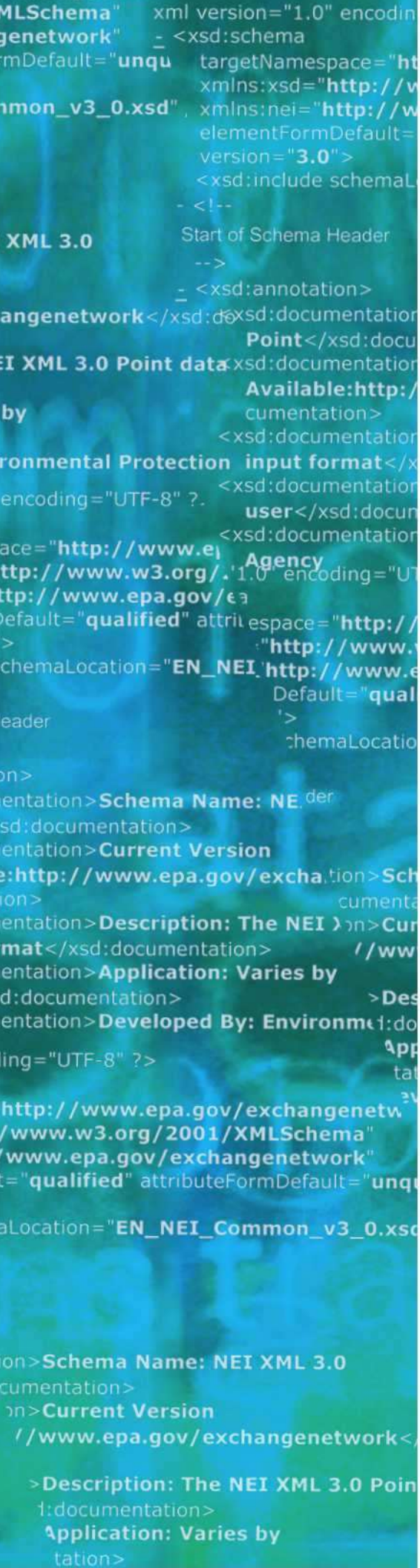

OpenNode2

Installation Guide (.NET)

Revision Date: 11/08/2017

Applies to .NET OpenNode2 v4.0.8

Prepared By:

4386 SW Macadam Ave, Suite 101
Portland, OR 97239
(503) 675-7833

Environmental Information exchange Network

Revision History

Date	Modification Description
4/2009	Initial version
9/2009	Added Node upgrade instructions.
3/8/2010	Added Configuration steps and steps for IIS 7+, Windows 7/2008, added steps for configuring NCT and Windsor plugins.
5/31/2011	Improved Node upgrade instructions in Appendix A
9/10/2012	Updated to require .NET Framework 3.5 for OpenNode2 v2.5
6/10/2013	Added installation steps for REST endpoint
10/1/2013	Fixed incorrect database script name. Other minor corrections
10/16/2013	Added new Step 7 to Appendix A – run database upgrade script
5/6/2014	Added IIS 7.5+ specific instructions.
7/17/2014	Fixed document formatting error in Overview section
9/29/2014	Update Node upgrade instructions in Appendix A to remove need to re-upload plugins.
3/3/2017	Updated to require .NET Framework 4.6.2 for the latest version of OpenNode2. Remove support options for IIS 7.0 and earlier
9/21/2017	Fixed wording in App Pool Config section
10/5/2017	Clarified installation instructions.
11/8/2017	Added ISAPI/CGI Restrictions step in upgrade instructions.

Table of Contents

Document Overview

This document describes the installation and configuration of the Exchange Network OpenNode2 open-source Node for .NET. The primary audience is a System Administrator or Deployment Specialist experienced with installing and managing .NET Web applications. The sections entitled "Relational Database" and "Install the Node Administration Database" are intended for Database Administrators.

The sections titled "Configure IIS Application Pools and Virtual Directories" and "Install and Configure the Node Orchestration Service" apply specifically to installing and configuring OpenNode2.

The main steps for installing OpenNode2 described in this document are:

- Perform prerequisite server software installations
- Install the OpenNode2 database
- Install the four IIS components
    - Node Administration Application
    - Node Service v1.1 Endpoint
    - Node Service v2.0 Endpoint
    - Node REST Endpoint
- Install and configure the Node Orchestration Service (NOS)
- Test the Installation.

Assumptions

Single Server Deployment

Although the OpenNode2 architecture allows for deploying the component applications across multiple physical machines, for clarity of this document, these guidelines assume that all components will be installed on a single machine, even when deploying to a managed cluster of application servers.

Because of the additional complexity inherent in a distributed environment, and the potential variability in configuration, additional assistance should be sought if installing in these environments. Please see the Exchange Network Website for additional support information.

Network Connectivity

Due to the Service Oriented Architecture (SOA) of the Exchange Network, and due to the Node's dependency on external connectivity to the Network Authentication and Authorization Services (NAAS), this document assumes that the server on which the Node will be deployed has already been configured on the local network and is accessible from the Internet.

Installation Prerequisites

Before proceeding with the OpenNode2 installation and configuration, ensure that the components that follow are installed on the physical server machine. While some steps can be done at the same time that the Node is installed, some steps should be taken prior to installation in order to allow for any delays in satisfying the requirements. Those steps that can occur at the same time as the Node installation are noted.

Hardware Requirements

A Pentium III processor is required at a minimum, though faster processors are preferred.

The minimum RAM requirement is 2 gigabytes. All programs running on the server must be considered when considering memory requirements.

The minimum hard drive space required is 2 GB. Again, considering the programs and file processing running on the server, additional hard drive space is likely required.

Software Requirements

Operating System

Microsoft Windows Server 2008 R2 or later operating systems with latest Windows updates installed are acceptable for OpenNode2.

Relational Database

OpenNode2 uses a relational database as its metadata repository. All metadata about the Node configuration and its activity is stored in that database. You will need to have database administrator rights to install the Node database. Alternatively, if the database has already been created, you will only need an account with an Owner role to this database.

If SQL Server Express is used, the 'Microsoft SQL Server Management Studio Express' application can be downloaded from the Microsoft Website to provide a GUI interface to the SQL instance.

If installing a dedicated SQL Server instance for the Node, the following parameters should be set:

- Feature Selection: Database Services > Data Files: C:\Data
- Instance Name: Named Instance: Node
- Authentication Mode: Mixed Mode: Enter a valid Windows password to use for the 'sa' (system account). The second option is to use standard Windows Authentication for database access.
- Configuration Options:
    - Enable User Instances: UN-CHECKED/UN-SELECTED
    - Add user to the SQL Server Administrator role: CHECKED/SELECTED

The default settings for all other installation options can be accepted.

If Oracle i9 or later is used, the Oracle SQL Developer application can be used to interface with the database.

Microsoft Internet Information Services (IIS)

IIS 7.5 or newer must be installed for the OpenNode2 Web components. The following table lists the versions of IIS and .NET Framework shipped with each Windows version. The .NET Framework version can be updated to a newer version on any of the operating system/IIS versions below.

Operating System	IIS Version	.NET Framework Version
Windows Server 2008 R2, Windows 7	IIS 7.5	v3.5
Windows Server 2012, Windows 8	IIS 8.0	v4.0
Windows Server 2012 R2, Windows 8.1	IIS 8.5	v4.5.1
Windows Server 2016, Windows 10	IIS 10.0	v4.6.2

Microsoft .NET Framework 4.0 (OpenNode2 v2.7 and older)

Older versions of OpenNode2 require the use of the .NET Framework 4.0. To verify that .NET v4.0 is installed, perform the following steps:

1. Open the Internet Information Services (IIS) Manager
2. Browse to 'SERVER_NAME' -> Application Pools
3. Double-click the default Application Pool
4. Verify that .NET Framework 4.0.x is available in the list of options

If needed, setup files can be downloaded from Microsoft Website at:

http://www.microsoft.com/en-us/download/details.aspx?id=17851

The .NET Framework installation can be done before or at the same time as the Node installation.

Note: *Oracle database drivers may also require .NET Framework v3.5. See Oracle product documentation for details.*

Microsoft .NET Framework 4.6.2 (OpenNode2 v4.x and newer)

OpenNode2 v4.0 and above requires the use of the .NET Framework 4.6.2. Setup files to perform an update to .NET Framework 4, 4.5, 4.5.1. 4.5.2, 4.6, or 4.6.1 can be downloaded from Microsoft Website at:

https://www.microsoft.com/en-us/download/details.aspx?id=53345

Note: *Oracle database drivers may also require .NET Framework v3.5. See Oracle product documentation for details.*

ODBC for .NET 2.0 (optional)

If the OpenNode2 will be connecting to a data source other than Microsoft SQL Server or an Oracle database, the .NET 2.0 ODBC drivers will need to be installed. ODBC for .NET 2.0 can be downloaded from the Microsoft Website.

SSL Certificate (production installations only)

A 128-bit encrypted SSL certificate must be installed and the OpenNode2 URL must be listening on port 443. The SSL certificate must be externally-facing to Internet clients. If the target server/environment is a Test/Development server, it is not necessary to get a commercial SSL certificate. The Exchange Network certificate (issued by the CDX helpdesk), a self-signed certificate, or no SSL certificate at all is acceptable.

To install the SSL certificate, perform the following steps in the IIS Manager:

IIS 7.5+

1. Double click the certificate file (.pfx)
2. Follow the steps in the Certificate Import Wizard

User Accounts

NAAS Node Account

An account *for each environment under which OpenNode2 will run* (e.g., "Production," "Test," etc.) will need to be set up with the EPA's Network Authentication and Authorization Services (NAAS). This account is typically named node@*NODE_DOMAIN*.com, where "*NODE_DOMAIN*" is the same domain that the Node URL uses. This account will be used by OpenNode2 to perform all Network interactions, such as authentication, submitting, and downloading.

The following steps assume that an onsite Node administrator has an active NAAS account for administering their Node.

1. Browse to https://naas.epacdxnode.net/ (Test NAAS) or https://cdxnode.epa.gov/usr (Production NAAS)
2. Click the 'Login' link in the left frame and log in using the Node administrator account. If you do not have a Node administrator account, contact the CDX helpdesk to have one created.
3. Click the 'Add User' link in the left frame and create a user with the following parameters:
   - User ID: node@myagency.gov
   - Password: (*an acceptable NAAS password*)
   - User Type: operator
   - Affiliate: *(choose your agency NodeID)*

Installation Steps

Unblock Zip Archive

By default, when the **DotNET OpenNode2 v<*version number*>.zip** archive is downloaded from the Internet, the file will be saved with a "blocked" attribute. This can be confirmed by selecting **Properties** and viewing the **General** tab. At the bottom of the tab, if the file is blocked the following message will be shown next to Security:

This file came from another computer and might be blocked to help protect this computer.

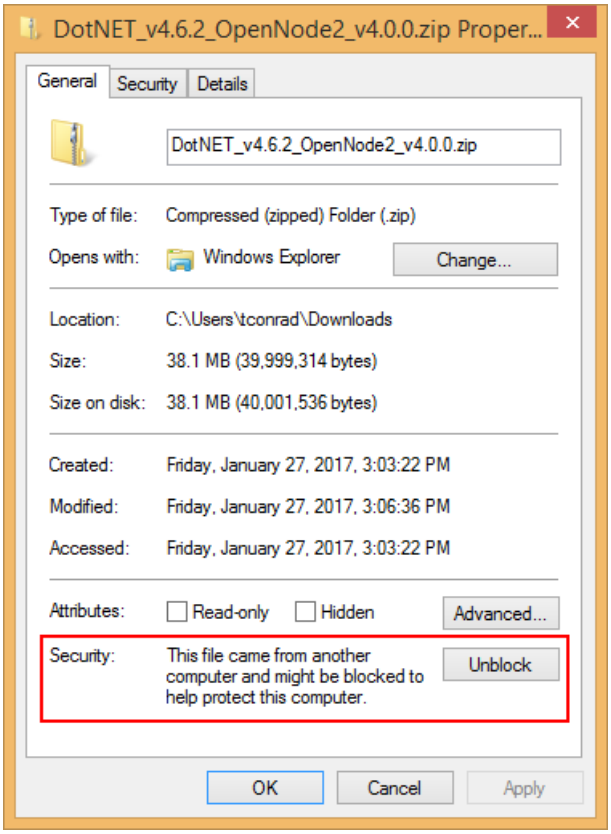

Figure 1: Security block message

If this message is shown, click **Unblock** to remove this attribute from the archive before extracting its contents.

Note: Some versions of Windows show a checkbox labeled "Unblock" instead of the button above. Simply check the Unblock checkbox and then click OK.

Extract Installation Files

Extract the files from the **DotNET OpenNode2 v<*version number*>.zip** archive to a designated parent directory on the server. Note that the archive contains only compiled code. For the purposes of the installation guide, it is assumed that the parent directory path is **C:\OpenNode2**.

The directory structure is as follows:

Path	Description
OpenNode2\Config	Location of the configuration settings for OpenNode2. Only Deployment.config contains user-configurable settings.
OpenNode2\Logs	Location of detailed OpenNode2 server logs
OpenNode2\Plugins	Location of all plugins uploaded through the Node Admin interface
OpenNode2\Repository	If documents are configured to be stored in the file system, they will be saved to this directory. Otherwise the Node database's NDocument table will be used.
OpenNode2\Server	The directory containing the .NET Windows NOS service that handles all node operations.
OpenNode2\Sql	Contains SQL scripts to create Node database objects
OpenNode2\Temp	Temporary storage of documents processed by the Node. This folder is cleaned up automatically by a NOS worker process.
OpenNode2\www	Directory containing four subfolders for the Node Admin, two SOAP Web Service endpoints, and REST endpoint. Each folder will be set up as a virtual directory.

Install the Node Administration Database

1. Create a database named **OpenNode2** on the target SQL Server or Oracle database.
2. Run the file **C:\OpenNode2\Sql\OpenNode2_DDL_*xxx*.sql** (where *xxx* is either SQL Server or Oracle) to create the tables and objects used for the main node data repository.
3. Open the file **C:\OpenNode2\Sql\OpenNode2_DATA_*xxx*.sql** in a text editor (where *xxx* is either SQL Server or Oracle). Edit the first INSERT statement to contain your NAAS admin account name. Edit the second INSERT statement to contain your NAAS runtime account name.
4. Run the **OpenNode2_DATA_*xxx*.sql** script to insert the accounts into the OpenNode2 Metadata database. The script also sets up the Node Certification Test (NCT) flow and services as well as the Windsor flow that contains utility services.
5. If necessary, create a database user account for OpenNode2 service to use. The account should have permission to read, insert, update and delete to all tables in the OpenNode2 database. The connection string containing user credentials will be set in the NOS configuration in a later step.

Configure IIS Application Pools and Virtual Directories

Create and Configure the Application Pool

.NET 4.0 and below:

The following steps pertain to configuring the application pool for the .NET 3.5 version of OpenNode2, **DotNET_v3.5_OpenNode2_v2.7.3.zip**.

IIS 7.5+:

1. In the Internet Information Services administrative console, create a new application pool named "OpenNode2". Set the Managed Pipeline Mode to **Classic** utilizing **.NET Framework v2.0** (i.e., **.NET CLR Version v2.0.NNNNN**). In Advanced Settings, change Identity from ApplicationPoolIdentity to NetworkService. Other default settings are acceptable. This application pool will be used by Node Admin and the SOAP Web Service endpoints.
2. In the Internet Information Services administrative console, create another new application pool named "OpenNode2_v4". Set the Managed Pipeline Mode to **Integrated** utilizing **.NET Framework v4.0** (i.e., **.NET CLR Version v4.0.NNNNN**). In Advanced Settings, change Identity from ApplicationPoolIdentity to NetworkService. Other default settings are acceptable. This application pool will be used by REST endpoint.

.NET 4.6.2 and above:

The following steps pertain to configuring the application pool for the .NET 4.6.2 version of OpenNode2, **DotNET_v4.6.2_OpenNode2_v4.0.0.zip**.

IIS 7.5+:

1. In the Internet Information Services administrative console, create a new application pool named "OpenNode2". Set the Managed Pipeline Mode to **Classic** utilizing **.NET Framework v4.0** (i.e., **.NET CLR Version v4.0.NNNNN**). In Advanced Settings, change Identity from ApplicationPoolIdentity to NetworkService. Other default settings are acceptable. This application pool will be used by Node Admin and the SOAP Web Service endpoints.
2. In the Internet Information Services administrative console, create another new application pool named "OpenNode2_v4". Set the Managed Pipeline Mode to **Integrated** utilizing **.NET Framework v4.0** (i.e., **.NET CLR Version v4.0.NNNNN**). In Advanced Settings, change Identity from ApplicationPoolIdentity to NetworkService. Other default settings are acceptable. This application pool will be used by REST endpoint.

In addition to configuring the application pools, ensure the IIS Server is configured to allow ISAPI and CGI extensions for ASP.NET v4.0.NNNNN:

1. Select the server name.
2. Open **ISAPI and CGI Restrictions**.
3. Select **Allowed** for **ASP.NET v4.0.NNNNN** for both **Framework** and **Framework64**.

Create and Configure Virtual Directories / Applications

Three virtual directories (or Applications in IIS 8.0+) will be created in this section; one for the Node Admin interface and one for the Node endpoint. Use the appropriate directions for the version of IIS used.

IIS 7.5+ Directions

Create the Node Admin Application

1. Open the Internet Information Services (IIS) Manager
2. Browse to 'SERVER_NAME > Sites > WEB_SITE_NAME' (e.g., Default Web Site)
3. Right-click and select 'Add Application'
4. Type the alias **Admin** and set the physical path to **C:\OpenNode2\www\Admin**
5. Change the application pool setting to **OpenNode2**
6. In IIS Manager's Features View, double click the Default Document icon. On the Default Document dialog, remove all default entries and add **login.aspx** to the list.

Create the Node Endpoint1 Application

1. Right-click and select 'Add Application'
2. Type the alias **Endpoint1** and set the physical path to **C:\OpenNode2\www\Endpoint1**
3. Change the application pool setting to **OpenNode2**

Create the Node Endpoint2 Application

1. Right-click and select 'Add Application'
2. Type the alias **Endpoint2** and set the physical path to **C:\OpenNode2\www\Endpoint2**
3. Change the application pool setting to **OpenNode2**

Create the REST Endpoint Application

1. Right-click and select 'Add Application'
2. Type the alias **RestEndpoint** and set the physical path to **C:\OpenNode2\www\RestEndpoint**
3. Change the application pool setting to **OpenNode2_v4**.

The virtual directories / applications are now properly configured.

Configure Directory Security

It is necessary to set directory security so that the NOS and Websites can access folders for writing temporary files, logs, and reading configuration settings.

***Note**: The default settings for Windows 7/2008+ and IIS 7.0+ do not require special folder permissions to operate correctly, however it may become necessary to follow these steps if file access errors occur or if the files are located in a directory that is not locally accessible to the NOS and/or Admin web application.*

Follow the steps below to configure directory security:

1. Browse to the **C:\OpenNode2\Config** directory.
2. Right click on the directory and choose **Properties**
3. Click the **Security** tab
4. Click **Edit**
5. Click **Add**
6. Grant the user assigned to the Application Pools (e.g., ASP.NET Machine Account – **ASPNET,** IIS AppPool\OpenNode2 and/or IIS AppPool\OpenNode2_v4) **Read**, **Read and Execute**, and **List Folder Contents** rights to the directory. Next, grant the Internet Guest Account **IUSR_<*MachineName*>** (or **IIS_IUSRS**) the same rights as outlined above.
7. Click OK to finish.

8. Perform the steps above for both the **C:\OpenNode2\Logs** directory and the **C:\OpenNode2\Temp** directory.

Install and Configure the Node Orchestration Service (NOS)

The Node Orchestration Service (NOS) is a Windows service responsible for processing all inbound node requests and executing scheduled tasks.

Setting the NOS Configuration Parameters

1. Open the deployment.config file in a text editor. The config files are located in the **C:\OpenNode2\Config** directory. Go through all settings and set to values that are appropriate for your environment. Appendix B has detailed information on each setting.

   Firewall Note: Depending on firewall configuration or if the Node is deployed on multiple servers, it may be necessary to open a port so that the Web server can communicate with the NOS using .NET remoting. The port number is set in the **Deployment.config** file within the **Config** directory. Look for the setting with the key "wnos.service.port".

2. Save and close the Deployment.config file.

Installing the NOS Service (IIS 7+)

3. Open a command prompt using the "Run as Administrator" option. Navigate to the OpenNode2 Server directory. Register the NOS service by executing the following command:

   .NET 4.0 and below

```
"%SystemRoot%\Microsoft.NET\Framework\v2.0.50727\InstallUtil.exe" /i
".\Windsor.Node2008.WNOSServiceApp.exe"
```

   .NET 4.6.2 and above

```
"%SystemRoot%\Microsoft.NET\Framework\v4.0.30319\InstallUtil.exe" /i
".\Windsor.Node2008.WNOSServiceApp.exe"
```

   The texts above can also be copied from the install_service.bat file located in the Server directory. Do not run the batch file directly as this will not work.

If the service needs to be uninstalled, a **uninstall_service.bat** file is provided. Follow the steps above that are appropriate for the IIS and .NET version used.

Configuring the NOS Service Account

4. Open the Windows Services management console and locate "OpenNode2 Node Orchestration Service." (The name and description of the service is set in the Deployment.config file.)
5. Right click on the service and click **Properties**.
6. **Optional step:** Configure the service to run under an account with additional needed rights.

   In most circumstances, the NOS can be left to run under the Local System account, but in cases where the database is using Integrated Security (for SQL Server), or if the NOS must use network resources (such as file shares), then it will be necessary to run the NOS service under an account with additional permissions. Follow the steps below to change the user account.
7. Change the startup type to **Automatic**.
8. Save changes and close.
9. Right-click on the service and click **Start**.

> **Note:** The service will start and then stop if there is a problem with the configuration in Deployment.config. Check the Windows Application Event Log for a description of error if this occurs. Common errors are invalid connection string, insufficient database permissions, or an incorrect node.home path. In some cases you may need to uninstall the service to stop the service from hanging.

Test the Installation

This section contains information on ensuring that the installation was successful.

Testing the Node Orchestration Service

Follow these steps to verify that the Node is installed and functioning properly.

1. The first indication that the NOS is operating successfully is that the service starts successfully in the final step of the previous section.
2. Next, open Windows Event Viewer and view the Application Event Log. The most recent events should be from NOS. The most recent log entry should contain a message stating that the service was started successfully, as shown below:

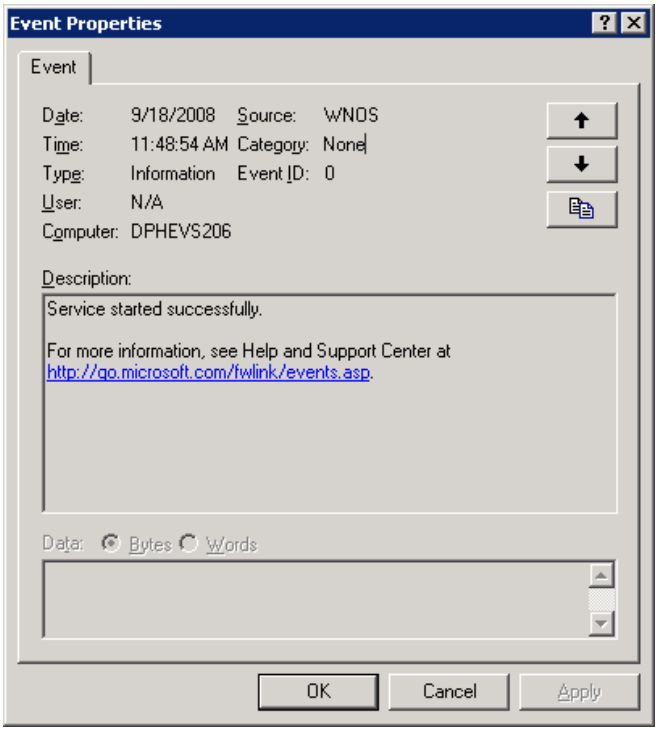

Figure 1. Windows Event Log

3. Lastly, examine the **OpenNode2.log** file located in the **C:\OpenNode2\Logs** directory. This log file will contain an entry each time the service scheduler interval fires. It also will contain a log of any errors that may occur.

Testing the Node Administration Application

Simply navigate to the URL where the Node Administration Utility is hosted. It is recommended to do this from the machine where the NodeAdmin is installed since errors will not display in the browser from remote connections. If the directions in this guide are followed, the address will be **http://localhost/Admin/login.aspx**.

The screen should appear as follows:

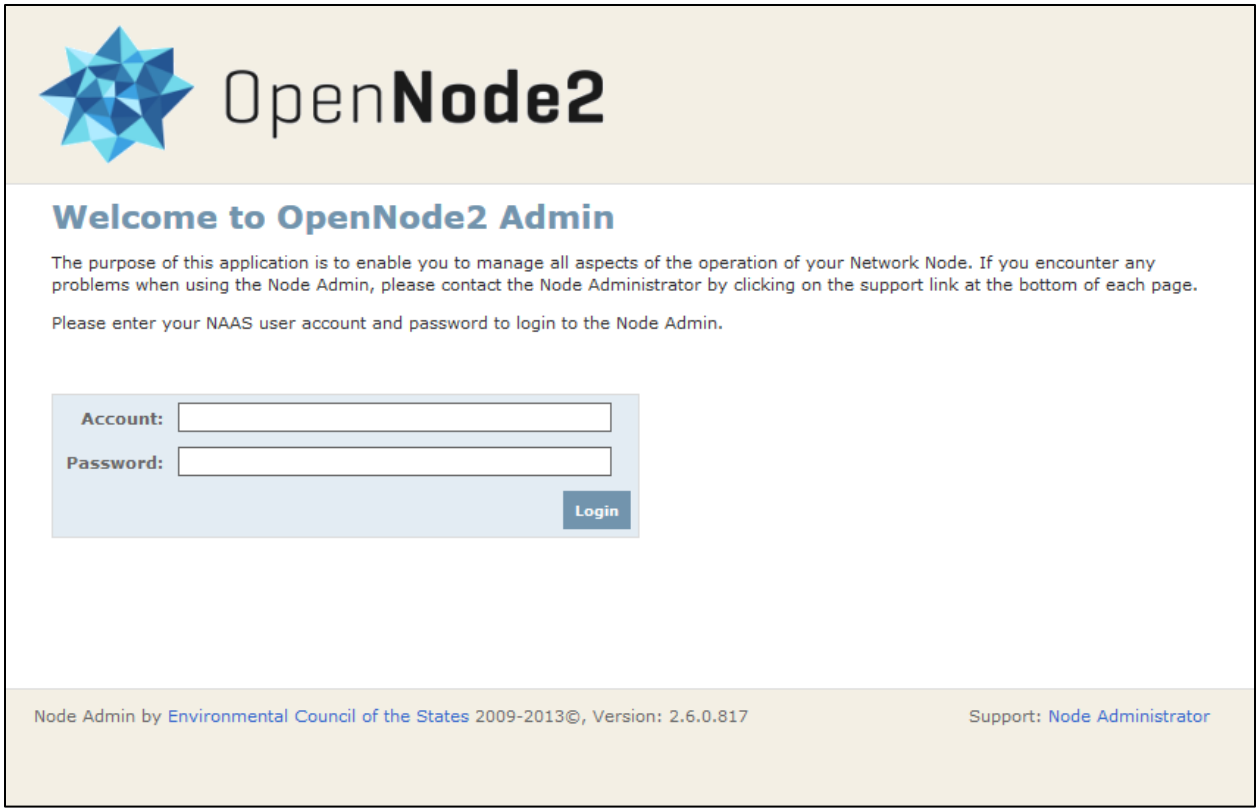

Figure 2. Node Admin Login Page

Log in using the administrator account that was inserted into the database as specified above. Use the NAAS password for the account.

If an error message appears upon attempting to log in, this may indicate one of the following:

- The remoting port may not have been opened. See the Firewall Configuration section for more information.
- The server may not be able to access the Internet to validate the credentials against NAAS. Ensure that the Node server has Web access.

Testing the Node Service Endpoints

Navigate to the URL where the Node Service is hosted. It is recommended to do this from the machine where the Node Service is installed since errors will not display in the browser from remote connections. If the directions in this guide are followed, the address will be: **http://localhost/Endpoint1/ENService11.asmx**. Also test the v2.0 endpoint at **http://localhost/Endpoint2/ENService20.asmx**

The following image shows the page that should be displayed at the service v1.1 endpoint URL (a similar screen should also be displayed at the service v2.0 endpoint URL):

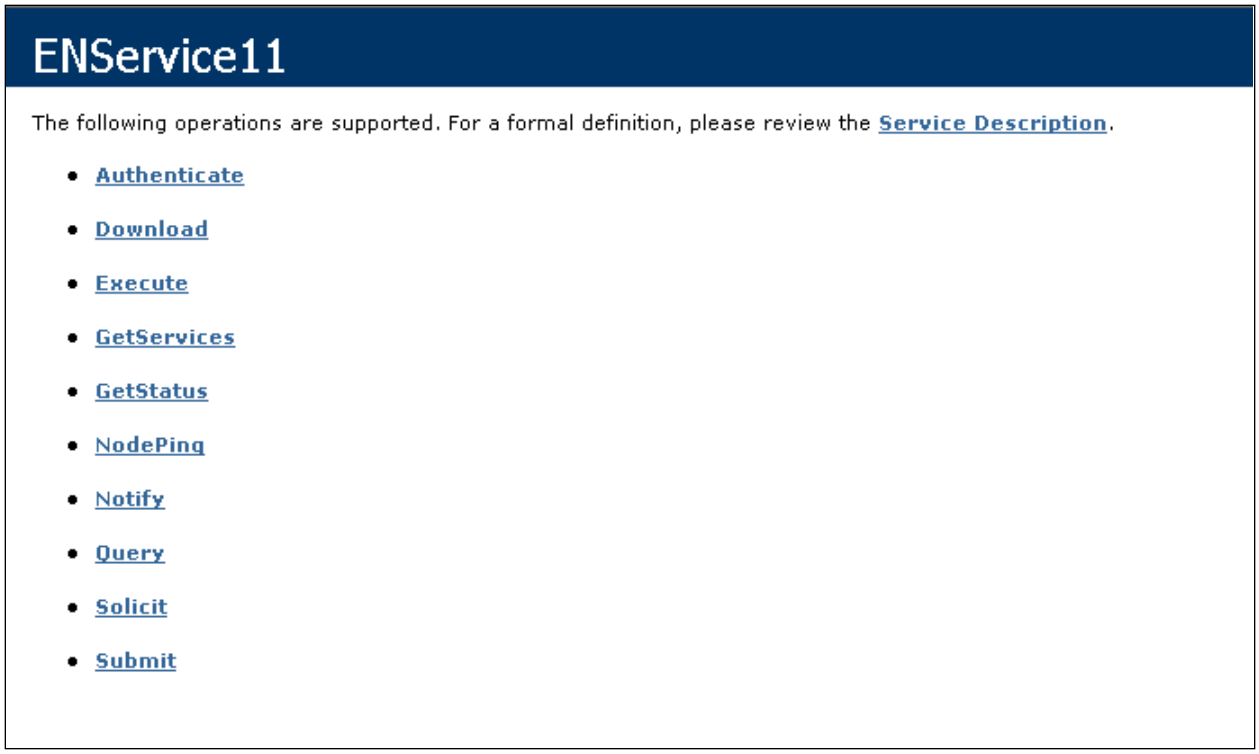

Figure 3. Service 1.1 Endpoint

After testing using localhost, it is recommended to ensure that the service endpoints are exposed externally to the Internet.

Appendix A - Upgrading OpenNode2

Node and Plugin Versioning

The OpenNode2 software versioning scheme is:

[NodeMajorVersion].[NodeMinorVersion].[Revision].

Plugins must be installed on the OpenNode2 software with the same [NodeMajorVersion] and [NodeMinorVersion] as the plugin. Revision numbers indicate incremental bug fixes. Higher numbers always indicate a newer revision.

Each new OpenNode2 version release includes all new plugins. While some older plugins may be compatible with the newer OpenNode2, it is highly recommended to use the plugins that are bundled with the installation zip file.

Warning: Any custom-developed plugins that are not included with the download should not be used with the 4.0+ versions of the node until they have been recompiled for use with .NET 4.6.2!

Step-by-Step Upgrade Instructions

Note: *A .NET Framework upgrade may be needed! Please review the relevant sections of this guide for .NET Framework information.*

The upgrade process consists primarily of backing up the current configuration, replacing the application files, restoring the configuration, and updating plugins (if needed).

Step 1: Shut Down the Node Orchestration Service (NOS)

1. Open the Windows Services management console and locate "OpenNode2 Orchestration Service."
2. Right-click on the service and click **Stop**.
3. **OPTIONAL:** If you are upgrading from a version 2.* node to a version 4.* node, *uninstall the NOS completely by running the command in Server\uninstall_service.bat in a command console with administrative rights.*

Step 2: Backup the NOS Configuration Parameters

4. Locate the deployment.config file in the designated parent directory (e.g., **C:\OpenNode2\Config**).
5. Save a copy of this deployment.config file to a new location outside of the designated parent directory (e.g., **C:\OpenNode2**), which will be accessible following the upgrade.

Step 3: Extract/Replace the Installation Files

6. Extract the unblocked files from the **DotNET OpenNode2 v**<***version number***>**.zip** archive to the existing designated parent directory (e.g., **C:\OpenNode2**) on the server. This extraction will replace/overwrite the existing files in the designated parent directory. Note that the archive contains only compiled code.

EXTREMELY IMPORTANT: *Be sure to unblock the zip archive as described in the normal installation steps above or Windows will prevent the code from executing properly.*

Step 4: Restore the NOS Configuration Parameters

7. Locate the backed up copy of the deployment.config file which was saved in Step 2 above.
8. Locate the new deployment.config file in the designated parent directory (e.g., **C:\OpenNode2\Config**).
9. Replace the new deployment.config file in the designated parent directory (e.g., **C:\OpenNode2\Config**) with the backed up copy of the deployment.config file which was saved prior to the upgrade.

Step 5: Reinstall and/or Restart the NOS Orchestration Service

10. If you uninstalled the Node Orchestration Service to move from version 2 to version 4, you must re-install the version 4.* service as described above.
11. Open the Windows Services management console and locate "OpenNode2 Orchestration Service."
12. Right click on the service and click **Properties**.
13. If necessary, configure the service to run under an account with administrative rights.
14. If necessary, change the startup type to **Automatic**.
15. Save changes and close.
16. Right-click on the service and click **Start**. Note that the service will not start if the database connection string has not been configured properly. You may need to uninstall the service to stop the service from hanging (see step 1 above).

Step 6: Ensure the IIS Server is configured to allow ISAPI and CGI extensions for ASP.NET v4.0.NNNNN:

17. Open IIS Manager
18. Select the server name.
19. Open **ISAPI and CGI Restrictions**.
20. Select **Allowed** for **ASP.NET v4.0.NNNNN** for both **Framework** and **Framework64**.

Step 7: Run Database Update Script (Upgrades to v2.6 and Newer only!)

Starting with v2.6, OpenNode2 supports configuring different NAAS accounts to be used by different schedules. This is done by adding "Endpoint user" accounts on the Configuration tab within the OpenNode2 Admin application.

To enable this new feature, run the script named OPENODE2_2.0-SQL-UPDATE_001.sql (SQL Server) or OPENODE2_2.0-ORA-UPDATE_001.sql (Oracle) included in the installation file download.

Lastly, access the OpenNode2 Admin application and log in to verify that the upgrade was successful. The version number should appear on the footer of every page indicating the latest installed version number.

Appendix B – Configuration Settings

This section provides a description of each of the settings in Deployment.config located in the **C:\OpenNode2\Config** directory.

NOTE: Any changes to the service configuration file will require that the NOS be restarted for the changes to take effect. Deployment.config is located in the **C:\OpenNode2\Config** directory.

Setting Key	Description
node.home	Root directory where the OpenNode2 runtime files are located. For example "C:\OpenNode2".
node.is.production	Set to true or false. False will trigger node to use the test NAAS endpoint for obtaining and validating EN security tokens.
node.is.demo	Not used.
node.(north/south/east/west)	"Bounding box" coordinates for the geographic area covered by the node's organization. Used to populate ENDS.
node.organization.name	The organization that is hosting the node. Used to populate Node Admin UI footer text.
wnos.service.host	The IP address, server, or DNS name of the machine hosting the NOS.
wnos.service.port	The port used by the WNOS to listen for requests from the Node Admin or Service endpoints.
wnos.service.admin.interface.url	The address of the Node Administration Utility. Used to build the URL of transaction history sent in notification emails.
wnos.service.description	The description of the NOS service that appears in the Windows Services list.
wnos.service.display.name	The display name of the NOS service that appears in the Windows Services list.
wnos.service.service.name	The name of the NOS service that appears in the Windows Services list.
wnos.service.dependencies	The name of any other services that need to start first. For example, MSSQL$SQLEXPRESS if SQL Express is used as the node database and it is not set to start automatically.
wnos.data.provider	The fully qualified name of the data provider used to access the Node database.
wnos.data.connection	The connection string for the Node database.
wnos.document.manager.fs.repo.path	Path to the Node repository for documents, if the file system is configured for use.
wnos.temp.dir.path	Path to the Node temporary directory. The temp folder is used to store documents temporarily before they are compressed.

Setting Key	Description
wnos.naas.node.id	The Node identifier used when interacting with the NAAS user manager for adding, editing, and deleting NAAS accounts.
wnos.naas.node.admin.email	The email address displayed on the Node Admin UI and used in notification email messages.
wnos.naas.user.admin	The username of the NAAS account used to perform administrative functions (user management).
wnos.naas.user.admin.password	The password of the NAAS account used to perform administrative functions (user management).
wnos.naas.user.admin.domain	The value to be passed in the NAAS "domain" parameter when performing authenticate operations. "Default" is the default value.
wnos.naas.user.runtime	The username of the NAAS account used to perform Node primitive operations.
wnos.naas.user.runtime.password	The password of the NAAS account used to perform Node primitive operations.
wnos.naas.user.runtime.domain	The value to be passed in the NAAS "domain" parameter when performing authenticate operations. "Default" is the default value.
wnos.util.crypto.key	Private key used to encrypt certain values in the Node database.
wnos.proxy.address	If a proxy server is used, the address of the proxy server. Leave blank if no proxy server is used.
wnos.proxy.port	If a proxy server is used, the port of the proxy server. Leave blank if no proxy server is used.
wnos.proxy.bypassOnLocal	Set to either True or False. Indicates whether the Node should bypass the proxy server for requests that originate locally.
wnos.proxy.credential.useDefault	If a proxy server is used, indicates whether the proxy server's default credentials should be used.
wnos.smtp.host	The IP address, machine name, or DNS name of the SMTP server used for sending notifications.
wnos.smtp.port	The port of the SMTP server. Usually "25."
wnos.smtp.fromAddress	The email address to provide to the proxy server if the message must originate from a trusted account.
wnos.smtp.deliveryMethod	Default to "Network."
wnos.smtp.enableSsl	Set to True or False. Indicates whether SSL should be used for SMTP operations.
wne.request.ipholder	Default to "REMOTE_HOST"
wne.request.ipdefault	The default IP address to use for activity logging if one is not found or provided.

Setting Key	**Description**
wne.request.version20	Identifier used internally and for logging to identify operations that originate from the v2.0 service endpoint.
wne.request.version11	Identifier used internally and for logging to identify operations that originate from the v1.1 service endpoint.
wne.version11.url	Endpoint 1.1 URL.
wne.version20.url	Endpoint 2.0 URL.