="node")	Modify the "connectionString" attribute with the appropriate the Node database connection information. The providerName attribute should be "System.Data.OracleClient" if the node database is Oracle or "System.Data.SqlClient" if the node database is SQL Sever. Reference Appendix 1 for information on setting the connection string. **i.e.** `<add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword" providerName="System.Data.SqlClient"/>`
***Node.TaskHandler* directory:**		
Node.TaskHandler.exe.config	<appSettings> (key="LOG_PATH")	Modify the value attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** `<add key="LOG_PATH" value="F:\Web\Node.TaskHandler\TaskLog.txt" />`
	<connectionStrings>	Modify the "connectionString" attribute with the appropriate

File Path	Section in File	Instructions
	(name="node")	the Node database connection information. The providerName attribute should be "System.Data.OracleClient" if the node database is Oracle or "System.Data.SqlClient" if the node database is SQL Sever. Reference Appendix 1 for information on setting the connection string. **i.e.** `<add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword" providerName="System.Data.SqlClient"/>`
***Node.TaskService* directory:**		
TaskService.exe.config	<connectionStrings> (name="node")	Modify the connectionString attribute with the appropriate the Node database connection information. The providerName attribute should be "System.Data.OracleClient" if the node database is on an Oracle Database or "System.Data.SqlClient" if the node database is a SQL Sever database. Reference Appendix 1 for information on setting the connection string. **i.e.** `<add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword" providerName="System.Data.SqlClient"/>`
	<applicationSettings> <TaskService.Properties.Settings> (name="Path")	Modify the value attribute to point to Node.TaskHandler.exe file located in the Node.TaskHandler directory. **i.e.** `<add key="Node.TaskHandler.Path" value="F:\Web\Node.TaskHandler\Node.TaskHandler.exe"/>` **Note:** Node.TaskHandler, Node.TaskServices and Node.Administration should be installed on the same server
***Node.WebServices* directory:**		
Web.config	<appSettings> (key="LOG_PATH")	Modify the value attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** `<add key="LOG_PATH" value="F:\Web\logs\WebServicesLog.txt" />`
	<appSettings> (key="SOLICIT_LOG_PATH")	Modify the value attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** `<add key="SOLICIT_LOG_PATH" value="F:\Web\logs\WebServicesLog.txt" />`
	<connectionStrings> (name="node")	Modify the "connectionString" attribute with the appropriate the Node database connection information. The providerName attribute should be "System.Data.OracleClient" if the node database is Oracle or "System.Data.SqlClient" if the node database is SQL Sever. Reference Appendix 1 for information on setting the connection string. **i.e.** `<add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword"`

File Path	Section in File	Instructions
		providerName="System.Data.SqlClient"/>
	<system.web> <webServices>	Uncomment the appropriate block (as indicated in the file) for Node 1.1
***Node.WebServices2* directory (if applicable):**		
Web.config	<appSettings> (key="LOG_PATH")	Modify the value attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** <add key="LOG_PATH" value="F:\Web\logs\WebServicesLog.txt" />
	<appSettings> (key="SOLICIT_LOG_PATH")	Modify the value attribute to identify the full path under which the log file is created for this application, including the file name (create the log file in the appropriate directory, if it does not already exist). **i.e.** <add key="SOLICIT_LOG_PATH" value="F:\Web\logs\WebServicesLog.txt" />
	<connectionStrings> (name="node")	Modify the "connectionString" attribute with the appropriate the Node database connection information. The providerName attribute should be "System.Data.OracleClient" if the node database is Oracle or "System.Data.SqlClient" if the node database is SQL Sever. Reference Appendix 1 for information on setting the connection string. **i.e.** <add name="node" connectionString="Server=enfo_001;Database=node2008;UID=nodeuser;PWD=nodepassword" providerName="System.Data.SqlClient"/>
	<system.web> <webServices>	Uncomment the appropriate block (as indicated in the file) for Node 2.0

11.) Run the **TaskServiceInst.bat** file located in the Node.TaskService directory. This will install and start a windows task service on the server that will be used any time the node is running recurring tasks.

Note 1: *The Task Service will need to uninstalled and reinstalled from the Node.TaskServices folder every time a modification is made to the Node.TaskHandler.exe.config (under the Node.TaskHandler folder) or TaskService.exe.config (under the Node.TaskServices folder) configuration file such that the new configuration settings is extracted and used by the task service.*

Note 2: *If you are having difficulty running this batch file without errors, please refer to Appendix B: Troubleshooting Guide.*

12.) If using the optional Data Flow Wizard application (sold separately). Open IIS Manager and add a MIME Type (application/x-silverlight-app) with the file extension of (.xap) for the DataWizard Component.

13.) If you will be using https for secured transmission, add <security mode="Transport"><transport clientCredentialType="None"/></security> to <system.serviceModel><bindings><basicHttpBinding><binding>

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.

Obtaining Essential Prerequisite Information from EPA

Before you can use your Node, there are several pieces of information that must be obtained from EPA. These are:

1. **Unique Node Identifier:** this will uniquely identify your Node on the Exchange Network
2. **Node Administrator Account:** this will be a NAAS username and password that is used to administer your node.
3. **Server Certificate (optional):** some organizations are eligible to obtain an SSL certificate from EPA. This certificate can be installed on the Node server to support hosting of https traffic.

Please contact the Node Helpdesk at: nodehelpdesk@epacdx.net or 1-(888)-890-1995 to obtain this information. There will be some forms that must be filled out to obtain this information.

After the above information is obtained, you can then follow these step-by-step instructions for initial Node 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.

Figure 6: Node.Administration Login

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).

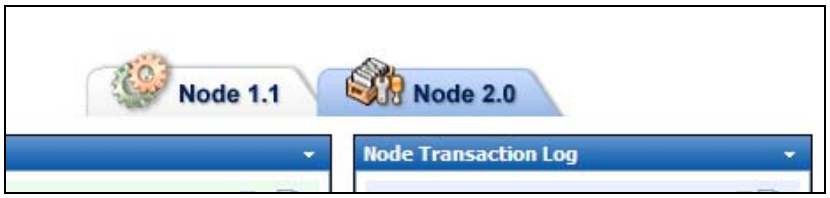

Figure 7: Node.Administration Dashboard

3.) Locate the 'Node Configuration' web part on the dashboard.

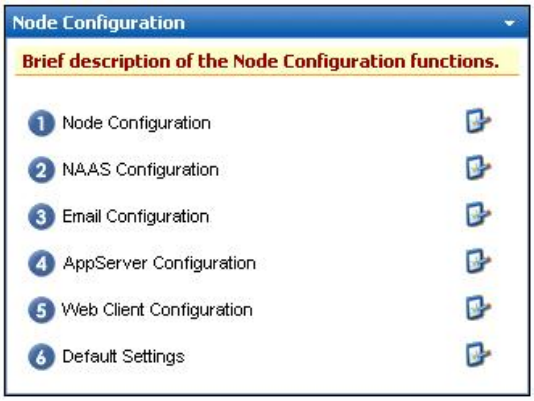

Figure 8: Node Configuration Webpart

You can navigate through each configuration option by selecting on the  icon.

4.) Proceed to option 1, *Node Configuration*, and modify the following configuration settings:

- **Node Identifier:** 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. *(Note: this setting needs to be set on both the Node 1.1 and Node 2.0 tabs)*

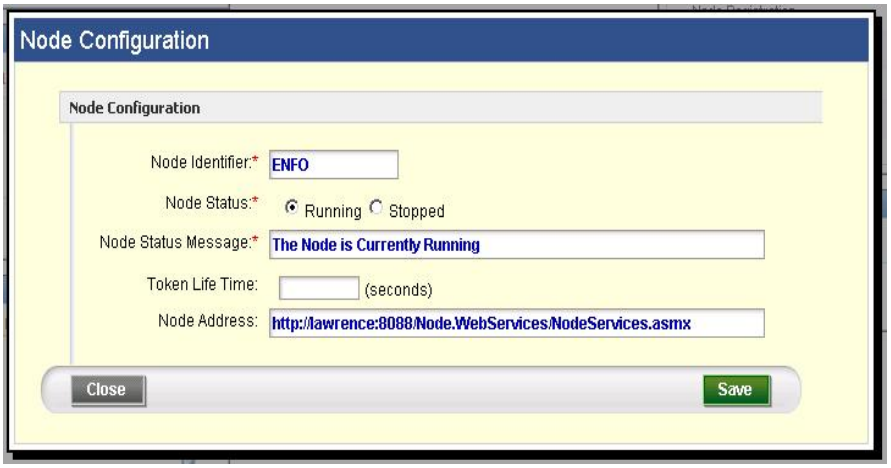

Figure 9: Node Configuration Settings

Click the Save button when you are done.

5.) Proceed to option 2, *NAAS Configuration*, and modify the following NAAS account information:

- **Node Administrator Name:** Node Administrator's Full Name
- **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.
- **Node Administrator Password:** *NAAS assigned Password*
- **NAAS Authentication Server Address:**
    i. **Test Server:** https://naas.epacdxnode.net/xml/auth.wsdl
    ii. **Production Server:** https://cdxnodenaas.epa.gov/xml/auth.wsdl
- **NAAS User Management Server Address:**
    i. **Test Server:** https://naas.epacdxnode.net/xml/usermgr.wsdl
    ii. **Production Server:** https://cdxnodenaas.epa.gov/xml/usermgr.wsdl
- **NAAS Policy Management Server Address:**
    i. **Test Server:** https://naas.epacdxnode.net/xml/policy.wsdl
    ii. **Production Server:** https://cdxnodenaas.epa.gov/xml/policy.wsdl

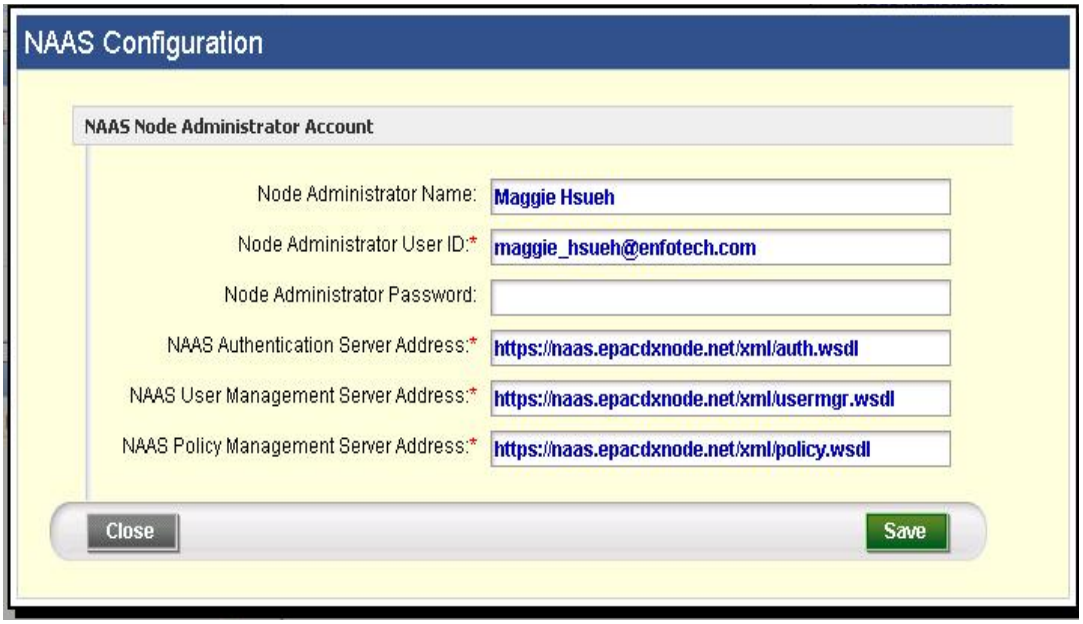

Figure 10: NAAS Configuration Settings

Click the Save button when you are done.

6.) Proceed to option 3, *Email Configuration*, and modify the email server and account information with the appropriate configuration information

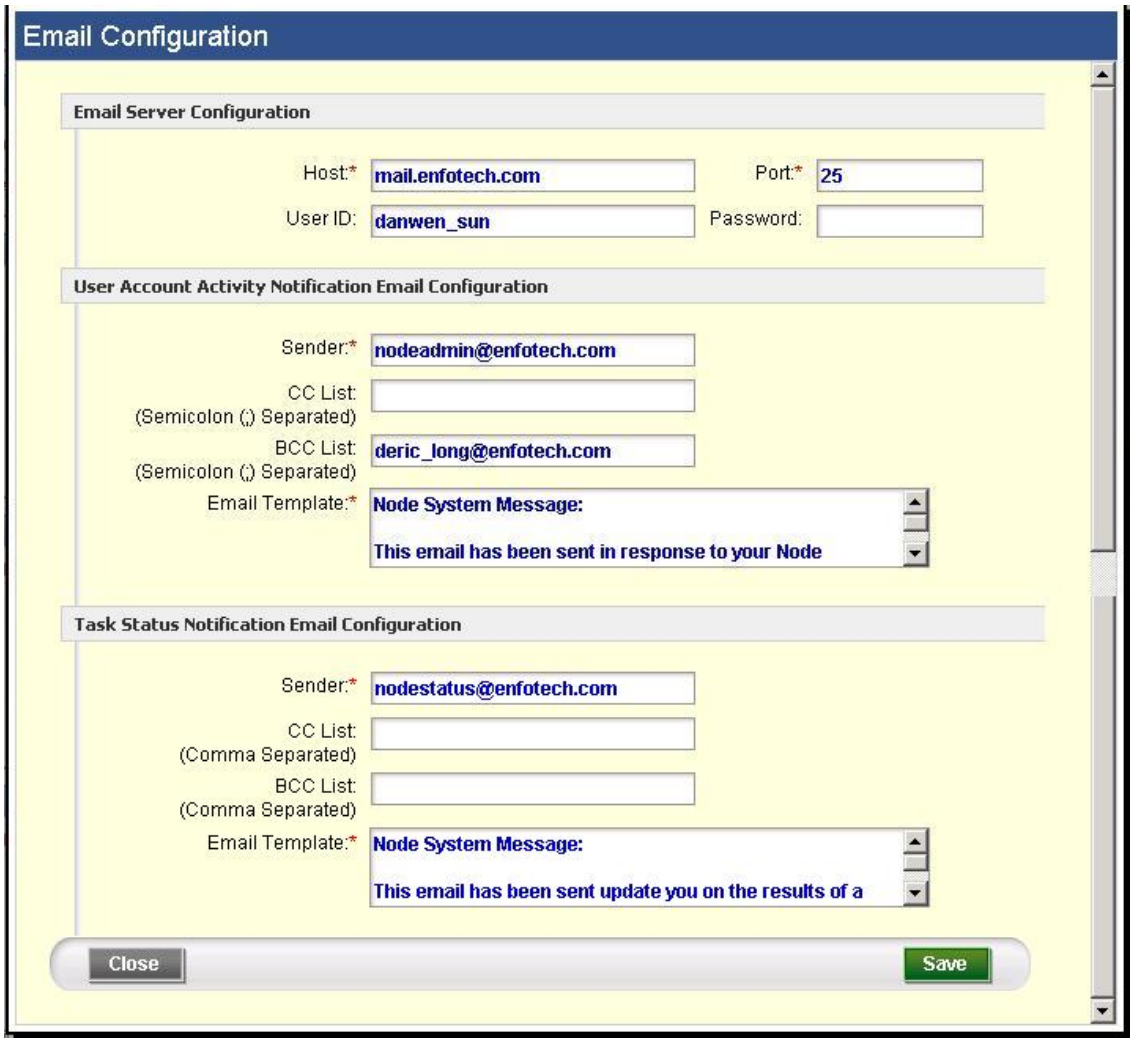

Figure 11: Email Configuration Settings

Click the Save button when you are done.

7.) Proceed to option 4, *AppServer Configuration*, and modify the application server with the appropriate server configuration settings. Check the Proxy check box and fill in the appropriate information if a proxy is used. In the second section, select the level of logging for each application and the full directory path to the log files (this is on the app server) including the file names.

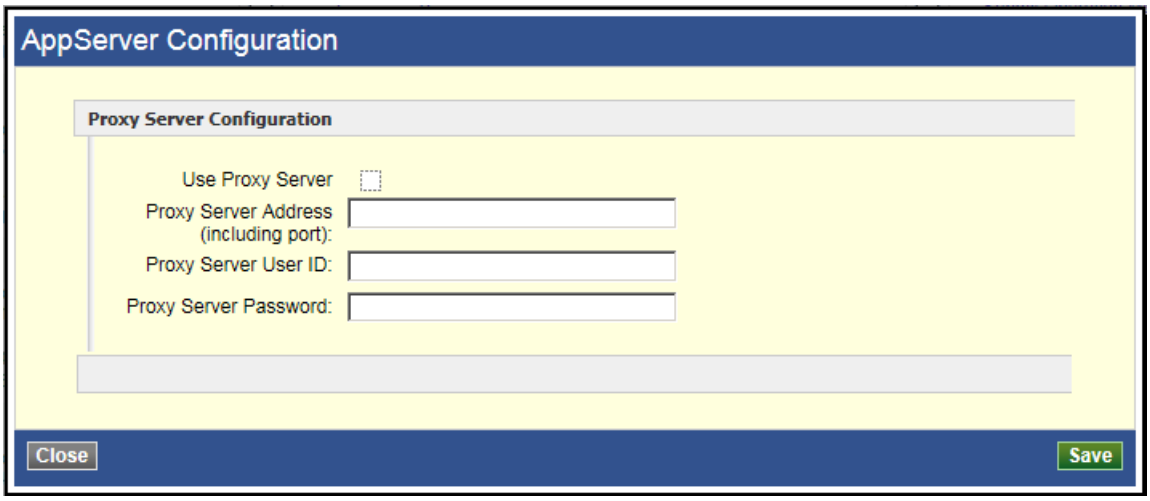

Figure 12: AppServer Configuration Settings

Click the Save button when you are done.

8.) Proceed to option 5, *Web Client Configuration*, and modify the web client URL settings. The URLs in this list are those that you wish to appear in the drop down of your Node.Client application.

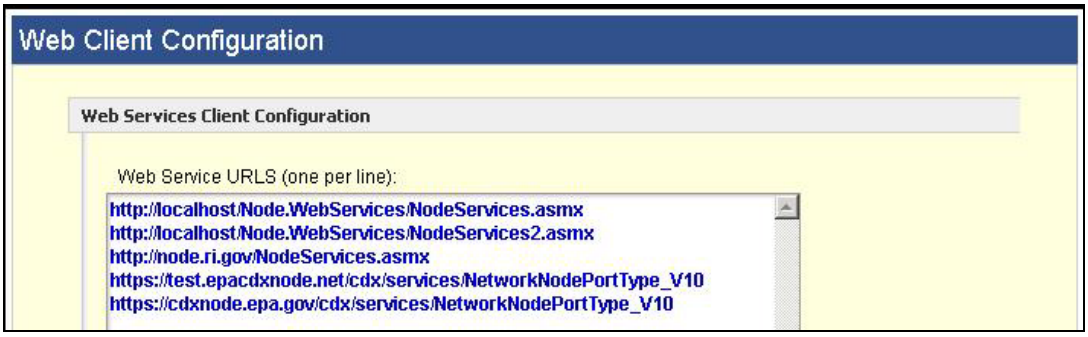

Figure 13: Web Client Configuration Settings

Click the Save button when you are done.

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 **Hello Message** textbox and click the **Node Ping** button. The screen should refresh and the Node Ping Result should be "hello", as shown here:

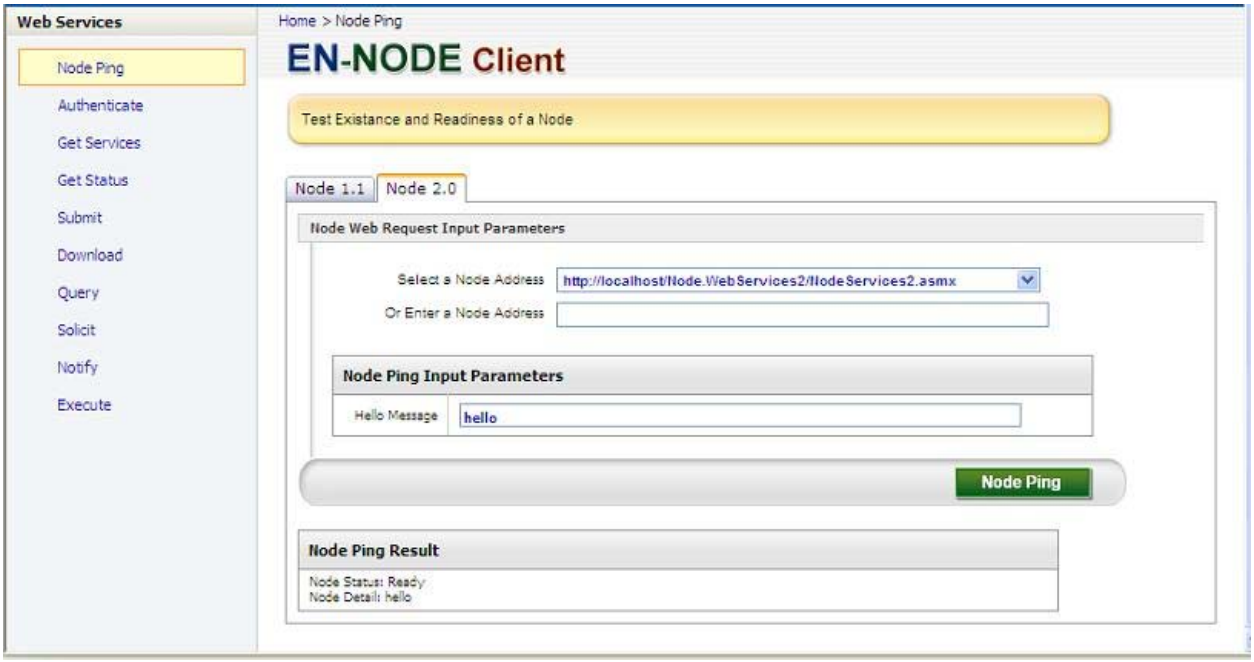

Figure 14: Node.Client Node Ping Test

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:

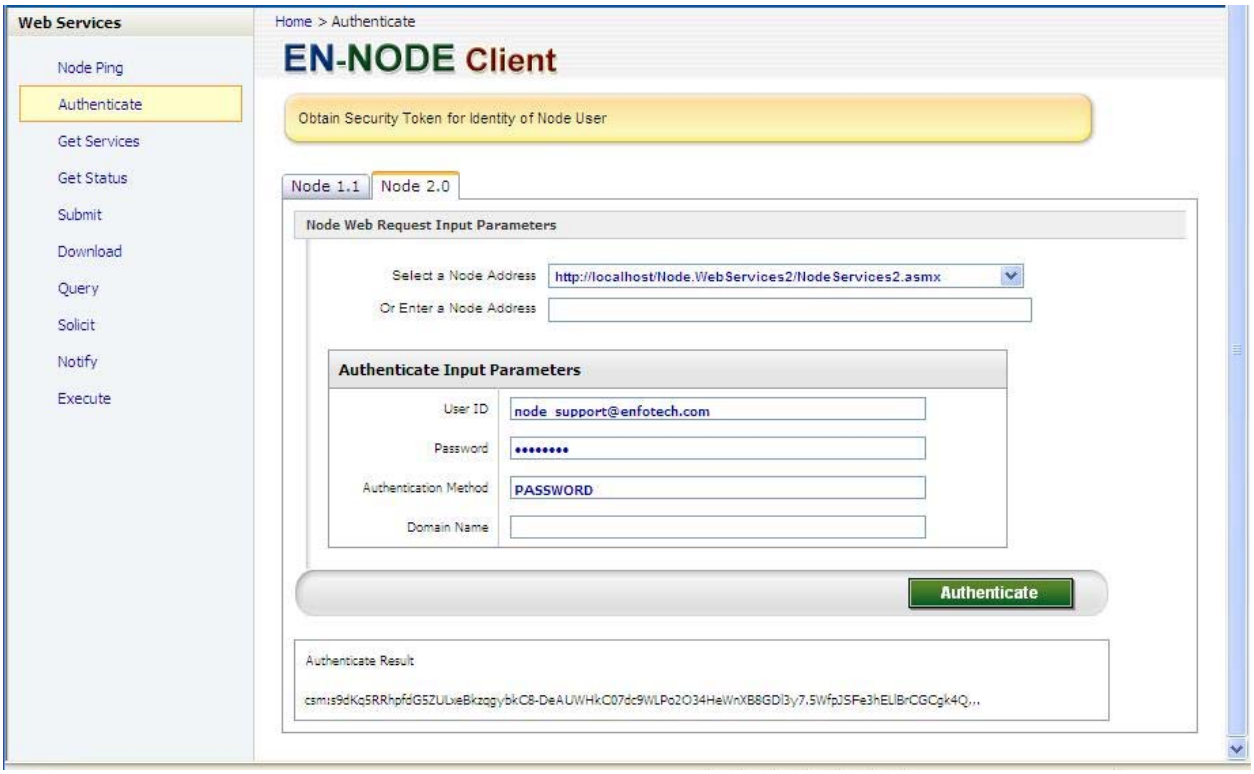

Figure 15: Node.Client Authenticate

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 “NCT” from **Request** 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:

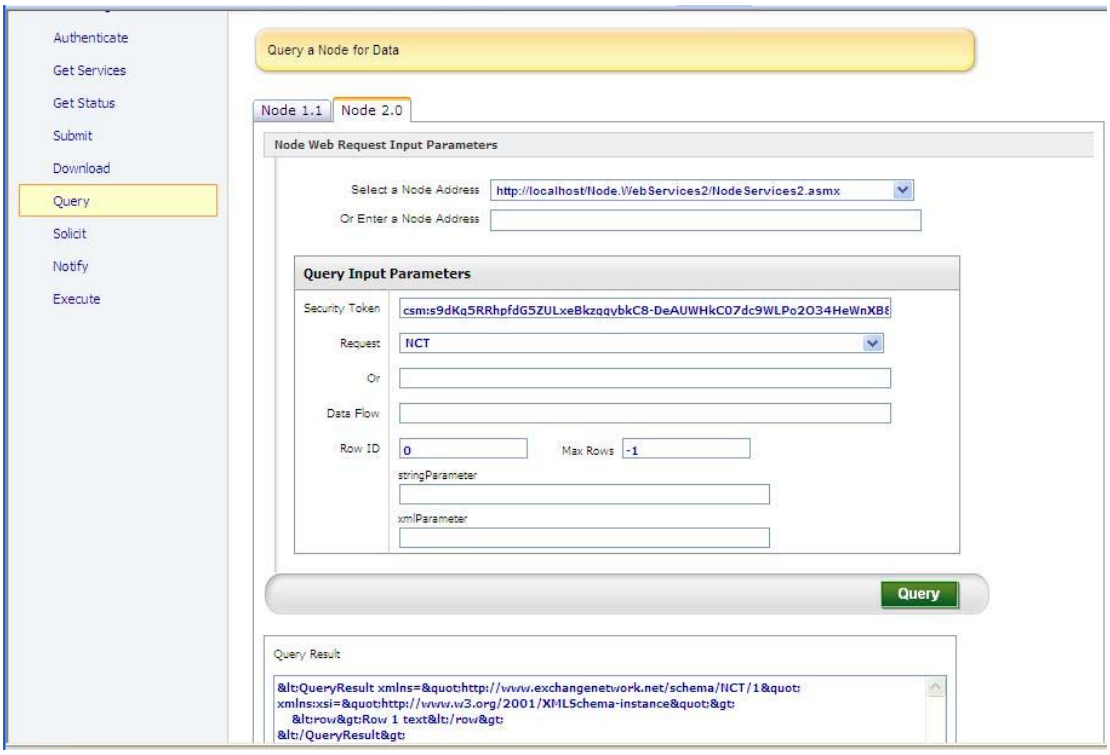

Figure 16: Node.Client Query

Appendix A: Database Connection Formats

The italicized strings are to be replaced with their actual values:

a. For Oracle Database Server:

"Data Source=*SID*;User ID=*user*;Password=*pwd*"

Note: Oracle Client must be installed on the app server and there must be a TNS entry that matches the SID name that the connection string refers to.

b. For MS SQL Server:

"Server=*server*;Database=*database*;UID=*user*;PWD=*password*"

c. For MS SQL Server Express Edition:

"Server=*server**SQLEXPRESS*;Database=*database*;UID=*user*;PWD=*password*"

Appendix B: Troubleshooting Guide

1. Any time the *Node.TaskHandler.exe.config* file or *TaskService.exe.config* file is modified, the TaskServices will need to be uninstalled and reinstalled from the Node.TaskServices folder by selecting on the appropriate .bat files (TaskServiceInst.bat or TaskServiceUninst.bat), as instructed below:
    a. Proceed to the task server (where the Node.TaskServices component is installed)
    b. On the task server, stop the *TaskServices* service
    c. Proceed to the *Node.TaskServices* folder on the server
    d. Locate and select on the *TaskServiceUninst.bat* file (this will uninstall and remove the TaskServices service from the server)
    e. Locate and select on the *TaskServiceInst.bat* file (this will install the TaskServices service on the server)
2. The Node.Administration will need to be configured according to the appropriate server/website settings (i.e. Web Client address, Node address, email server, etc.)
3. If, when you run the TaskServiceInst.bat file when installing the Task Service, it does not run successfully. Please try the following:
    a. Right-mouse-click and choose “Run As Administrator”
    b. There have been reports that when selecting “Run As Administrator”, the batch file has difficulty finding the machine’s .NET Framework directory. To avoid this issue, please open the TaskServiceInst.bat file in a text editor, find the following text: **%windir%\Microsoft.NET\Framework\v2.0.50727**

       and modify to match the actual .NET Framework directory on your machine.

       (This may only be an issue for deployments on Windows Server 2008 64 bit)