EN-NODE ADMINISTRATOR'S GUIDE (JAVA VERSION)

Version: 2.8

August 26, 2015

1368 How Lane
North Brunswick, New Jersey 08912
www.enfotech.com

Revision History

Version	Date	Created By	Reviewed By	Description
2.6	10/02/2012	Charlie Tsai		Updated to company contact information
2.7	6/10/2015	Charlie Zhang	Karan Arora	Update some information to match JBoss 6
2.8	8/26/2015	Charlie Zhang	Karan Arora	Add some configuration and jar file information

Tables of Contents

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide instructions to Node Administrators and Data Flow Administrators on the configuration and administration of the EN-Node application. A companion document, called the ***EN-Node Data Flow Developer's Guide***, is also available and provides instructions on data flow development. Data flow developers looking for instructions on how to build data flows and plug them into EN-Node should use the ***Data Flow Developer's Guide***.

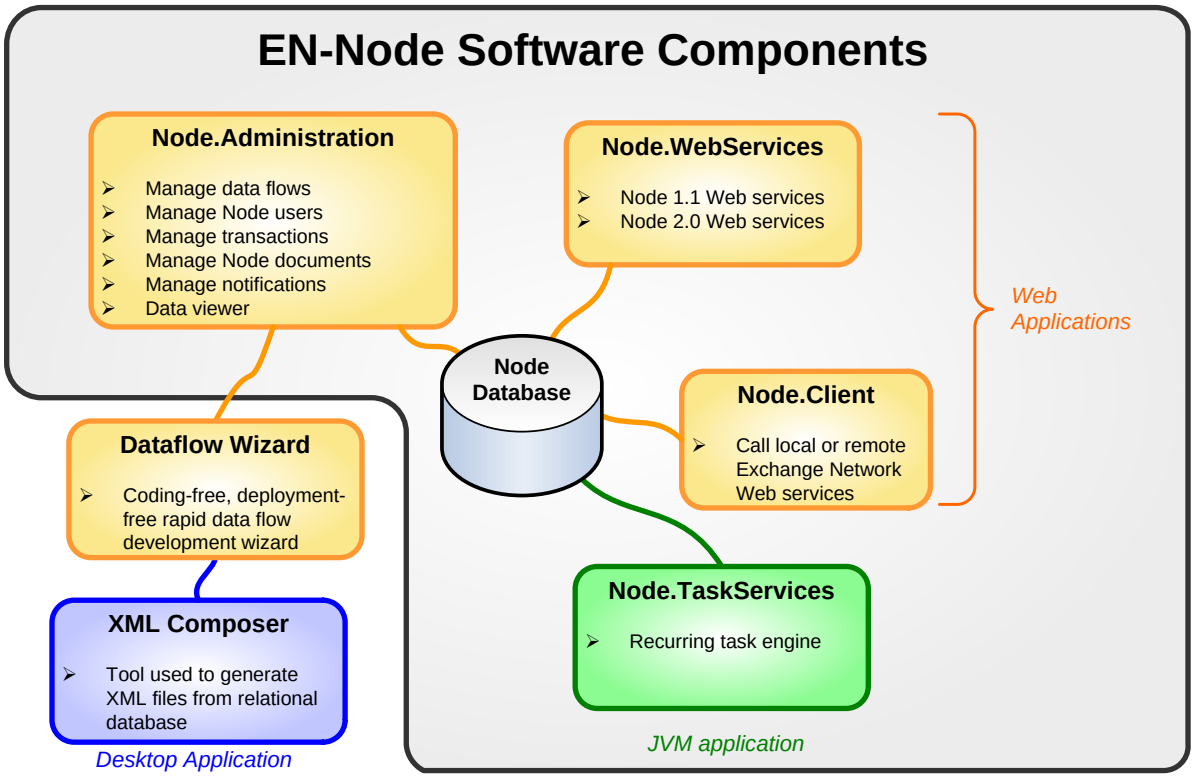
1.2 Terms and Definitions

In this document the following terms and acronyms are used:

General Terms	
Term	**Definition**
Web Method	The processing that a Web Service performs when invoked. In this document, it refers to one of the ten Web Methods that all Exchange Network Nodes implement as defined by the Node Specifications and Protocol Versions 1.1 and 2.0.
Handler	A set of code that executes the logic for a particular Web Method. There is one handler for each Web Method.
Operation	A specific implementation of a Web Method, usually defined by the set of parameters passed in to a particular Web Method that differentiates the type of request. *(For example, the Submit Web Method has the signature (<securityToken>, <transactionID>, <dataflow>, <flowOperation>, <recipient>, <notificationURI>, <documents>). Each allowable setting for the dataflow parameter (e.g. - NEI, NEICritHap, FRSMonthly, etc) defines a unique* ***operation*** *for the Submit* ***Web Method****.)* As such, each Handler can have many Operations.
Domain	A logical grouping of related Operations on which to base data flow administration through the EN-Node Administration console (e.g.: NEI, FRS, RCRA, etc). Every operation is tied to one and only one Domain.
Task	A service that has an associated schedule which determine the interval at which it is executed. Tasks are executable and can be executed either on a recurring schedule using the Node Admin Console, or on demand by a Node or Domain Administrator.
NAAS	Network Authentication and Authorization Services. A set of centralized security services hosted by EPA for performing authentication and service authorization of Node users.

1.3 Overview of EN-Node Components

EN-Node provides a one-stop solution for creating, deploying, managing, scheduling, and interacting with Web service-based Exchange Network data flows. As a Web services engine, it allows other trading partners to interact with your environmental data in a secure and consistent manner. The diagram below shows the EN-Node software suite components:



Each of these components is described briefly below:

- **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 2.1 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 2.0/2.1 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.Task:** 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.

1.4 Available Web Services

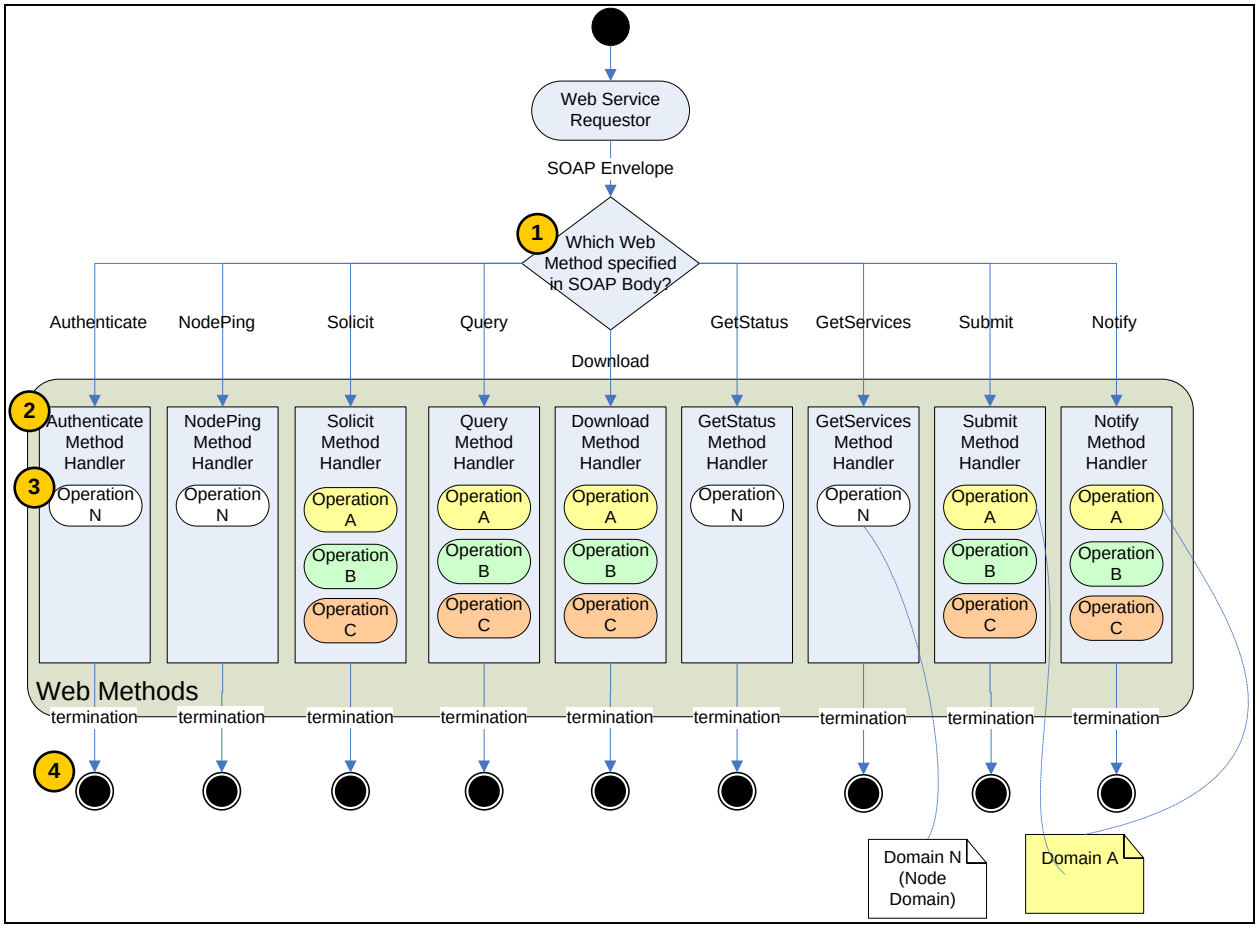
The Node comes preconfigured with nine standard Web services, as defined in the Node 1.1 and 2.0 / 2.1 specifications. These Web services include[1]:

- **Authenticate:** Authenticates a user using a supplied credential.
- **Download:** Provides a means for retrieving documents from a Node. These can either be documents associated with a previous transaction or general documents available for download.
- **GetServices:** The GetServices method is a function in the Admin interface. It allows requesters to query services provided by a Network Node.
- **GetStatus:** Provides transaction tracking of web service requests. Once submitted, a transaction enters into different processing stages. The GetStatus method offers a web service requester a way of querying the current state of the transaction.
- **NodePing:** A utility method for determining whether a Node is accessible.
- **Notify:** The Notify method has three (3) intended uses: document notification, event notification, and status notification
- **Query:** This method is intended to run a series of predefined information requests that return data in an XML instance document that conforms to a predefined standard schema.
- **Solicit:** The Solicit method performs a requested operation asynchronously in the background. It is designed especially for queries that may take a long time.
- **Submit:** Provides a generic way of sending one or more payloads to a service provider.

[1] For more information on the Node Web Services, please refer to the Node Version 1.1 or 2.1 Specifications at: http://www.exchangenetwork.net.

1.5 Overview of How EN-Node Processes Web Service Requests

When a web service request is received by the Node, it is processed as indicated in the diagram below:



1. When a request for any of the Web services listed in Section 1.4 is received at the Node address, the Node will read the Web Method identified in the SOAP body.

2. Based on the Web Method that is specified, the request will be routed to the appropriate Handler. The "handler" determines the logic/approach by which a Web Service request will be handled.

3. Within each Handler, multiple Operations can be defined that execute a specific sequence of logic. These Operations are defined by and organized by Domain Administrators.

4. After the operation is completed, a response is returned to the Web service requester.

Although each of the nine Web Services has its own Handler, they are all derived from one generic base Handler. The Generic Handler contains the generic logic that EN-Node executes regardless of the Web Service that is invoked. The generic handler has the following behavior:

- Generates a transaction ID each time a new Web Service request is made on the EN-Node

- Initiates logging of the Web Service request to track the status of the request
- Attempts to authorize the request to ensure that the person making the request has the appropriate security rights. The user could either be a locally managed user or NAAS-managed user. EN-Node distinguishes the two types of users by issuing different prefixes for their security token. During authorization, if the token is expired, then the authorization request will fail.
- Once authorization is completed, the EN-Node searches for and executes logic in a particular order. The order in which a dataflow is executed depends on how that dataflow was created:
    - **Dataflows created using the Dataflow Wizard:**
        - The dataflow is executed in the order as defined in the Data Flow Wizard. See Data Flow Developer's Guide for more information.
    - **Dataflows created using traditional Plug-In Approach:**
        - All Plug-Ins registered as "Pre-Processes" for the Operation are executed in sequential order
        - EN-Node then executes the registered Process for the Operation and returns the invocation response value to the requestor as per Node specifications.
        - Once the Web Service method has been processed, EN-Node then executes in order, any Post-Processes registered for the Operation.

2 Node Administration Using the Administration Console Dashboard

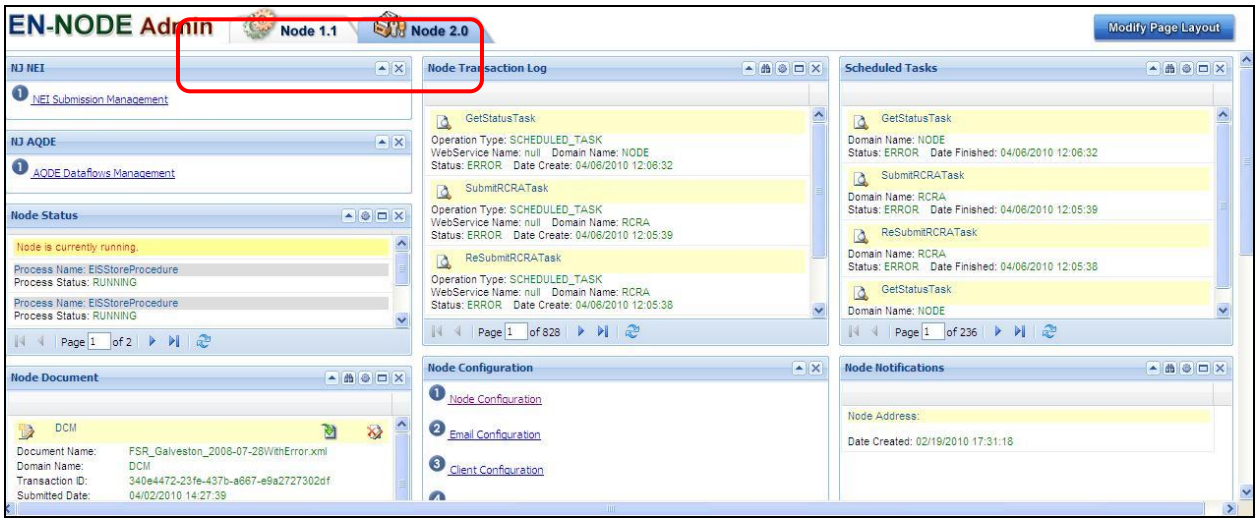
The Node Administration Console is the main interface for Node Administrators to configure the Node and Domain Administrators (or Data Flow Developers) to configure data flows. It can be run by loading the following page:

- .NET: http://<<InstallLocation>>/Node.Administration
- Java: http://<<InstallLocation>>/Node.Administration/Page/Entry/Login.do

2.1 Admin Console Dashboard Overview

When the Node Administration Console is initialized, the user is brought to the login page. You must have an Administration Console Account to access the Admin Console. Simply enter your username and password and click login to enter the utility. The login page is shown below in Figure 2-1.

After the user has correctly entered the username and password, the user is brought to the Administration Console Dashboard. The Administration Console Dashboard provides a one-stop location to access all Node and dataflow administration tasks, as shown in Figure 2-2 below:



The Admin Console Dashboard provides 2 tabs to allow you to manage Node 1.1 and Node 2.0 web services using the same application.

Before proceeding further, we will take a closer look at the layout of the Node Administration Console Dashboard. A brief description of each Web Part is given below:

- **Node Configuration:** allows the Node Administrator to control general node settings. This page is accessible only to Node Administrators. **See Section 2.3 for more details.**
- **Node Domains:** allows Node Administrators to create and manage Domains. This menu option also provides Node and Domain Administrators with the ability to create and manage operations (i.e. dataflows) under the domains. **See Section 2.4 for more details.**
- **Node Monitoring (Node Transaction Log):** allows the user to view a log of all activity captured by the Node. Node Administrators can view logging for all data flows, while Domain Administrators can only see logging for the domains they have access to. This web part allows viewing of logging for web service requests as well as Scheduled Tasks. **See Section 2.7 for more details.**
- **Node Documents:** provides the capability to search, upload, or download any documents that are stored in the Node. This can either be documents submitted to the Node or prepared for outgoing submission. **See Section 2.8 for more details.**
- **Favorite Links:** allows Node Administrator to add any URL for easy access. The following links are provided by default:
    - **Node Users:** allows Administrators to create and manage users (both Node Users and Administration Console Users). **See Section 2.6 for more details.**
    - **Node Registration:** allows Administrator to add/update data service's meta data to the EN-Node. **See Section 2.9 for more details.**
    - **Node Client:** A link to access Node.Client Application. **See EN-Node User's Guide for details.**
    - **Operation Manager:** allows Node Administrators to generate, review, upload and/or submit XML files to a specified endpoint from EN-Node. **See Section 2.5 for more details.**
- **Node Status:** displays the status of the Node and lists any current running background processes. **See Section 2.12 for more details.**
- **Scheduled Tasks:** allows the user to view the logs for scheduled tasks. **See Section 2.10 for more details.**
- **Node Notifications:** allows the user to view a listing of any notifications received from other Nodes. **See Section 2.11 for more details.**

2.2 Modifying the Admin Console Dashboard Layout

The layout and display of the Web Parts can be modified by clicking on the 

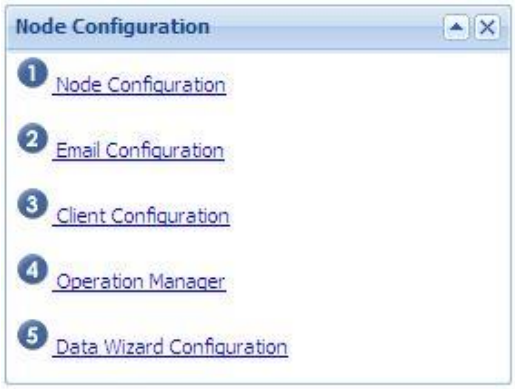
button in the top right corner, which puts the console into Edit Mode. Once this is clicked, a left panel will appear and display any Web Parts that aren't being displayed and lets you add them to the Console.

While in Edit Mode, you can add a Web Part by dragging-and-dropping it from the Config Center Console to its desired location on the screen.

Clicking the "X" in the left panel will exit the Console from Edit Mode and save your changes. Any changes you make will be saved and used for future times you log into the Admin Console. Each user can configure the dashboard appearance separately.

2.3 General Node Configuration Settings

The **Node Configuration** Web Part is accessible only to Node Administrators and allows them to control general node settings. Node settings are broken down into the links shown below:



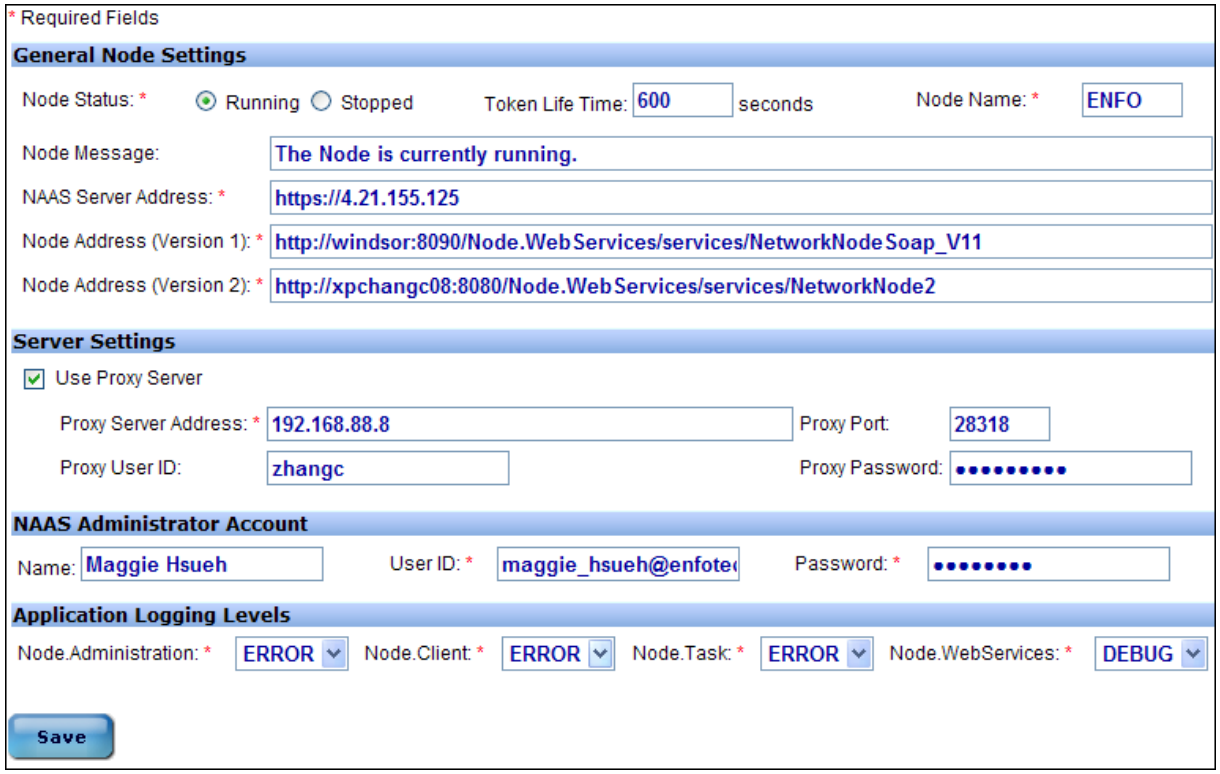
Each of these settings is described below.

Node Configuration:

As shown in the figure below, this page allows the Admin to change node configuration settings.

- General Node Settings:
  - Node Status: Allows the Administrator to start and stop the node
  - Token Life Time: The amount of time, in seconds, before a security token expires
  - Node Name: Unique identifier of the Node as defined by EPA-CDX. This is used when making web service calls to NAAS. If unsure, contact CDX to determine your Node's unique Node Name.
  - Node Message: Specify a message indicating the status of the Node that is displayed when users access the Node Status page.
  - NAAS Server Address: The URL or IP address of EPA's NAAS Authentication Web Service
  - Node Address (Version 1): Configure the URL of the Node 1.1 compliant Web Services endpoint for the EN-Node
  - Node Address (Version 2): Configure the URL of the Node 2.0 compliant Web Services endpoint for the EN-Node
- Server Settings:
  - Proxy Server: Set the outbound proxy server settings if your node server needs to access a proxy to access the internet.
- NAAS Administrator Account:
  - Node Administrator User ID / Password: The Name, User ID (email address) and password for the NAAS Administrator for your Node. If you are unsure about this account, contact EPA-CDX to obtain an account.
- Application Logging Levels: Allows you to define the level of exception logging that will be written to the Node logging text files. This is separate from typical Node transaction logging and it intended to capture exceptions/debugging. Different log levels can be defined for the 4 eNode application components:
  - Node Administration Application
  - Node Client Application
  - Node Task Service
  - Node Web Services

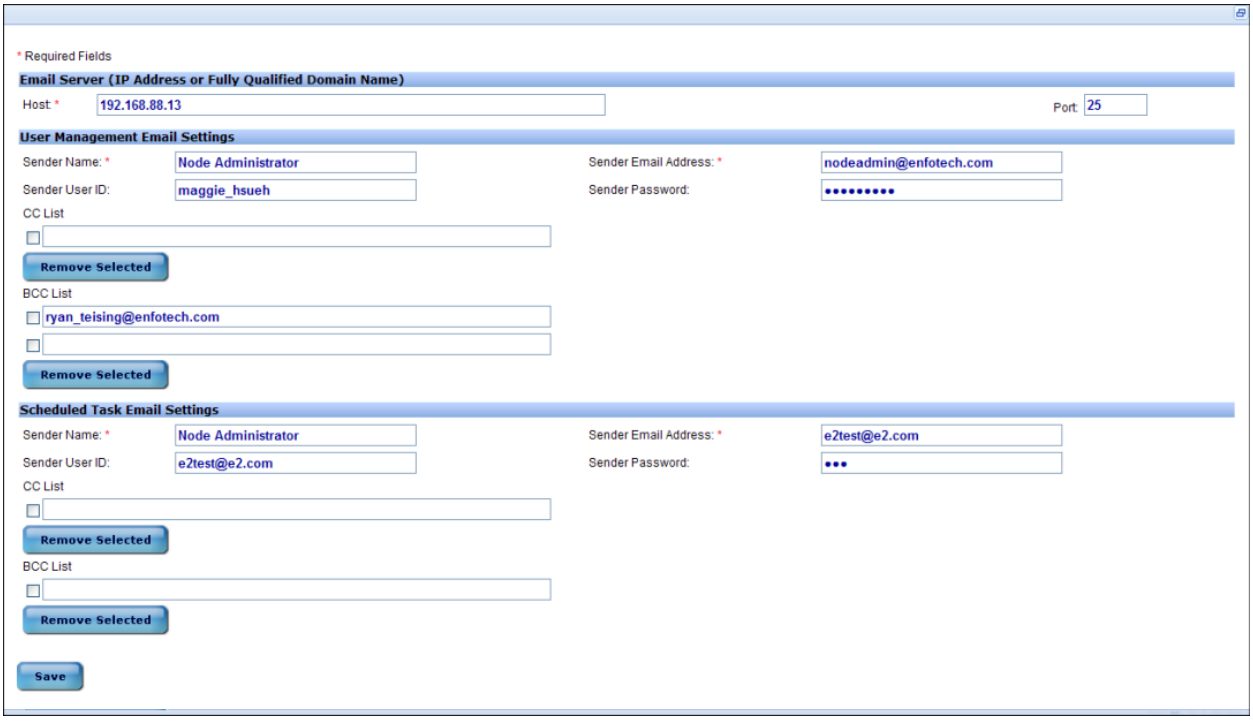
- Download/Upload Buttons: As an alternative to editing the configuration settings on the screen, you can download the configuration settings to an XML file, make edits, and then re-upload the modified configuration file. Caution: be careful if using this approach.
  Note: The most common situation when using the Download/Upload buttons are when new configuration settings have been added to the Node.



2 Email Configuration:

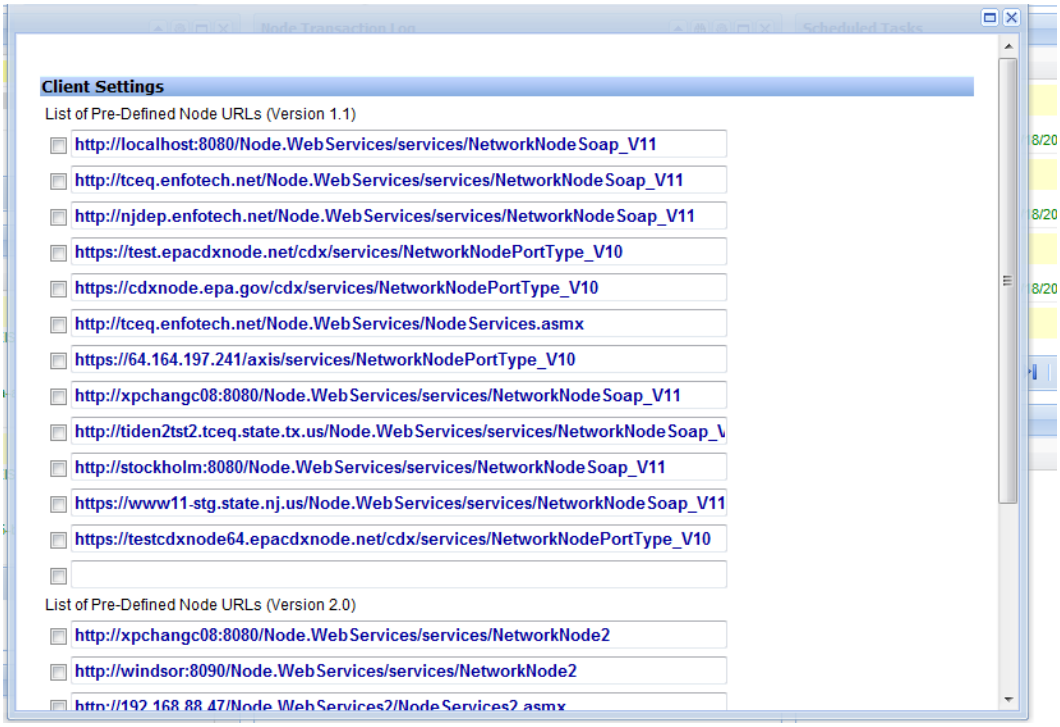
As shown in the figure below, this page allows the Admin to specify the general email settings. Correct configuration is essential to ensure that the eNode can send out emails, which is done during user account creation and, depending on how a data flow is configured, may also be done during data flow operation. This includes:

- Email Server Configuration: The admin has the ability to configure the following server email server information:
  - Email Server Host
  - Email Server Port
  - Email Server User ID
  - Email Server User Password
- User Management Email Settings: Used to control the behavior of emails sent out from the eNode any time a new user is created or the user's password is changed. *(If you would like to cc or bcc multiple people, enter in the first email address, then click **Save**. A 2nd box will then appear on the screen to add the 2nd email address.)*
- Scheduled Task Email Settings: Used to control the behavior of emails sent out from the eNode any time an eNode task is completed. *(If you would like to cc or bcc multiple people, enter in the first email address, then click **Save**. A 2nd box will then appear on the screen to add the 2nd email address.)*



3 Client Configuration:

Allows the Administrator to specify the default Node WebService endpoint URLs that will appear when people use the Node Client application. These URLs will then be displayed as default URLs that can be selected while using the Node Client application.



4 Operation Manager Configuration:

The Operation Manager Configuration page allows Data Flow Developers to configure the behavior of the Operation Manager page (which is accessible under “Favorite Links”).

What is the Operation Manager Module used for?

The Operation Manager Page is very useful anytime a Data Flow Developer wants to quickly create a customized webpage to allow internal agency staff to perform one or more of the following actions:

- Manually initiate the generation of XML files (i.e. “Generate”)
- Manually upload files such as XML files to the Node (i.e. “Upload”)
- Manually submit XML files to an external partner such as EPA (i.e. “Submit”)

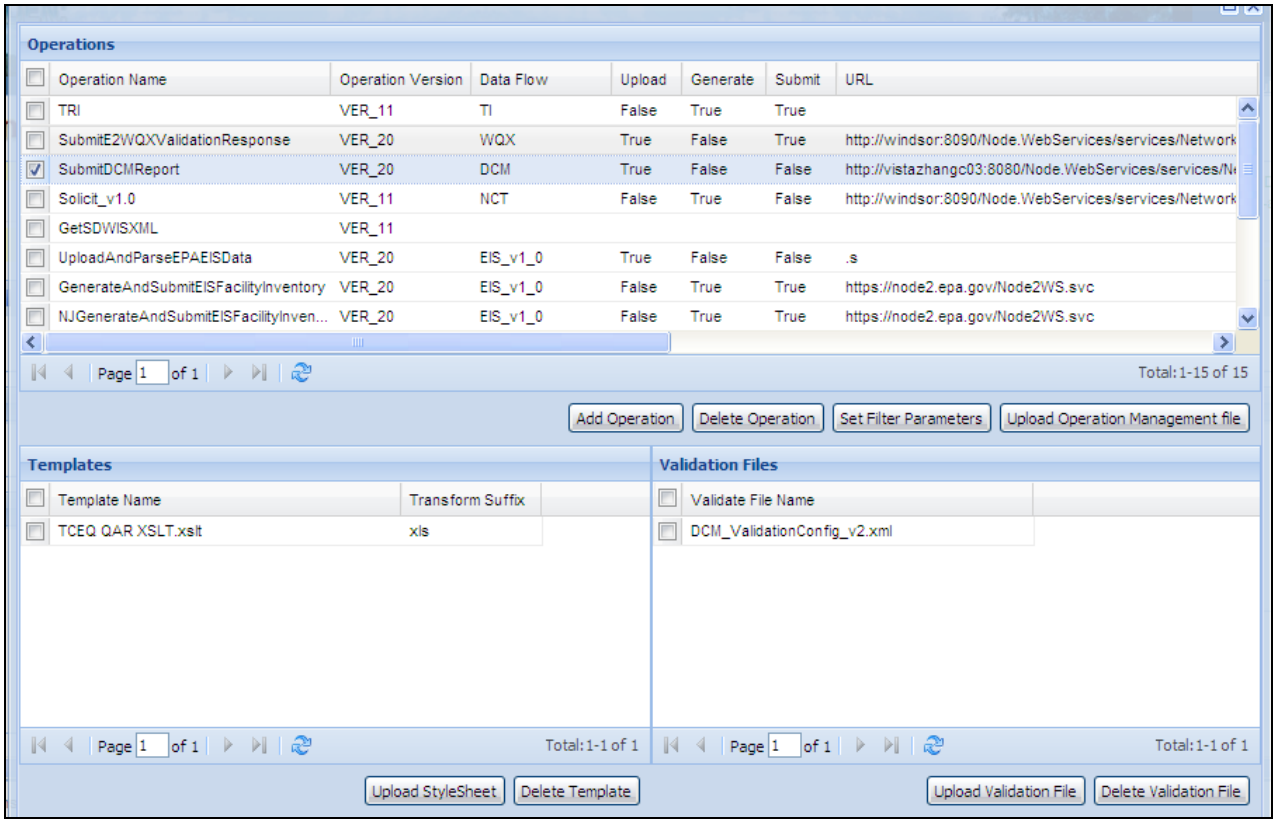
Simply creating a Node dataflow does not automatically accomplish the bulleted items above (especially if you want to link multiple bullets above together, for example when you want to Generate and XML file, then review, then Submit to EPA), which is why the Operation Manager Module is useful as an additional layer on top of dataflows. Some common scenarios are:

- **Scenario 1:** Each month, user wants to access a screen where they can click a button to initiate the generation of a AQS XML file, then have that XML file appear so they can review it for accuracy, then click a second button to submit the XML file to EPA. They don’t want to have to type in the CDX URL or CDX username/password every time they wish to make the submission.
- **Scenario 2:** User periodically get emailed XML files from an external partner and wants a screen where he/she can upload those XML files to the Node (perhaps for centralized data management), and then click another button to submit the XML file to EPA.
- **Scenario 3:** Each month, user wants to generate a FRS XML file, then have the node submit to EPA. After the submission, they want the node to automatically perform GetStatus and Download calls on the EPA node to check the status of the submission.

When is the Operation Manager Module not a Good Option?

In cases where you want complete automation (and therefore don’t want to be bothered with forcing users to access webpages to perform actions), the Operation Manager is not what you want. But there are many cases where users do want a screen that they can access to manually initiate a dataflow and review all past data flow executions.

The Operation Manager Configuration page is shown here:



This screen lets the Data Flow Developer or Node Admin configure the following:

- Specify which operations are available at the Operation Manager page. If the operation is shown in the top grid on this page, then that indicates that this operation is available at the Operation Manager screen.
- Control whether this Operation supports XML File Generation, XML File Upload, and/or XML File Submission to EPA. This is controlled by setting the “Upload”, “Generate”, or “Submit” columns to ‘true’.
- Upload one or more stylesheets to transform XML files that are generated by the Node or uploaded to the Node. Once uploaded they are available to data flow users.
- Upload a validation file used to validate XML files generated by the Node or uploaded to the Node

Specify Operations Available on the Operation Manager screen:

Administrators can define which operations are available at the Operation Manager screen by clicking on the Add Operation or Delete Operation buttons. After clicking “Add Operation”, a new row will appear at the bottom of the top grid on the screen. The Admin can then click the drop-down for the Operation Name column to specify which existing Operation will be made available at the Operation Manager screen.

The Administrator also has the ability to upload a list of operations into the system by clicking on the Upload Operation Management file button. This is a quick shortcut to load in multiple records at the same time.

Setting Operation Manager Attributes:

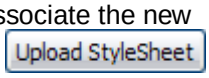
After and operation has been added, you can set the various configuration settings in the top grid:

- **Operation Version:** select “VER_11” or “VER_20” from the drop-down. (“VER_20” works for both 2.0 and 2.1 dataflows.) This field is important in cases where you are configuring the Operation to perform a Submit to another Node, to ensure the node knows to make the submit as Node 1.1 or Node 2.0 dataflow.

- **Data Flow:** specify the name of the dataflow that would be used when making a submit to a remote node.
- **Upload/Generate/Submit:** Set to "true" if you want the Operation to support this method. You can set more than one column to true. Based on your selection here, different buttons will appear on the Operation Manager page.
- **URL:** When configuring a Submit, this identifies the Node URL to where you wish to submit the file.
- **User Name/Password:** When configuring a Submit, this identifies the NAAS name and password used to authenticate prior to making the submission
- **Domain Name:** When configuring a Submit, this identifies the value that will be passed for the Domain parameter on the Submit call.
- **GetStatus:** When configuring a Submit, set to "true" if you want the Node to automatically perform a GetStatus and Download call on the remote node after the submission has been made. When this is set to true, the node will have a task continually running in the background that monitors whether any submissions have been made for this operation to a remote node. If yes, the node will periodically call a GetStatus on the remote node until the remote Node returns either "Failed" or "Completed". If the remote node returns "Failed", the node will also automatically download the error report from the remote node and make available on the Operation Manager page.
- **GetStatus Complete:** When configuring a Submit that also enables GetStatus calling, you can change the value of the status code returned from the remote node. Typically the remote node will return "Completed" when your submission completes successfully. But if the remote node returns another value (e.g. "Passed"), you can set this here.
- **Get Status Error:** When configuring a Submit that also enables GetStatus calling, you can change the value of the status code returned from the remote node. Typically the remote node will return "Failed" when your submission completes with error. But if the remote node returns another value (e.g. "Failure"), you can set this here.

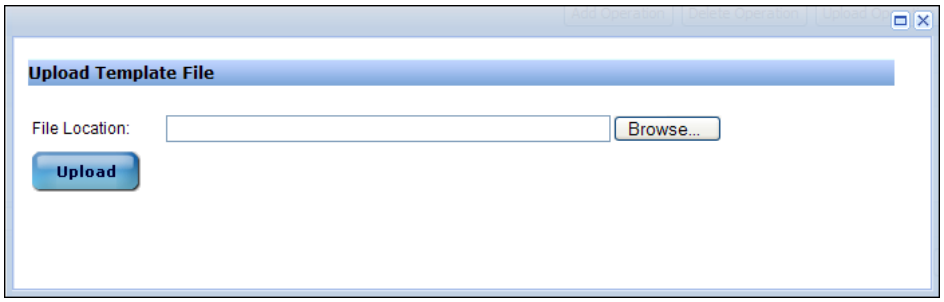
Upload/Delete Stylesheets:

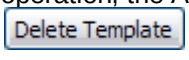
Stylesheets are used to transform XML files into a more human readable format. This is very helpful in the case where the Node is generating an XML Administrators have the ability to upload one or more XML stylesheets, which will then be available at the Operation Manager page when the user clicks the "View" link.

To upload a new stylesheet, the Administrator first selects the operation he or she wishes to associate the new templates with by checking the box next to the desired operation. The Administrator clicks the 

button, to display the Upload Template File screen as shown in Figure 2-9 below. The Administrator selects the file from his or her local computer and clicks the 

button.

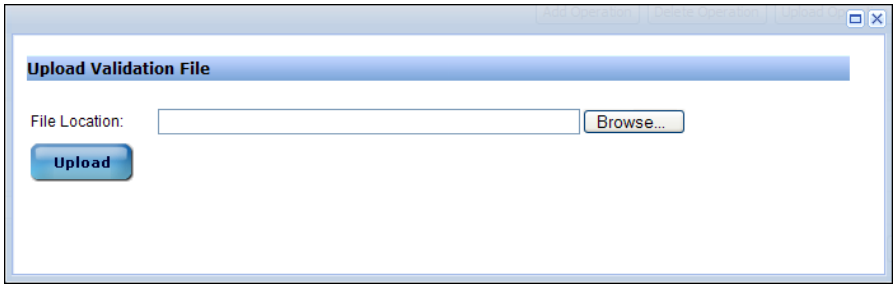


To delete a template from an operation, the Administrator selects the operation he or she wishes to delete templates from and clicks the 

button.

Upload/Delete Validation Files:

Validation files are used to perform a set of validation on XML files. Administrators have the ability to upload and delete validation files in the system. The validation files use the enfoTech Validation Rule Engine (VRE) that run against XML files generated for this operation. To upload a new validation file, the Administrator first selects the operation he or she wishes to associate the new templates with by checking the box next to the desired operation. The Administrator clicks the Upload Validation File button, to display the Upload Validation File screen as shown in Figure 2-10 below. The Administrator selects the file from a local drive and clicks the Upload button.



To delete a template from an operation, the Administrator selects the operation he or she wishes to delete templates from and clicks the Delete Validation File button.

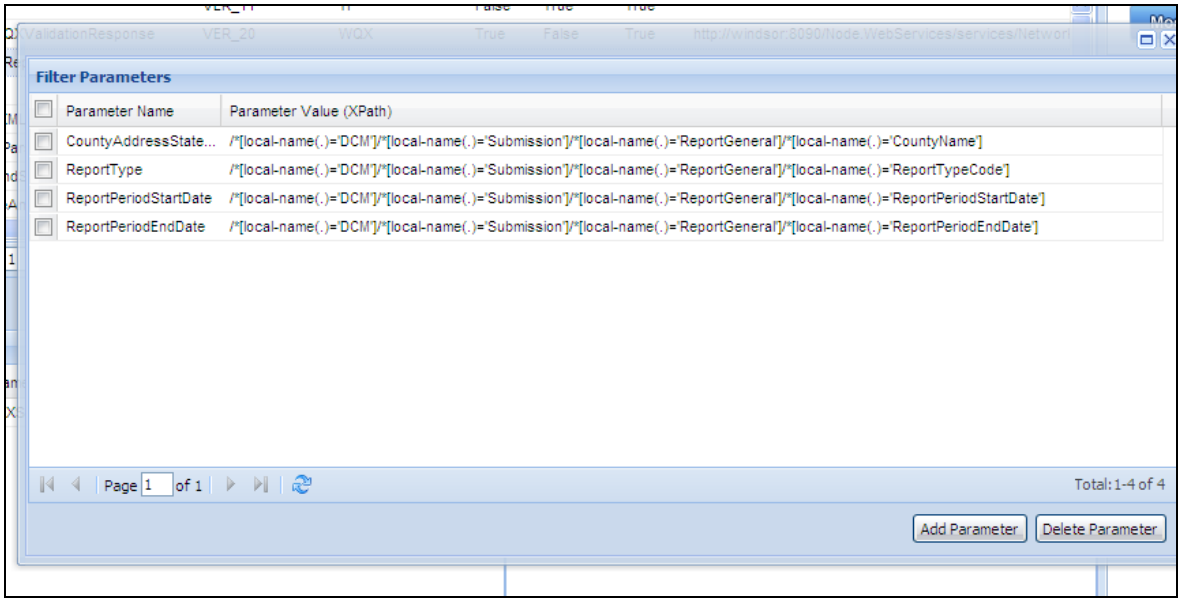
Set Filter Parameters:

At the Operation Manager page, every time a data flow runs (either by clicking a button to generate an XML file or clicking a button to upload an XML file), a new row is added to the Operation Manager data grid. This is very useful, as the Node therefore stores a history of all of your activity for the dataflow. However, over time this data grid can grow quite large. People have a need to query a subset of the past records that have been uploaded or generated.

Fortunately, the Node provides a very powerful method to filter past operation records based on the data within the generated or uploaded XML files. So for example if your Operation generates WQX XML files, you could have the operation manager page display only the past records that were generated for “Project ABC” where “Project ABC” is the value of a field within each of the XML files.

To take advantage of this feature, click on the “Set Filter Parameters” button, then click the “Add Parameter” button. Enter in a parameter name (can be anything) and Parameter value. The parameter value must be a valid XPath that points to the location of the data element within the uploaded or generated XML files. Here is an example:

```
/*[local-name(.)='DCM']/*[local-name(.)='Submission']/*[local-name(.)='ReportGeneral']/*[local-name(.)='ReportTypeCode']
```

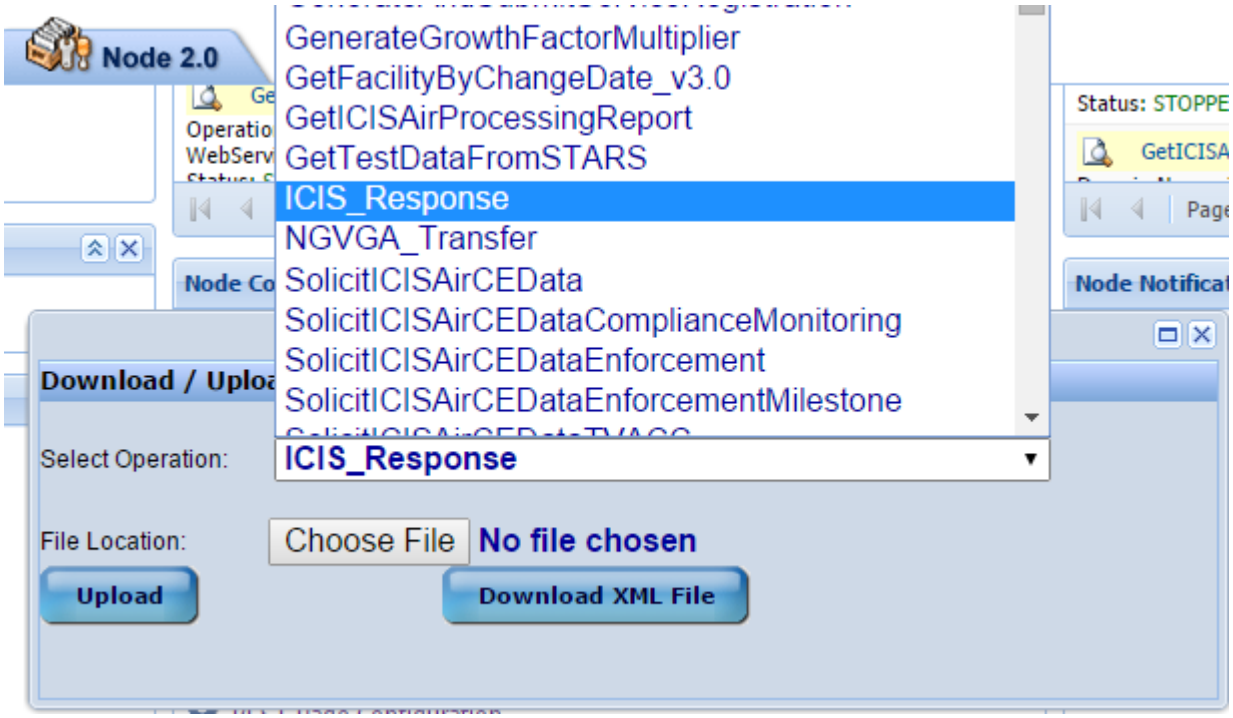


5 Data Wizard Configuration:

If you build dataflows using the Data Flow Wizard, the logic that controls the behavior of the dataflow is stored in an XML configuration file. The Data Wizard Configuration page allows user (especially users that have not purchased the separate Data Flow Wizard Utility) to quickly download and upload the data flow wizard configuration files.

More details on how to craft these XML files is provided in a separate Data Flow Developers Guide.

1. User select one data wizard driven operation from drop down list



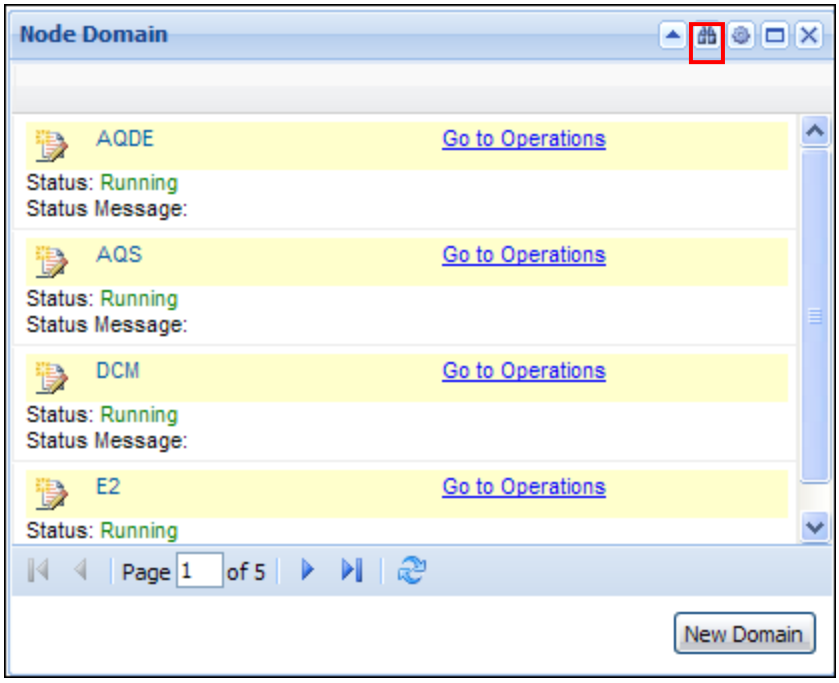
Data Wizard configuration: Select operation to download / upload Configuration window

2. User click “Download XML File” to download xml format operation configuration file.
3. After editing the configuration file, user can click “Upload” button to upload configuration file back.

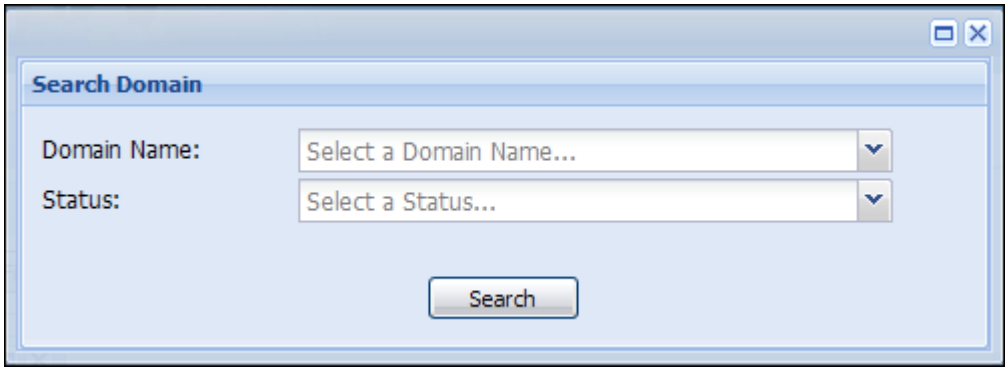
2.4 Domain and Data Flow Administration

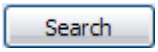
The **Node Domains** Web Part allows Node and Domain Administrators to search, view, and edit details about Domains, create new Domains, and manage operations on the Domains. As stated earlier, a Domain is a collection of node operations (or dataflows). Operations are organized into domains to allow for operation management. *(For example, you may have 3 web service operations called GetDrinkingWaterResults, GetDrinkingWaterInventory, and GetDrinkingWaterViolations. By grouping these 3 operations into a domain, you can setup 1 domain administrator who would be able to manage all 3 operations.)*

In the Node Domain Web Part, Node Administrators will be able to see all Domains, whereas Domain Administrators will only be able to see the domains they are associated with.



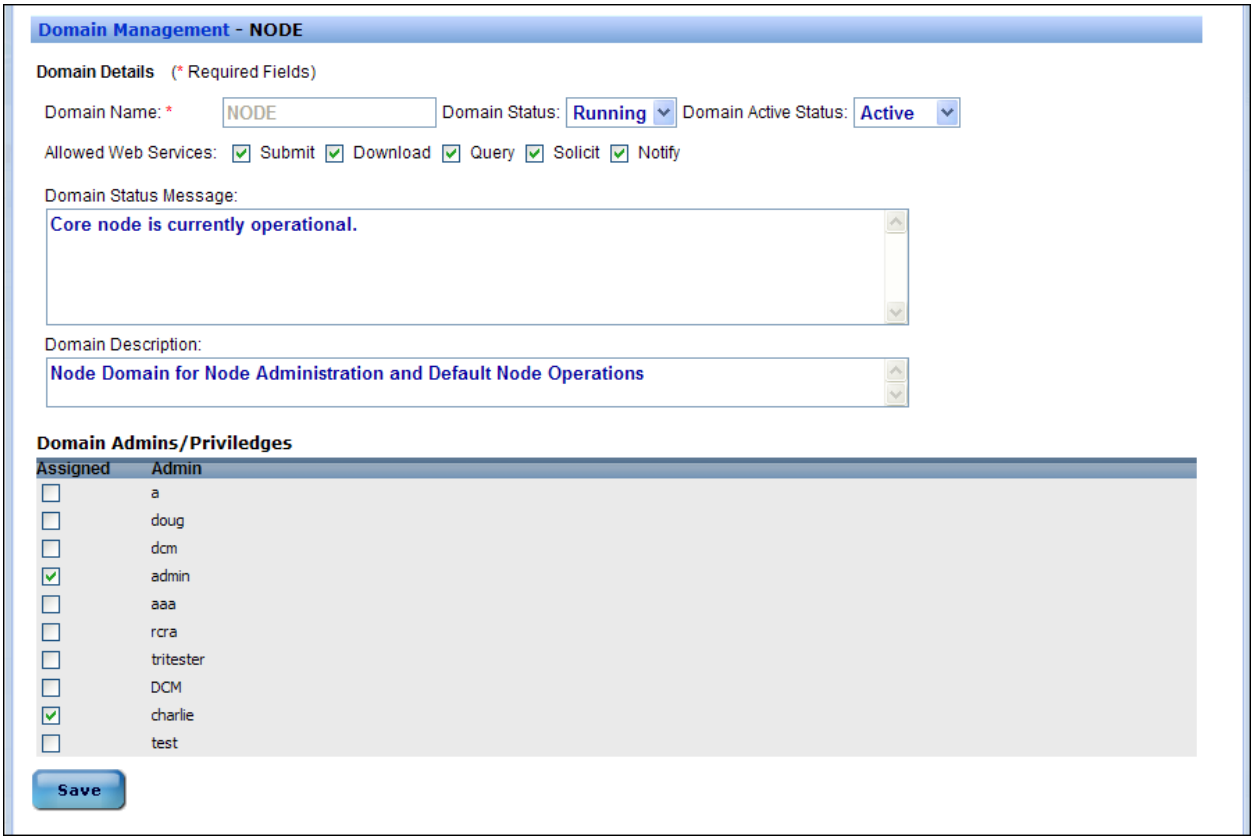
If many domains are listed, you can click the search icon  to display the domain search screen.



Selecting the Domain name and Status from the dropdown menus and clicking on the  button will display a list of Domains that satisfy the search criteria specified.

Note: Node Administrators can leave the Domain name drop-down blank to list all domains registered on the Node.

By clicking on the Domain name link, the Domain details screen will be displayed, as shown here:



The Domain Details page displays details about the domain:

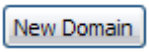
- **Domain name:** The name of the domain assigned by the Node Administrator while creating the domain. This field is not editable once created.

- **Domain Status:** This allows the Domain Administrator to set the Domain to either Running or Stopped Status.
- **Domain Active Status:** This allows the Domain Administrator to set the Domain to either Active or Inactive Status.
- **Allowed Web Services:** This displays the restricted set of Web Services that a Domain Admin is allowed to create when creating operations under this domain. This allows the Node Administrator to restrict which web services a Domain Administrator can create.
- **Domain Status Message:** This is a message that the Domain Admin can set for the Domain. This message will appear when a user views the Domain's status at the Node Status page.
- **Domain Description:** Provides a general description of the Domain.
- **Domain Admins/Privileges:** Allows the Node or Domain Admin to define which Admin Console users can be a Domain Admin.

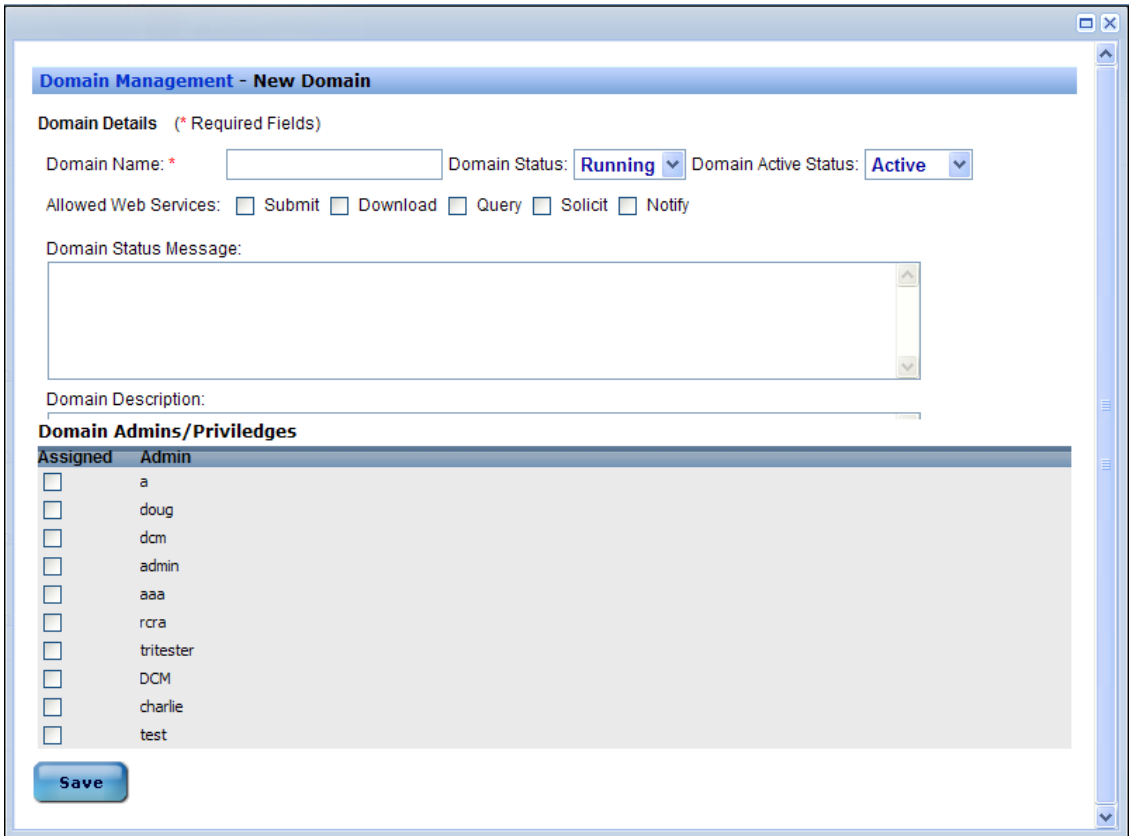
All the modifications made could be saved by clicking on the 

button.

2.4.1 Create New Domain

Administrators can create new Domains by clicking on the 

button which will bring up the following screen:



On this screen, Administrators could fill in the following details for the new Domain to be created:

- Domain name (required field)

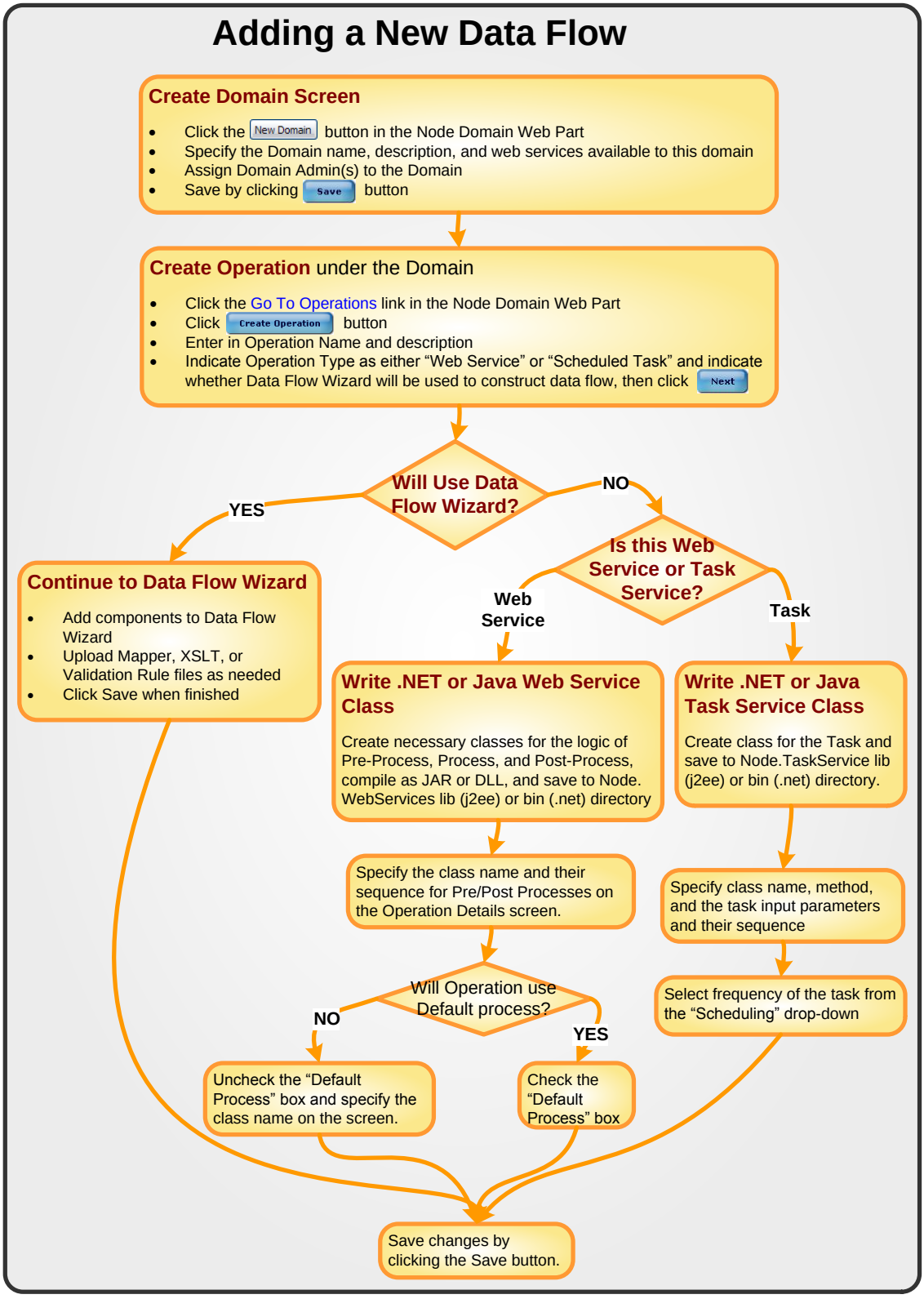
- Domain Status
- Domain Active Status
- Allowed Web Services
- Domain Status Message
- Domain Description
- Add/remove Domain Admins/Privileges associated with the Domain

See the previous section for a description of each of these fields. Upon filling up all the necessary details, the Domain can be created by clicking on the 

button.

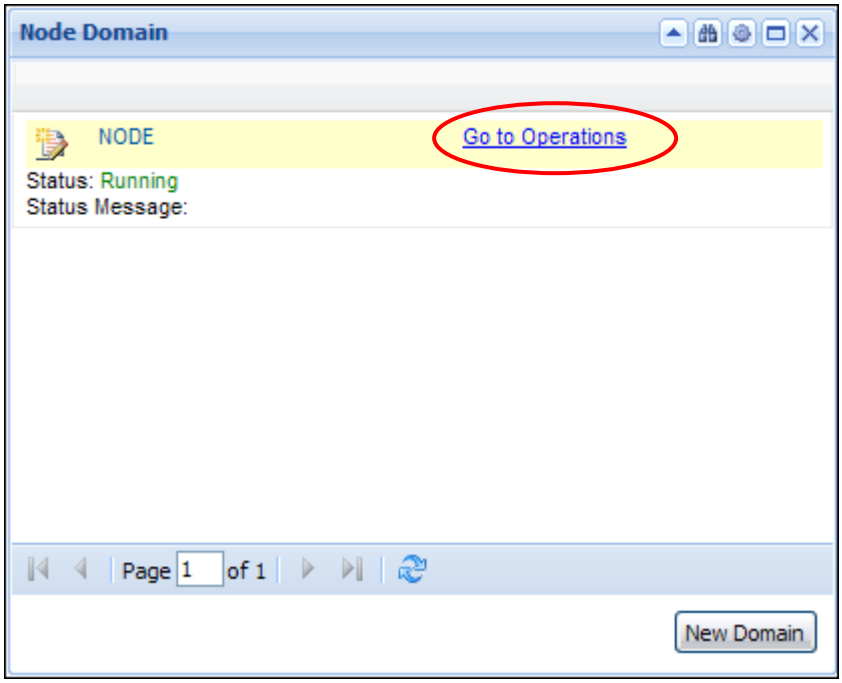
2.4.2 Create New Operations

2.4.2.1 Flowchart for adding new operation.

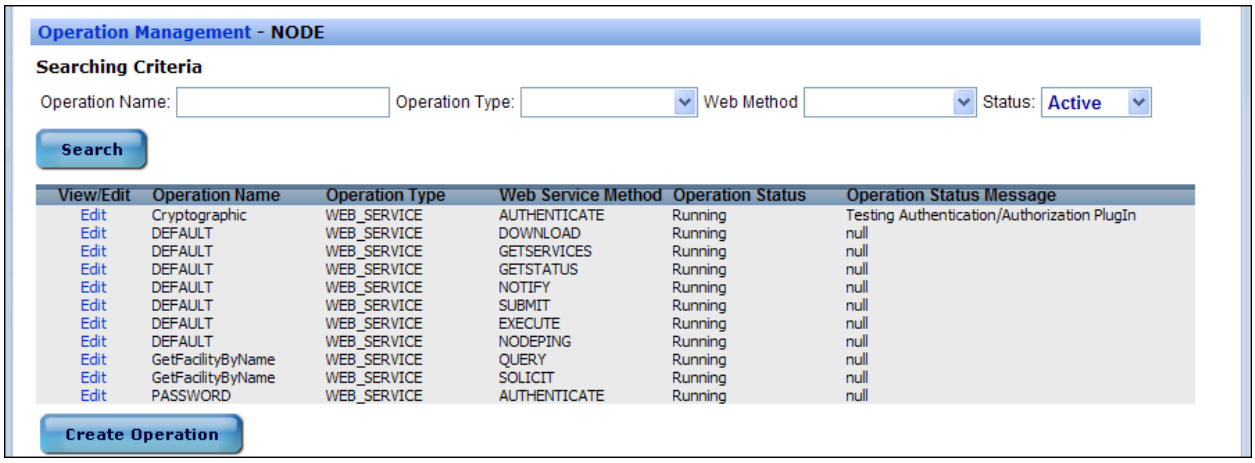


2.4.2.2 Create a new operation

Click on the “Go To Operations” link in the Node Domain Web Part, as shown here:

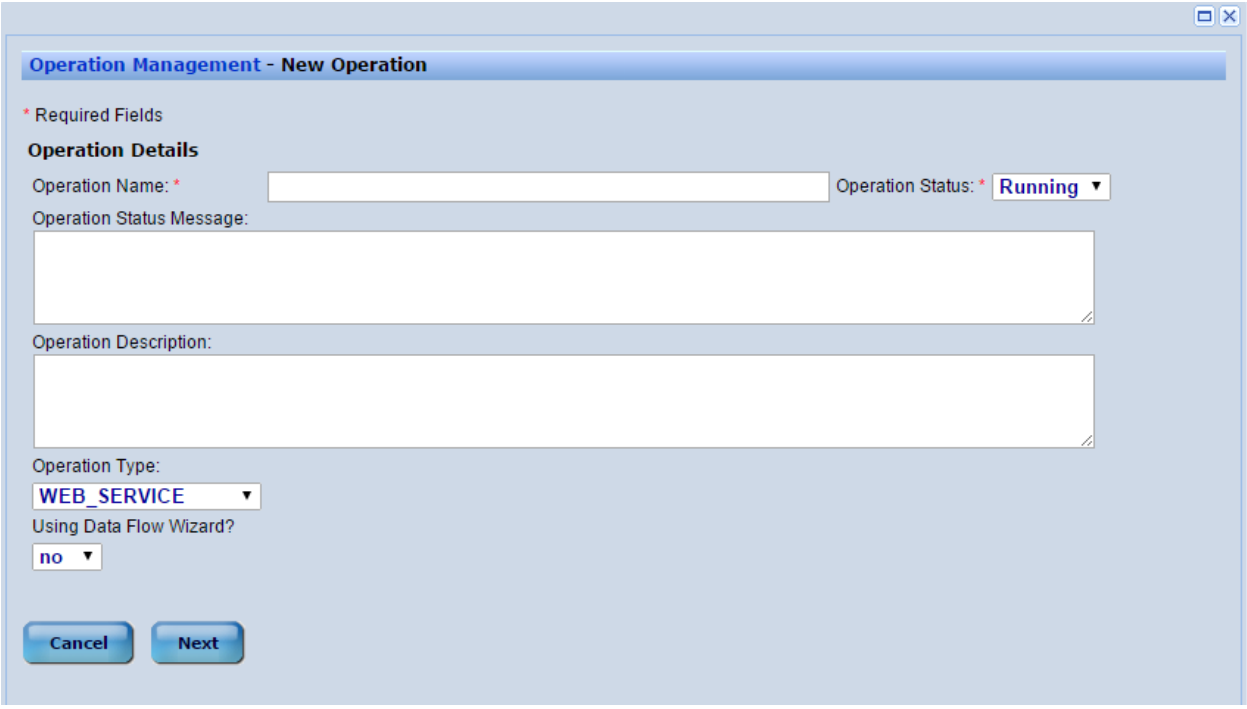


This will take the Administartor to the Operations Management screen as shown below:



View/Edit	Operation Name	Operation Type	Web Service Method	Operation Status	Operation Status Message
Edit	Cryptographic	WEB_SERVICE	AUTHENTICATE	Running	Testing Authentication/Authorization PlugIn
Edit	DEFAULT	WEB_SERVICE	DOWNLOAD	Running	null
Edit	DEFAULT	WEB_SERVICE	GETSERVICES	Running	null
Edit	DEFAULT	WEB_SERVICE	GETSTATUS	Running	null
Edit	DEFAULT	WEB_SERVICE	NOTIFY	Running	null
Edit	DEFAULT	WEB_SERVICE	SUBMIT	Running	null
Edit	DEFAULT	WEB_SERVICE	EXECUTE	Running	null
Edit	DEFAULT	WEB_SERVICE	NODEPING	Running	null
Edit	GetFacilityByName	WEB_SERVICE	QUERY	Running	null
Edit	GetFacilityByName	WEB_SERVICE	SOLICIT	Running	null
Edit	PASSWORD	WEB_SERVICE	AUTHENTICATE	Running	null

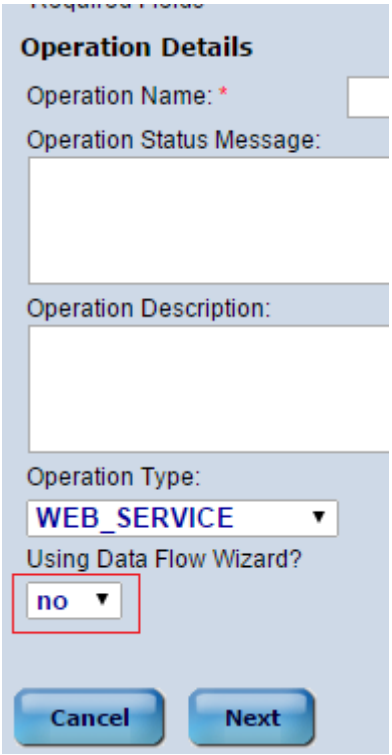
Click “Create Operation” button, user will be leaded to New Operation page



From this step, user need to decide what kind of operation should be created. There are two kind of operations:

- **Non Data Flow Wizard Flows**
- **Data Flow Wizard Flows**

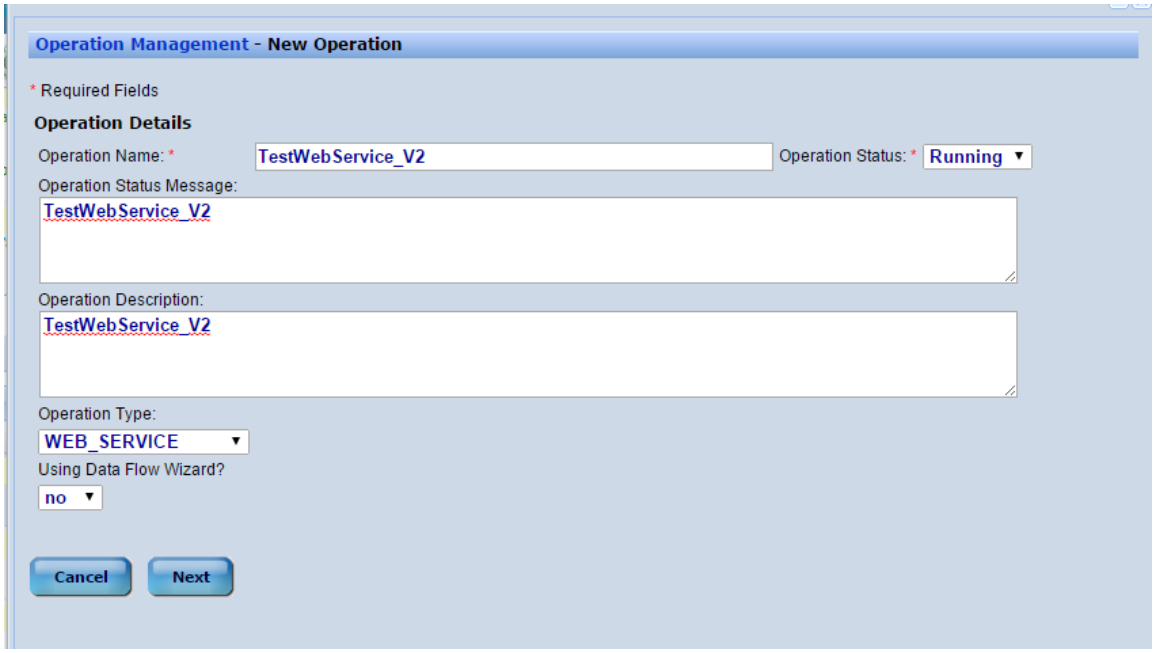
User should use “Using Data Flow Wizard” drop down list to select what kind of operation should be created. “Yes” option is for data flow wizard driven flow, “No” option is for non-data flow wizard driven flow.



2.4.2.3 Non Data Flow Wizard Flows

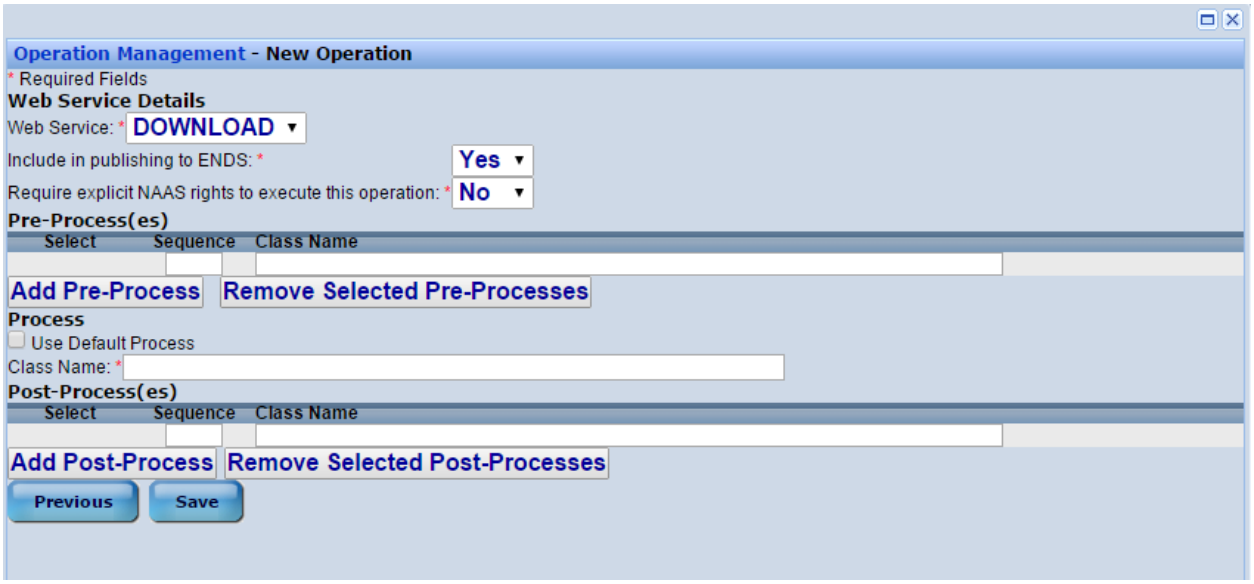
2.4.2.3.1 Operation type is: WEB_SERVICE

Fill the form like below and click “Next” button



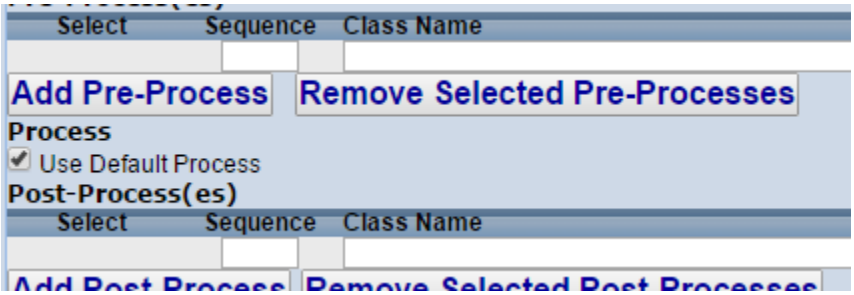
- Operation Name: The name of the operation. This field is not editable once created.
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Using Data Flow Wizard: Indicates whether the operation is data wizard driven flow or not. “Yes” option is for data flow wizard driven flow, “No” option is for non-data flow wizard driven flow.

User can fill more detail information in Operation detail page like below:

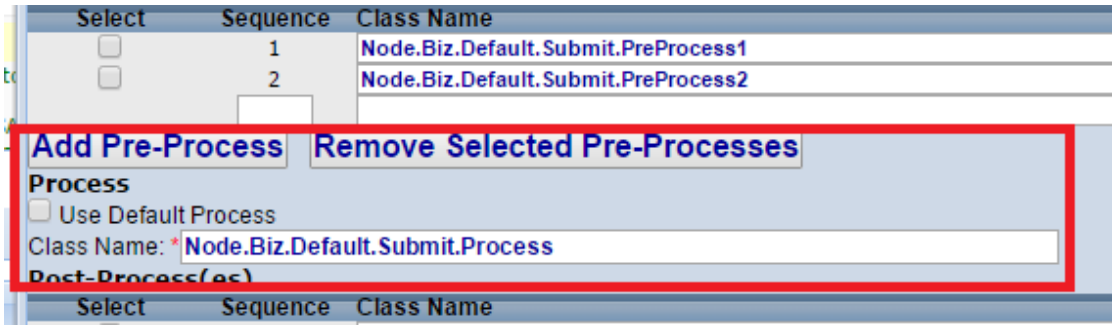


This screen allows user to edit the following information:

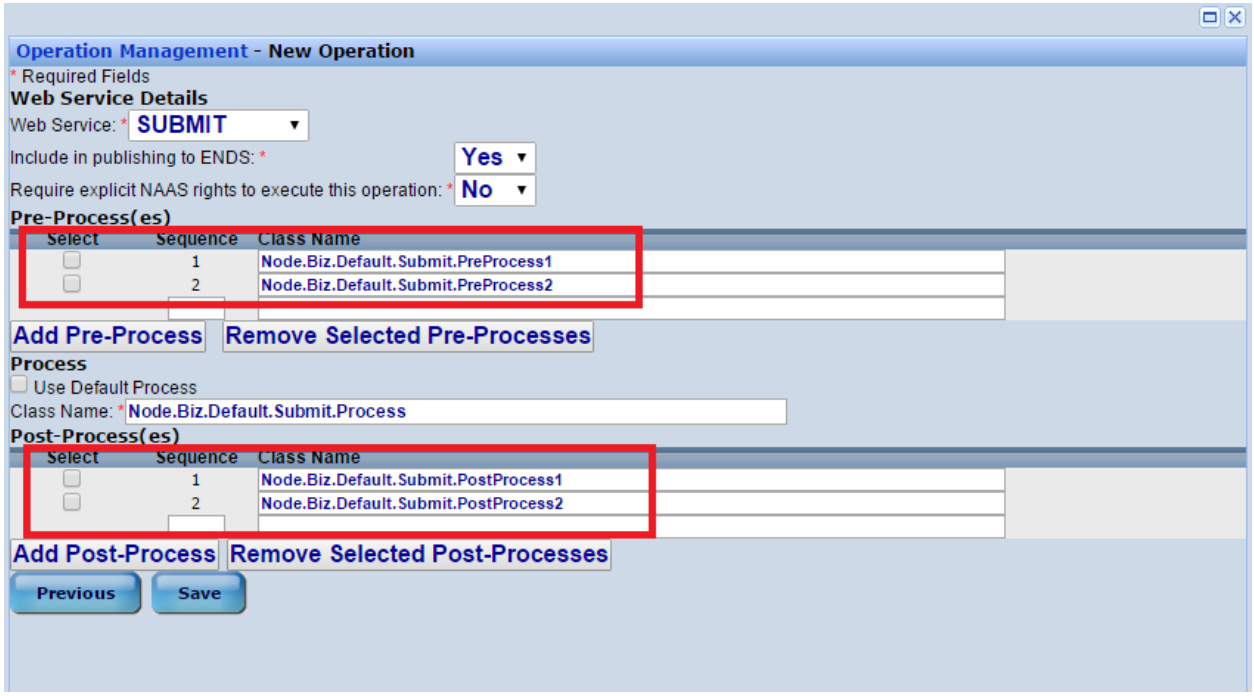
- Web Service: List of web service. User must select a suitable one.
- Include in publishing to ENDS: if you set this to "Yes", then any time someone calls the GetServices method on your node, this operation will be included. If you are integrated with EPA's ENDS, then this will determine if this operation will get registered in ENDS.
- Require explicit NAAS rights to execute this operation: if you set this to "Yes", then the only NAAS users who can execute this operation are those that you define at the User Management page. Alternatively, if you set this to "No" then all NAAS users will be able to execute this operation. Note: this rule only applies to Queries and Solicits.
- For a Web Service Operation, if the "Use Default Process" checkbox is checked, the Operation implements the "DEFAULT" Operation on the "NODE" domain.



If this is unchecked, user can simply enter the Class name that will define the process.

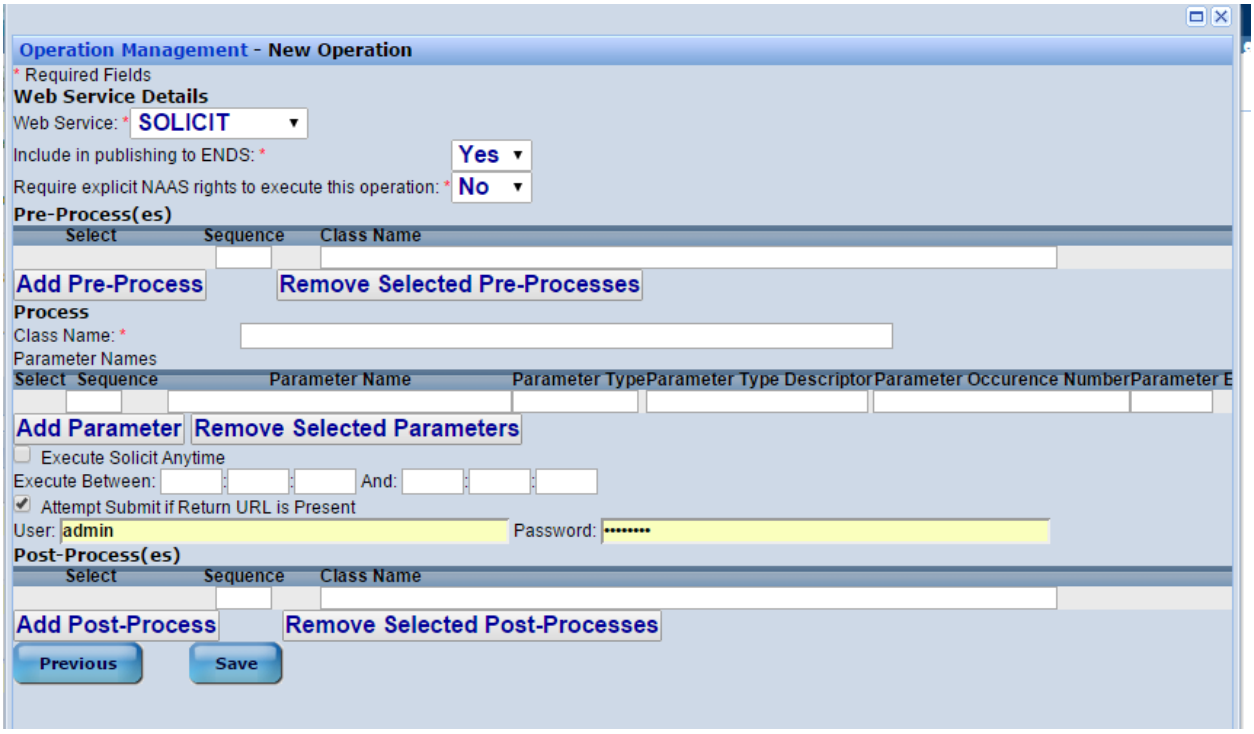


- New pre-processes and post-processes could be added to the Operation by specifying the Sequence and Class Name and selecting either the Add Pre-Process or the Add Post-Process buttons as shown in the figure below. The following screen will be displayed:



Below items will only be shown up for solicit web service.

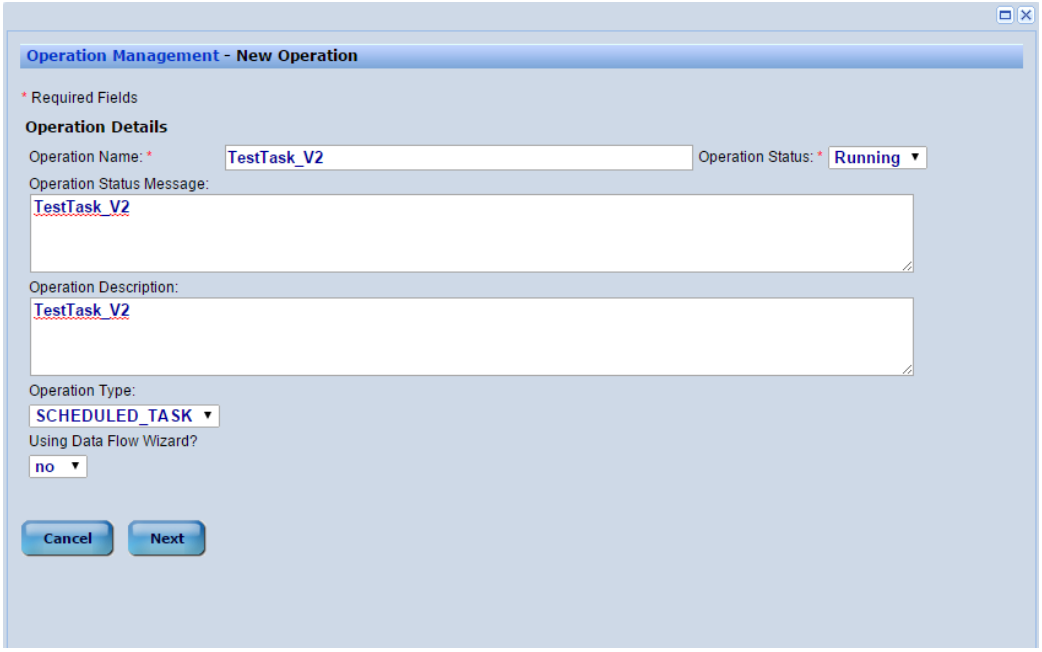
- Parameter Names: Indicates all parameters.
- Execute Solicit Anytime: If check this box, solicit will run in any time.
- Excute between: And: If user fill these two fields, solicit will only run in this time range.
- Attempt Submit if Return URL is Present: If user check this field, he must provide NAAS account name and password to allow Node submit file to “Return URL” (“Return URL” parameter is an optional parameter in solicit web service call).



The changes made to the Operation could be saved by clicking on the  button.

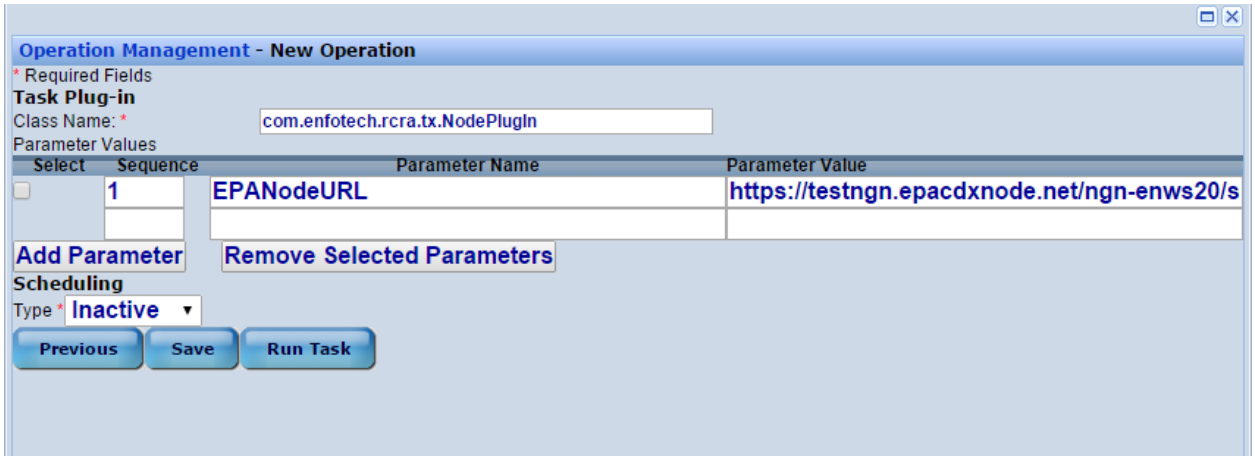
2.4.2.3.2 Operation type is: SCHEDULED_TASK

Fill the form like below and click “Next” button



- Operation Name: The name of the operation. This field is not editable once created.
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Using Data Flow Wizard: Indicates whether the operation is data wizard driven flow or not. “Yes” option is for data flow wizard driven flow, “No” option is for non-data flow wizard driven flow.

User can fill more detail information in Operation detail page like below:



- Class Name: The name of the real java Class.
- Parameter Values: The parameters need to pass to task process
- Scheduling: Set the task schedule.

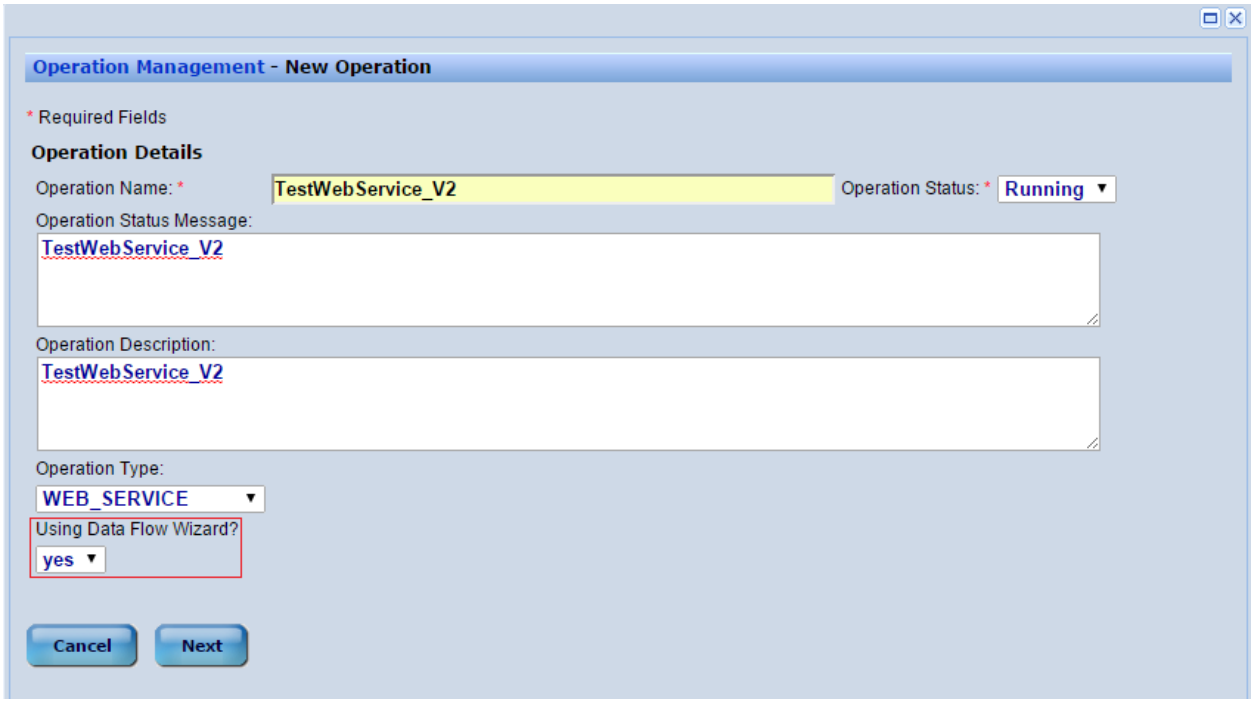
The changes made to the Operation could be saved by clicking on the 

button.

2.4.2.4 Data Flow Wizard Flows

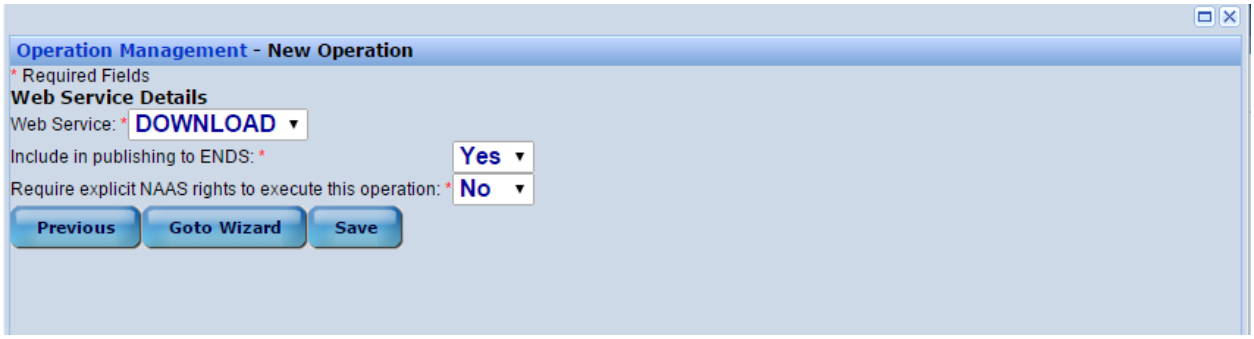
2.4.2.4.1 Operation type is: WEB_SERVICE

Fill the form like below and click “Next” button



- Operation Name: The name of the operation. This field is not editable once created.
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Using Data Flow Wizard: Indicates whether the operation is data wizard driven flow or not. “Yes” option is for data flow wizard driven flow, “No” option is for non-data flow wizard driven flow.

User can fill more detail information in Operation detail page like below:



- Web Service: List of web service. User must select a suitable one.
- Include in publishing to ENDS: if you set this to “Yes”, then any time someone calls the GetServices method on your node, this operation will be included. If you are integrated with EPA’s ENDS, then this will determine if this operation will get registered in ENDS.

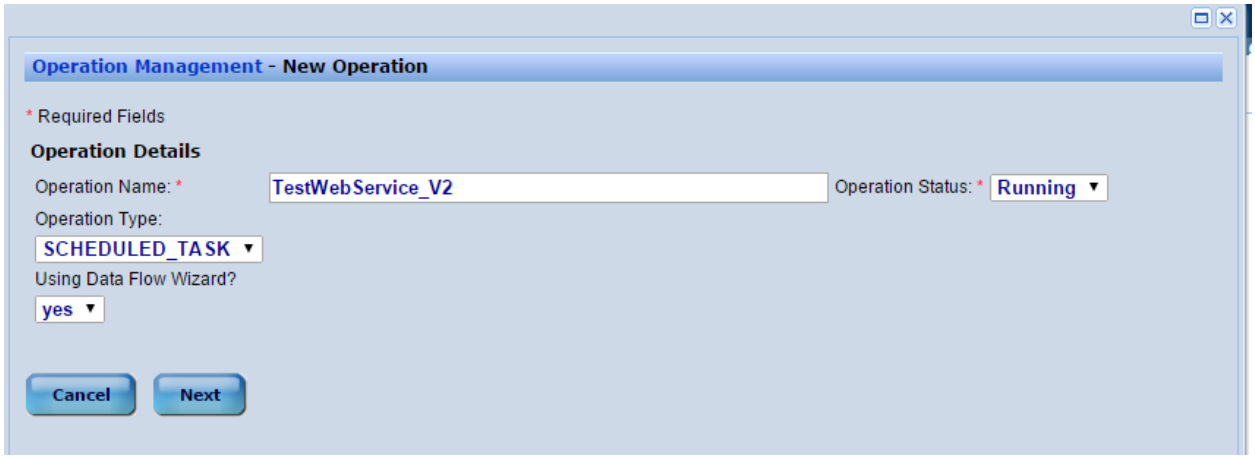
- Require explicit NAAS rights to execute this operation: if you set this to “Yes”, then the only NAAS users who can execute this operation are those that you define at the User Management page. Alternatively, if you set this to “No” then all NAAS users will be able to execute this operation. Note: this rule only applies to Queries and Solicits.

The changes made to the Operation could be saved by clicking on the 

button.

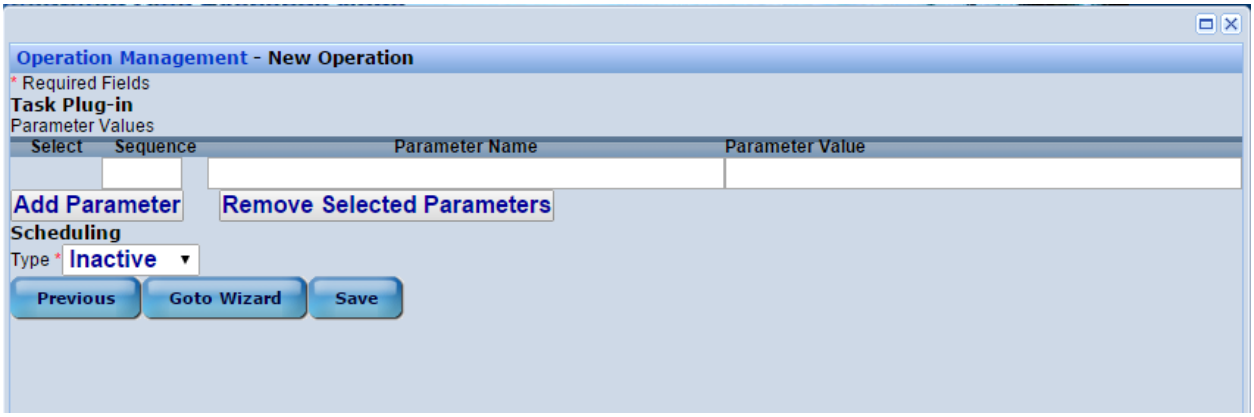
2.4.2.4.2 Operation type is: SCHEDULED_TASK

Fill the form like below and click “Next” button



- Operation Name: The name of the operation. This field is not editable once created.
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Using Data Flow Wizard: Indicates whether the operation is data wizard driven flow or not. “Yes” option is for data flow wizard driven flow, “No” option is for non-data flow wizard driven flow.

User can fill more detail information in Operation detail page like below:

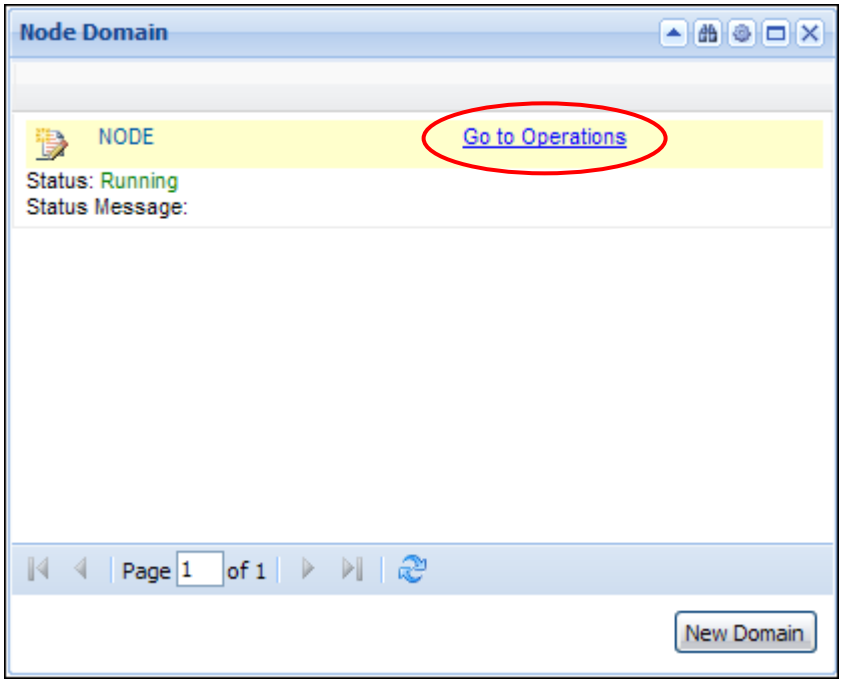


- Parameter Values: The parameters need to pass to task process
- Scheduling: Set the task schedule.

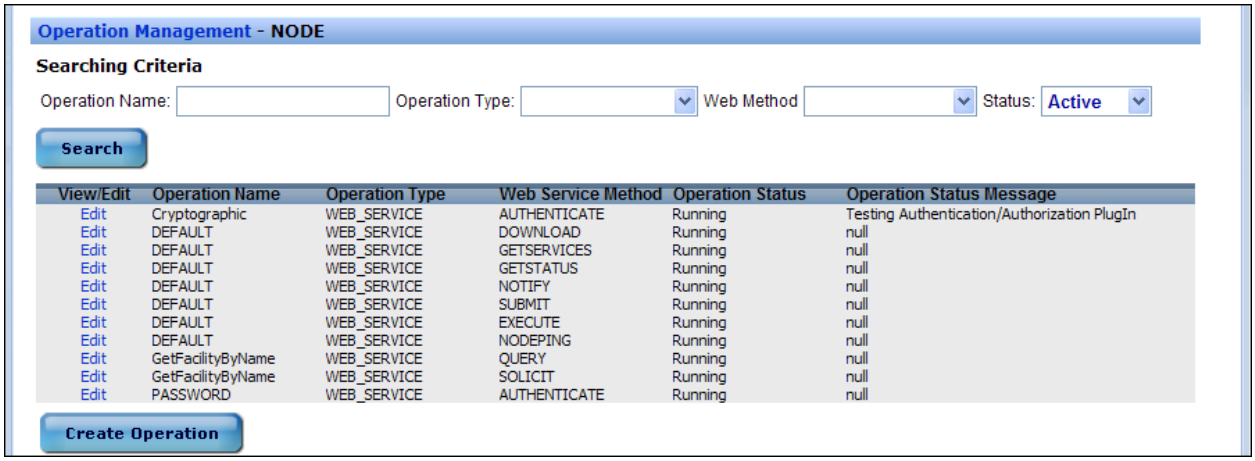
The changes made to the Operation could be saved by clicking on the  button.

2.4.3 Operations Management

In order to edit an existing dataflow to the Node, click on the “Go To Operations” link in the Node Domain Web Part, as shown here:



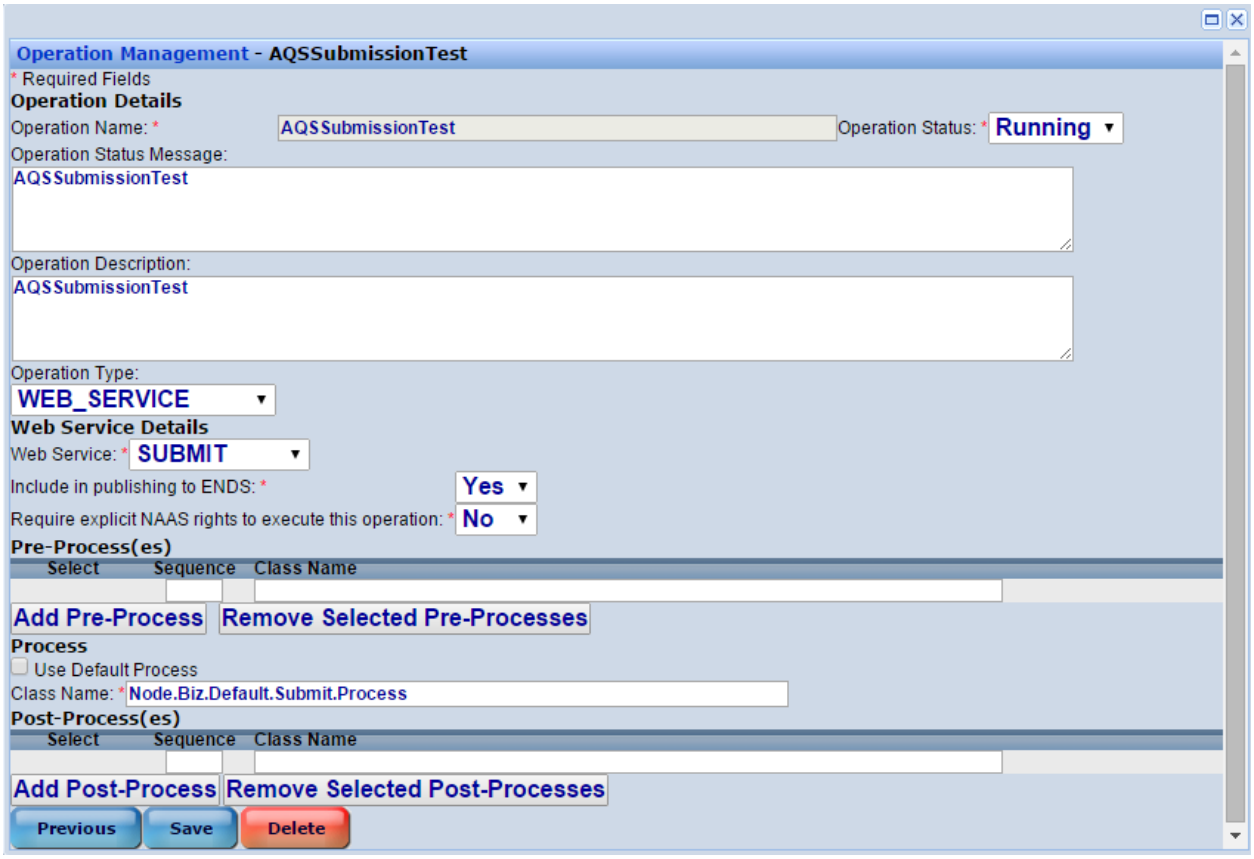
This will take the Admin to the Operations Management screen as shown below:



On this page, the Administrators can click buttons to view, edit, and create Operations for the Domain. This page also allows searching for Operations through the following criteria:

- Operation Name
- Operation Type: Web Service or Scheduled Task
- Web Method used in the Operation
- Status: Active, Inactive, Stopped, Running

If any operations exist in the list, you can click on the Edit link to view/edit the operation details:



This screen allows you to edit the following information:

- Operation Name: The name of the operation. This field is not editable once created.

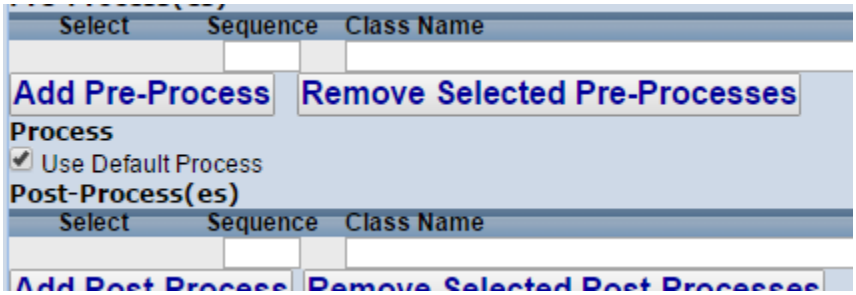
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Web Service: Select the web service
- Include in publishing to ENDS: if you set this to "Yes", then any time someone calls the GetServices method on your node, this operation will be included. If you are integrated with EPA's ENDS, then this will determine if this operation will get registered in ENDS.
- Require explicit NAAS rights to execute this operation: if you set this to "Yes", then the only NAAS users who can execute this operation are those that you define at the User Management page. Alternatively, if you set this to "No" then all NAAS users will be able to execute this operation. Note: this rule only applies to Queries and Solicits.
- Parameter Name: Indicates all parameters.

The changes made to the Operation could be saved by clicking on the 

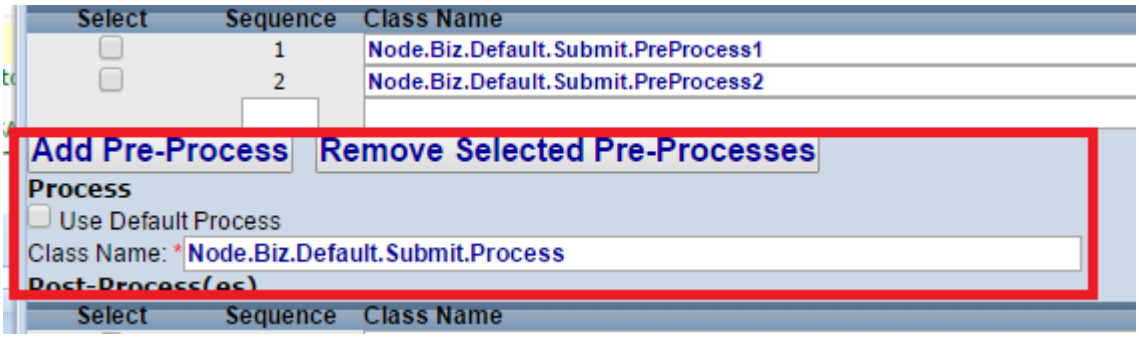
button.

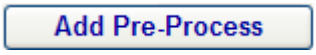
2.4.3.1 Non Data Flow Wizard Flows:

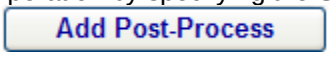
For a Web Service Operation, if the "Use Default Process" checkbox is checked, the Operation implements the "DEFAULT" Operation on the "NODE" domain.



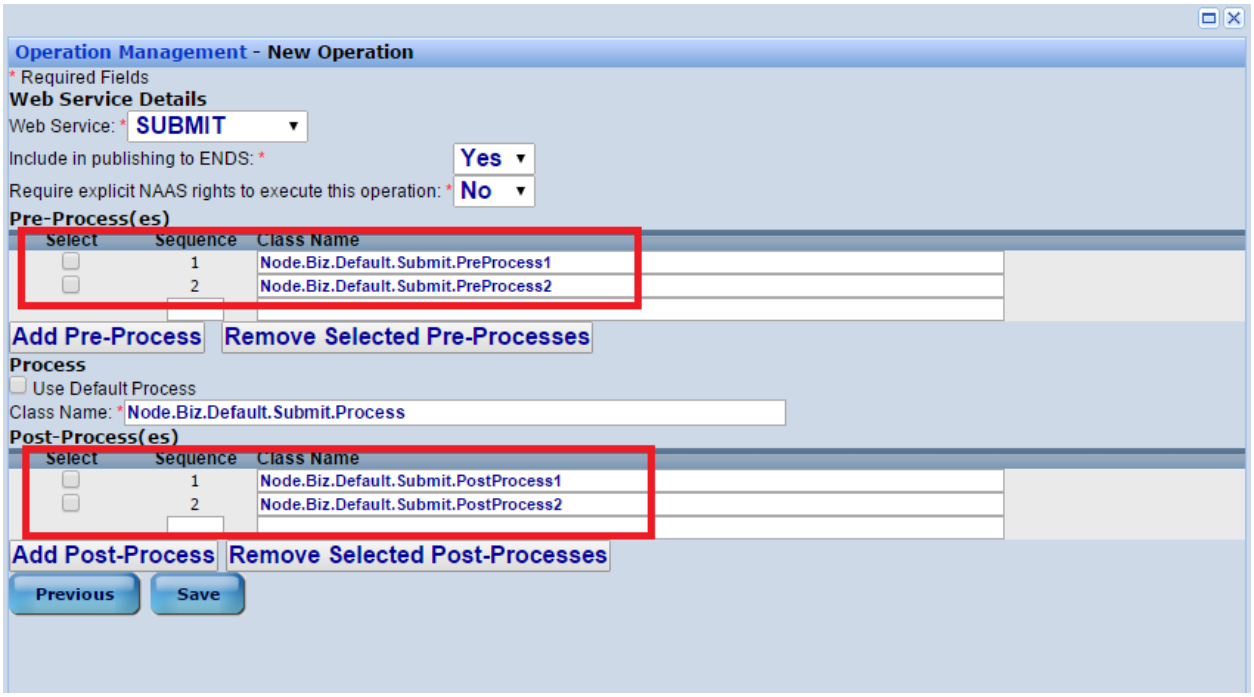
If this is unchecked, a text field will appear on the screen where the user can enter the Class name that will define the process.



New pre-processes and post-processes could be added to the Operation by specifying the Sequence and Class Name and selecting either the 

or the 

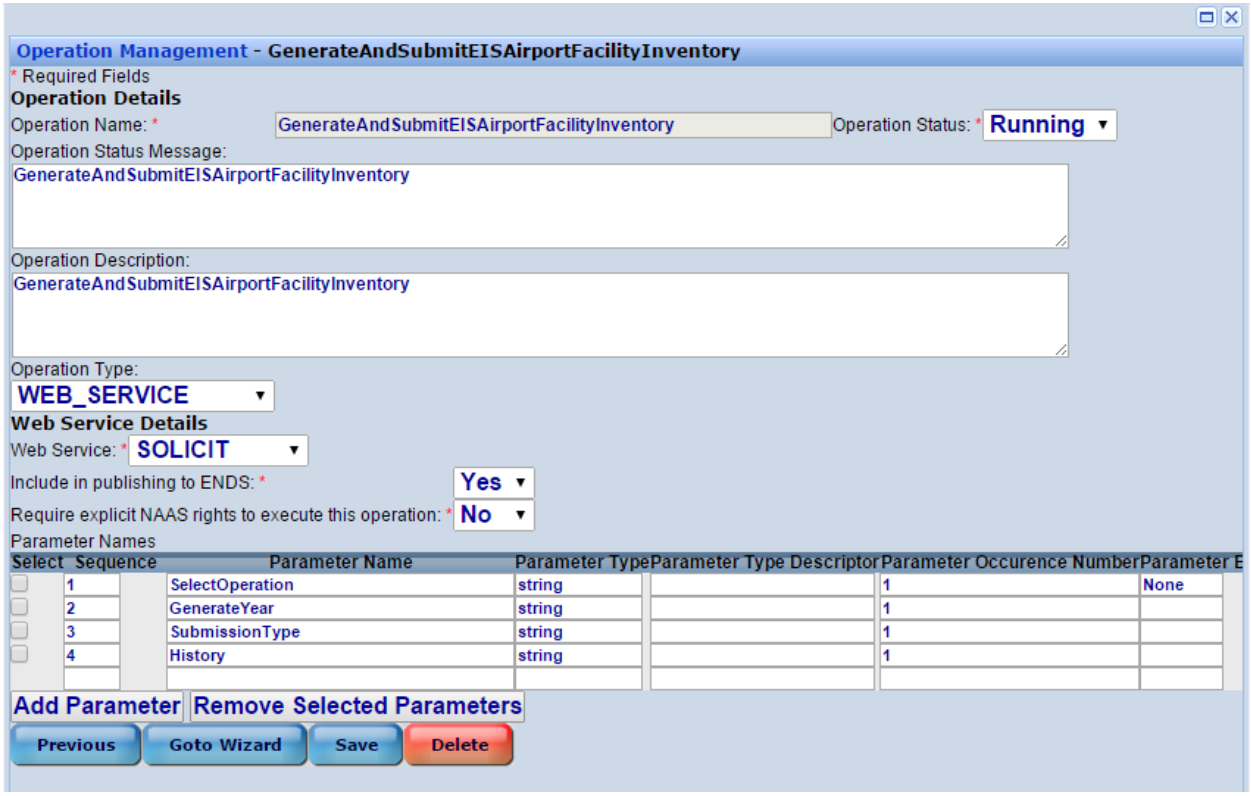
buttons as shown in the figure below. The following screen will be displayed:



Click the 

button to save the operation configuration file.

2.4.3.2 Data Flow Wizard Flows:



This screen allows you to edit the following information:

- Operation Name: The name of the operation. This field is not editable once created.
- Operation Status: Set the status of the Operation as either running, stopped, or inactive.
- Operation Status Message: Specify Operation Status Message, which will be displayed in the Node Status page.
- Operation Description: Specify a description of the operation
- Operation Type: Indicates whether this operation is a Web Service or Task Service.
- Web Service: Select the web service
- Include in publishing to ENDS: if you set this to "Yes", then any time someone calls the GetServices method on your node, this operation will be included. If you are integrated with EPA's ENDS, then this will determine if this operation will get registered in ENDS.
- Require explicit NAAS rights to execute this operation: if you set this to "Yes", then the only NAAS users who can execute this operation are those that you define at the User Management page. Alternatively, if you set this to "No" then all NAAS users will be able to execute this operation. Note: this rule only applies to Queries and Solicits.
- Parameter Name: Indicates all parameters.

The changes made to the Operation could be saved by clicking on the 

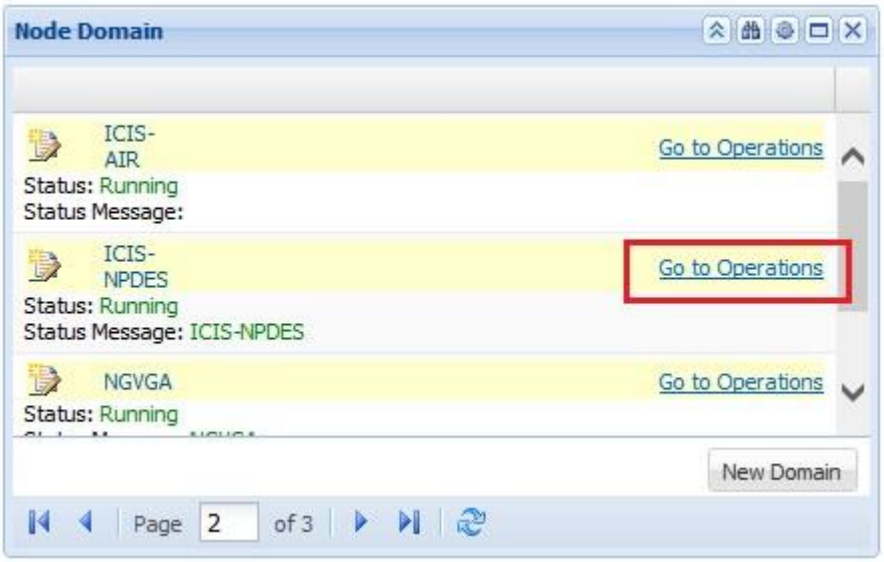
button.

2.4.4 Schedule Task Management

The task dataflow is designed to run as an unattended, automated process. To set or modify the schedule for the task Dataflow, follow these steps:

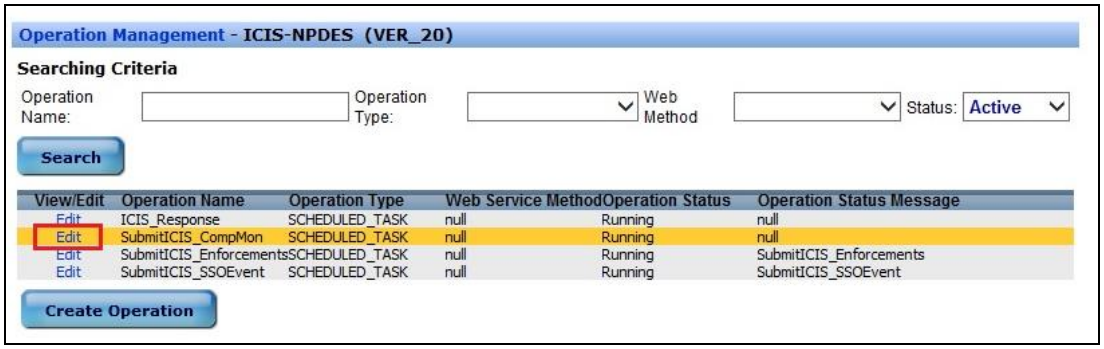
2.4.4.1 Find Operation in domain

Click on the "Go to Operations" link in the Node Domain module , here is the ICIS-NPDES



2.4.4.2 Go to operation configuration page

On the Operations listing page, click the "Edit" link for the operation you wish to schedule.

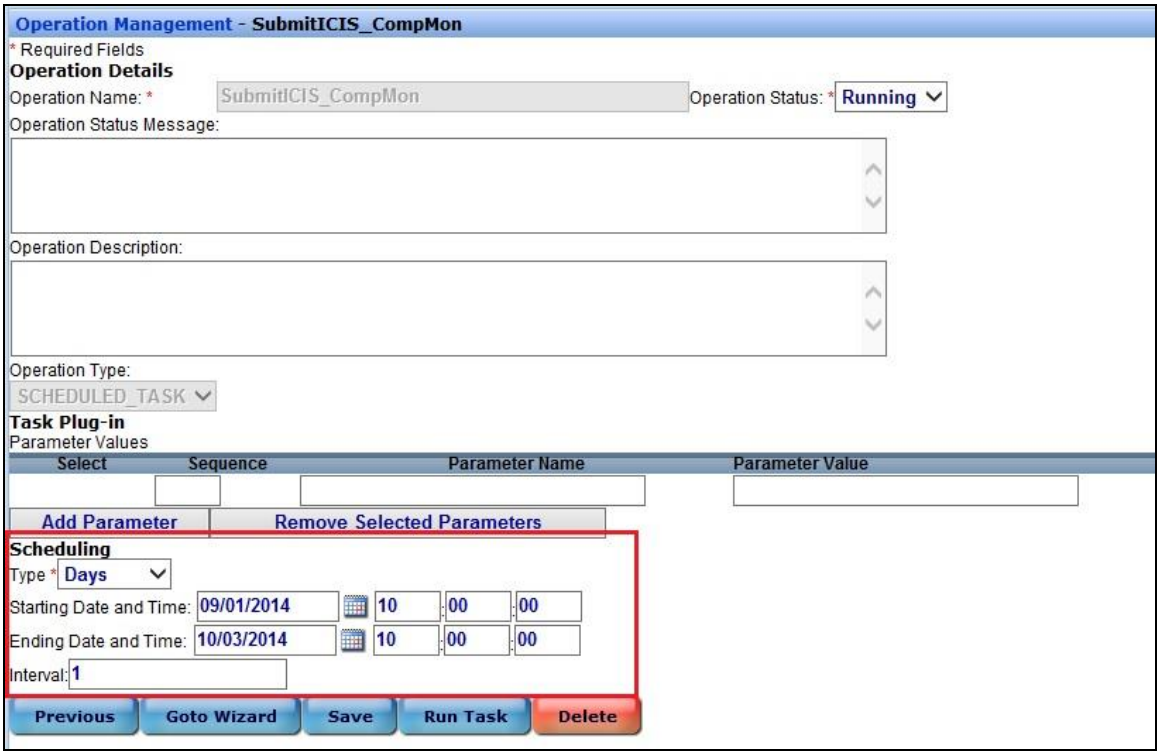


2.4.4.3 Set task schedule

Run Daily: To schedule the Dataflow to run once per day, change the Scheduling Type to “Days”, and set the Interval to “1”. Specify a start date and end date. If you want the task to run indefinitely into the future, specify an end date that is far in the future.

Run Every 15 Minutes: To schedule the Dataflow to run every 15 minutes (as in the case of retrieving error reports from EPA), change the Scheduling Type to “Seconds” and set the Interval to “900”. Specify a start date and end date. If you want the task to run indefinitely into the future, specify an end date that is far in the future.

Run Once: If you want the dataflow to run once, choose “Once” from the schedule drop-down and specify a time you want the dataflow to run. This option is used when you want to make an additional submission on demand.

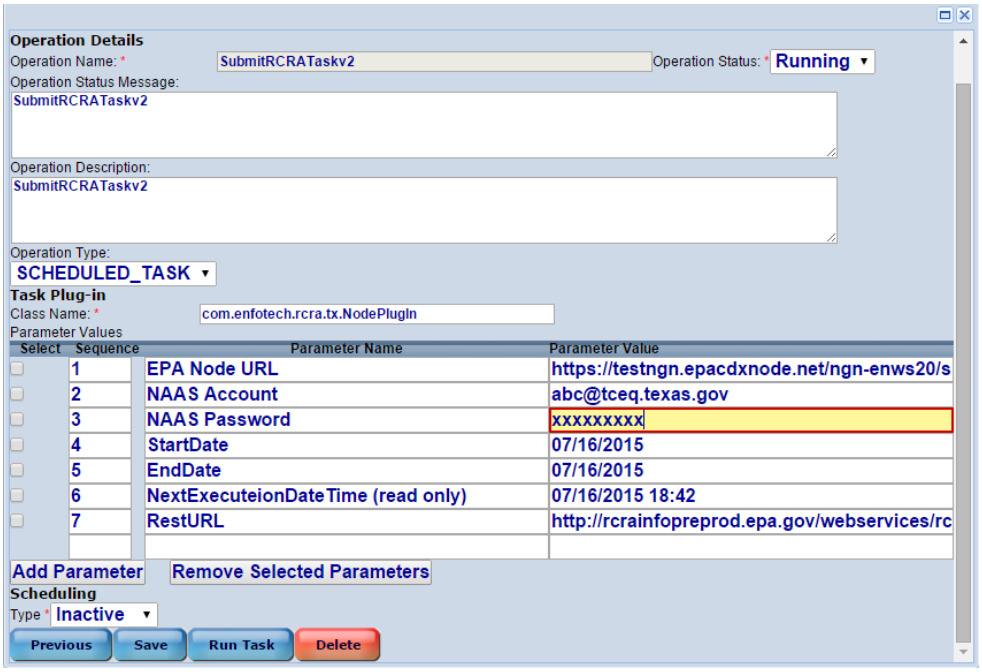


After fill the schedule, click “Save” button or “Run Task” button to save schedule. The “Save” button will save all information on the page such parameters and Operation status etc., the “Run Task” button only save task schedule information.

2.4.4.4 Update parameter values:

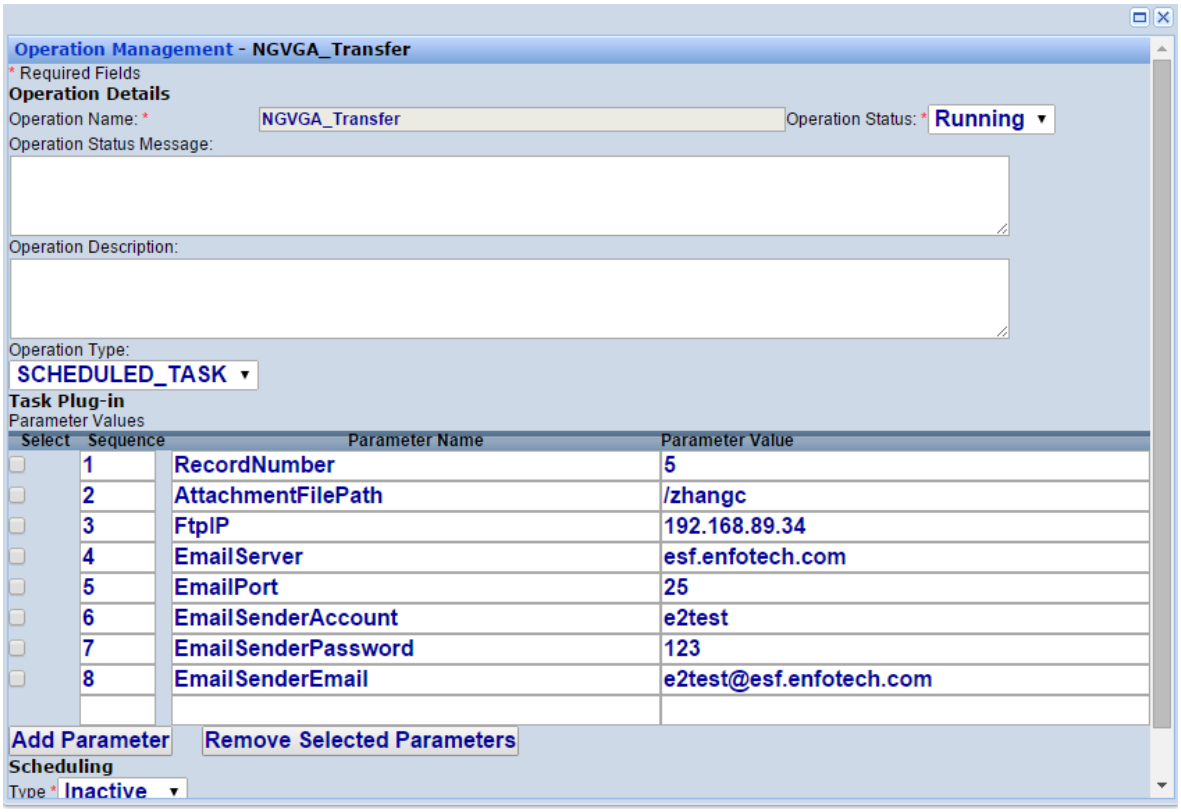
2.4.4.4.1 None Data Flow Wizard Flows:

1. **Simply update the parameters one by one and click “Save” button to save it.**

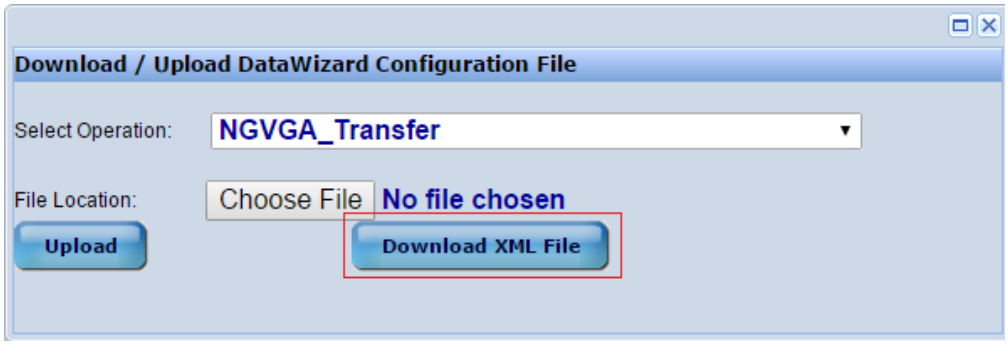


2.4.4.4.2 None Data Flow Wizard Flows:

1. **Update the parameters one by one and click “Save” button to save it.**



2. **Download the data wizard configuration file**

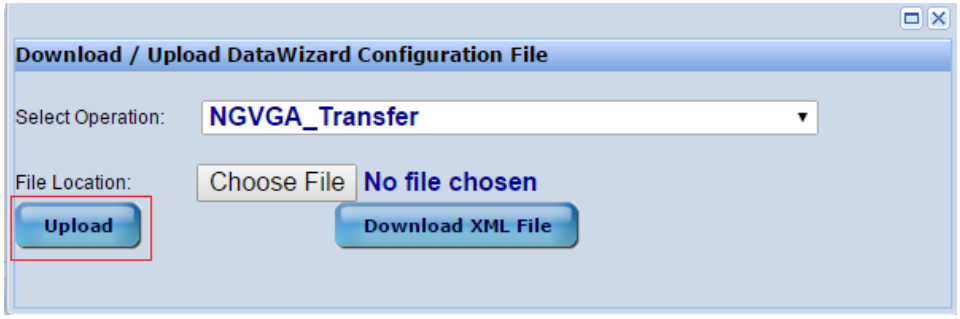


3. **Update the parameters in configuration file to match parameter list in task operation configuration screen**

```
<variables>
        <variable name="webMethodName">Execute</variable>
        <variable name="operationName">NGVGA_Transfer</variable>
        <variable name="transactionId"/>
        <variable name="securityToken"/>
        <variable name="dataflow"/>
        <variable name="request"/>
        <variable name="RecordNumber" />
        <variable name="AttachmentFilePath" />
        <variable name="FtpIP" />
        <variable name="EmailServer" />
        <variable name="EmailPort" />
        <variable name="EmailSenderAccount" />
        <variable name="EmailSenderPassword" />
        <variable name="EmailSenderEmail" />
```

```
        </variables>
        <sequence>
                <invoke partnerLink="ExternalAPI">
                        <extension namespace="">
                                <component aliasName="NGVGAProcessAPI" runAsThread="False"/>
                                <inputParameters>
                                        <parameter name="dllFullName" type="System.String" value=""/>
                                        <parameter name="classFullName" type="System.String"
value="com.enfotech.ngvga.tx.NodePlugIn"/>
                                        <parameter name="logMessage" type="System.String" value=""/>
                                        <parameter name="faultMessage" type="System.String" value=""/>
                                        <parameter name="transactionId" type="System.String"
value="{Global::transactionId}"/>
                                        <parameter name="RecordNumber" type="System.String"
value="{Global::RecordNumber}"/>
                                        <parameter name="AttachmentFilePath" type="System.String"
value="{Global::AttachmentFilePath}"/>
                                        <parameter name="FtpIP" type="System.String" value="{Global::FtpIP}"/>
                                        <parameter name="EmailServer" type="System.String"
value="{Global::EmailServer}"/>
                                        <parameter name="EmailPort" type="System.String"
value="{Global::EmailPort}"/>
                                        <parameter name="EmailSenderAccount" type="System.String"
value="{Global::EmailSenderAccount}"/>
                                        <parameter name="EmailSenderPassword" type="System.String"
value="{Global::EmailSenderPassword}"/>
                                        <parameter name="EmailSenderEmail" type="System.String"
value="{Global::EmailSenderEmail}"/>
                                </inputParameters>
                        </extension>
                </invoke>
```

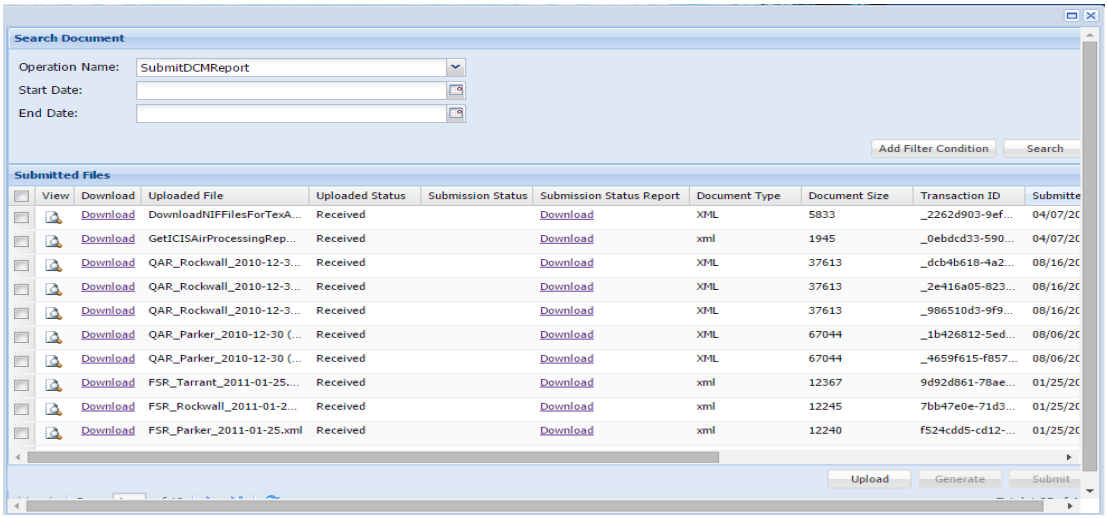
4. **Upload the data wizard configuration file back to Node database**



2.5 Node Operation Manager

The Node Operation Manager screen is accessible from the Favorite Links web part. This screen is useful anytime a Node Admin wants to create a customized page for performing one or more of the following actions:

- Manually initiate the generation of XML files (i.e. “Generate”)
- Manually upload files such as XML files to the Node (i.e. “Upload”)
- Manually submit XML files to an external partner such as EPA (i.e. “Submit”)



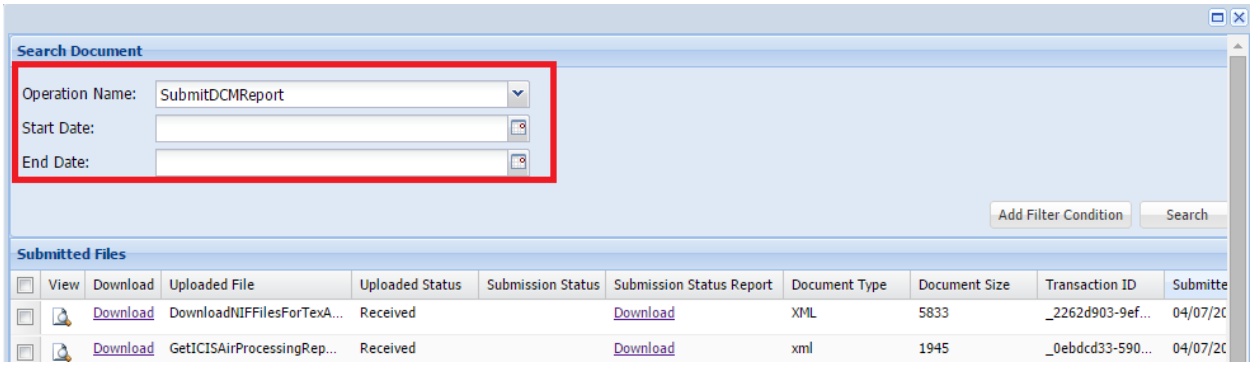
In order to use this screen, a Submit, Query, or Solicit Operation must be created (see Section 0) and then the Operation Manager must also be configured (see Section 2.3).

In the grid view users can do the following:

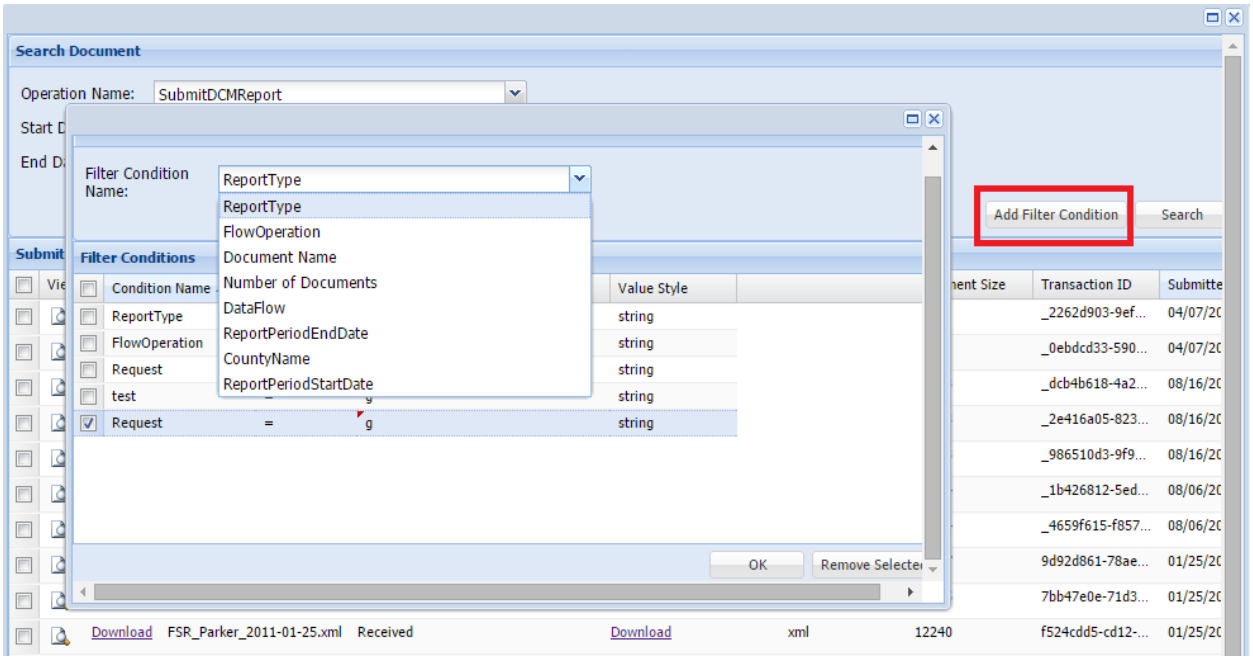
1) Click the “view” link to read the content of each submission
2) Click the “Download” link to download the XML file.
3) Click the “Download” link in the Submission Status Report column to read the error report (for submissions that have failed validation)
4) View the transaction ID, file size of generated XML file, start date and end date of the submission period, and current status

2.5.1 Search Document

The Search Document screen provide the ability to search documents based on operation. User can use three dropdown list to filter the search result:



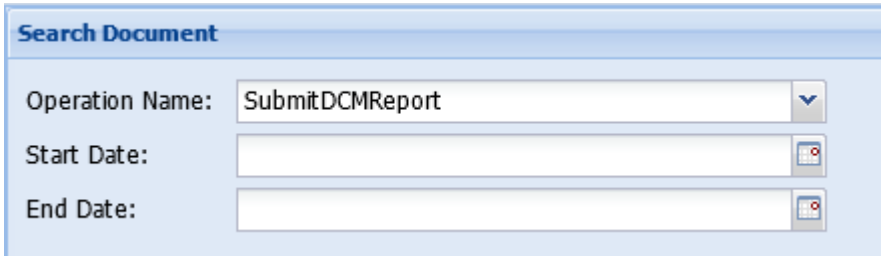
If user need advanced search function, he can click “Add Filter Condition” button to open a jump box window and fill more filter condition based on operation parameter. The “Filter Condition Name” was populated from “operation manager configuration / set filter parameters”. User can select multiple of them and add extra filter condition.



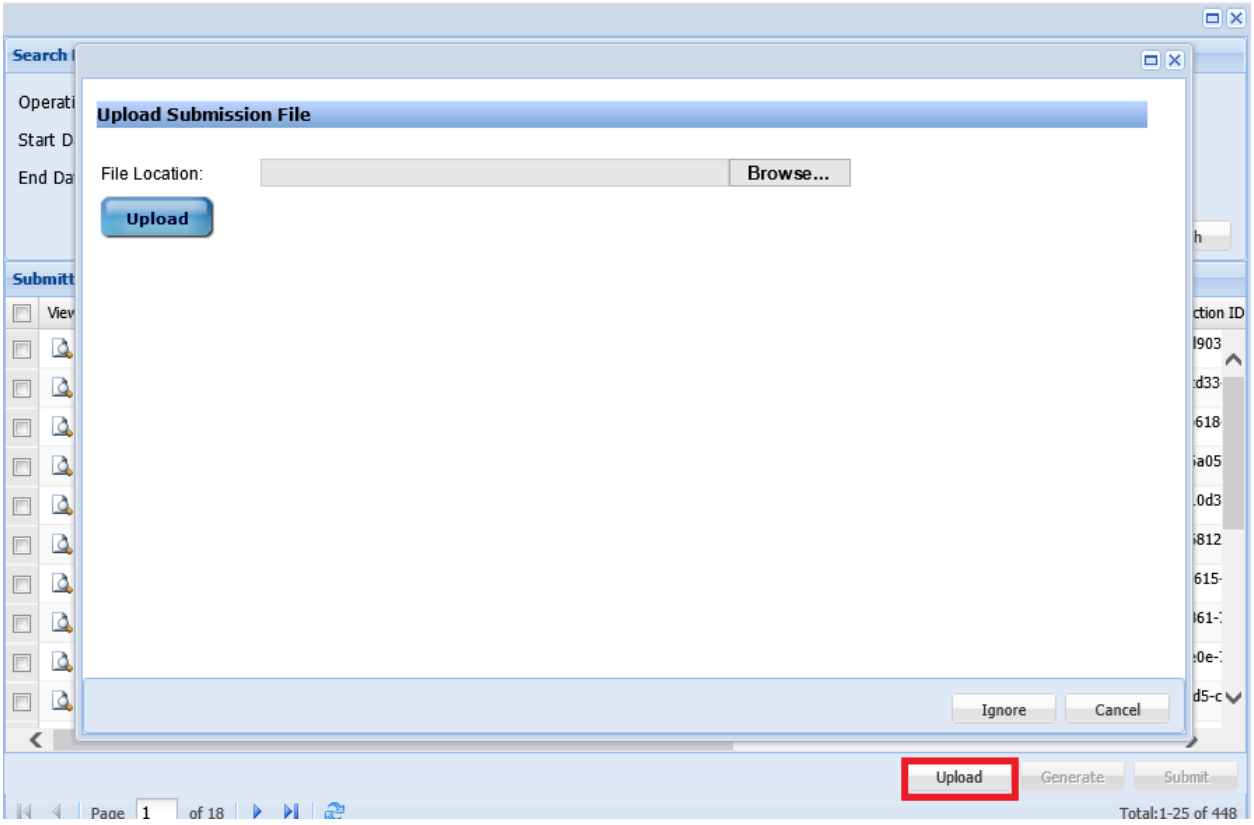
2.5.2 Upload File

In some cases the user may not submit the XML file to TCEQ, but will instead send the Excel file. In this case, TCEQ may need to upload the XML file to the Node. To do this,

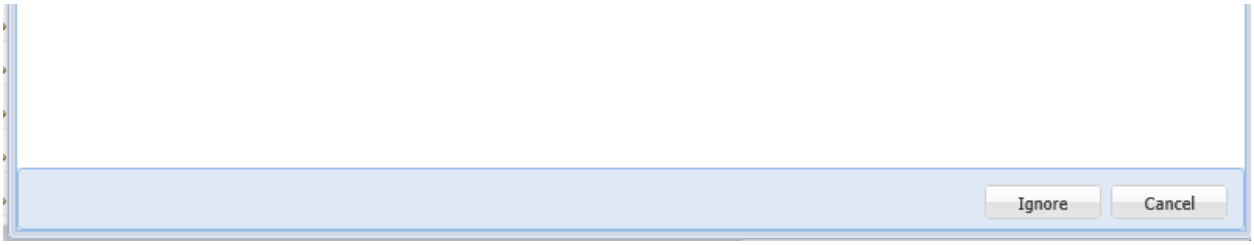
1. Select operation from operation name dropdown list.



2. Click on the Upload button. The system will ask you to select the file to upload.



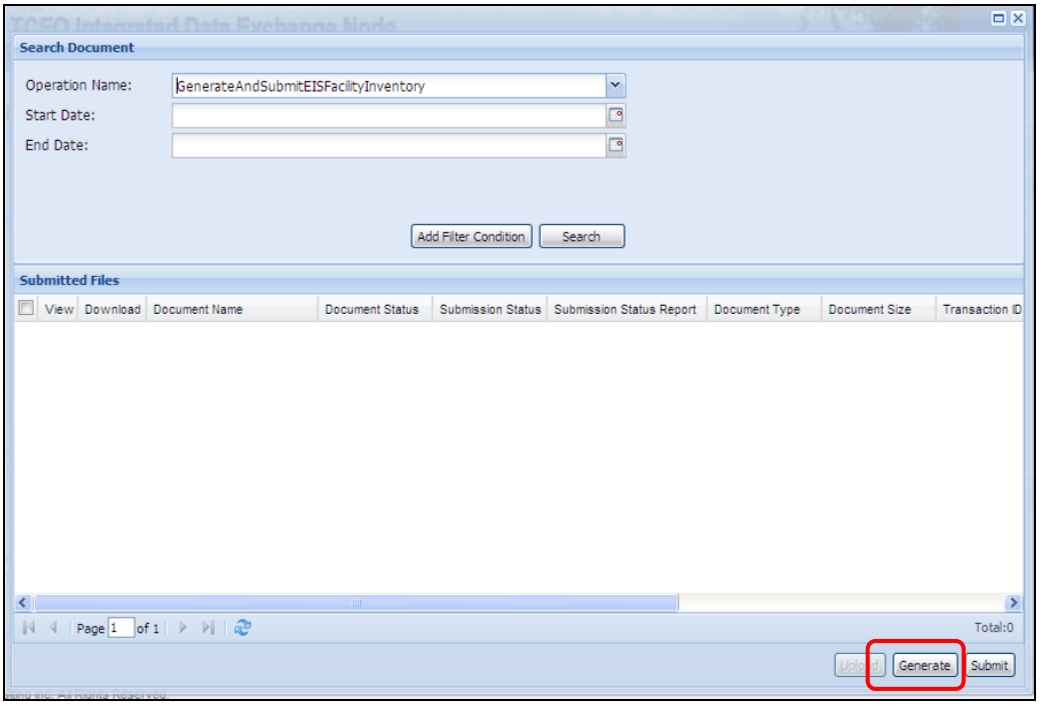
When you upload the file, the system will run the file through validation checks and display the validation results.



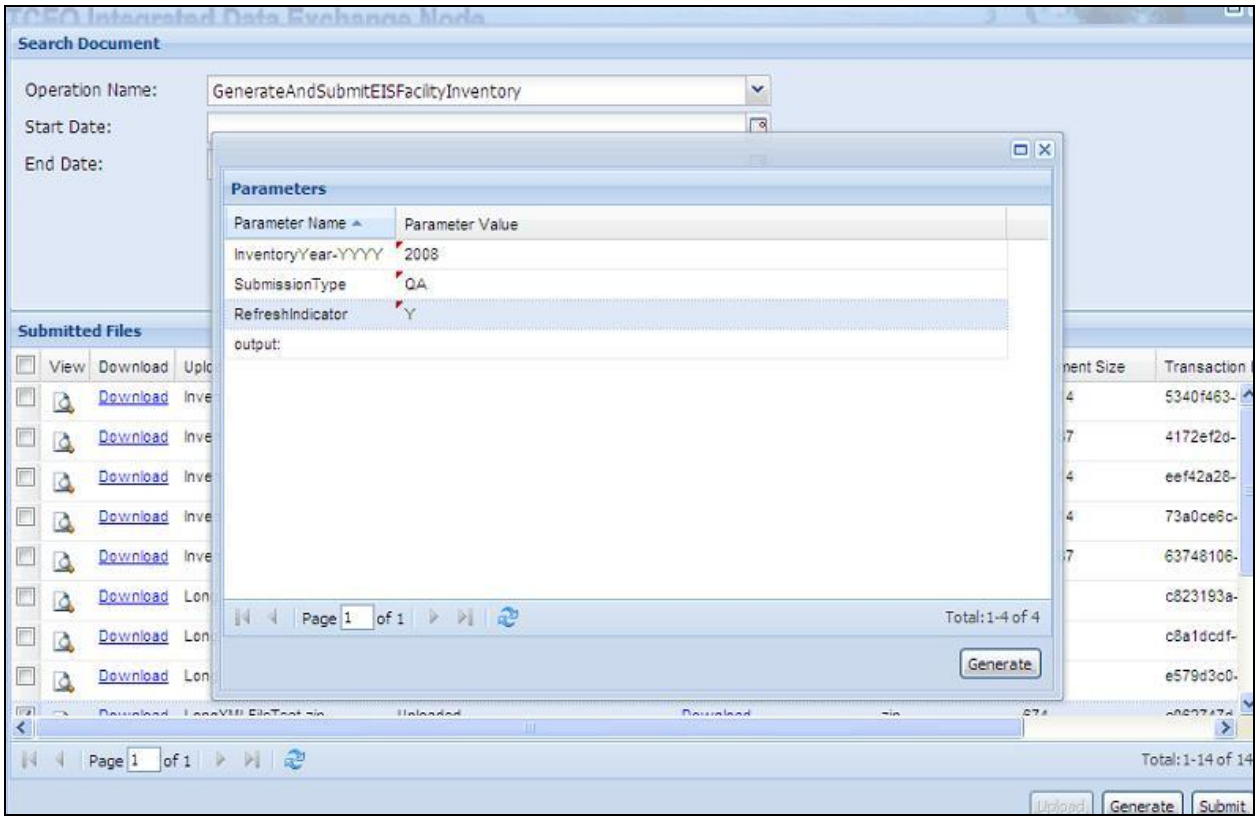
The user can then choose to ignore the validation errors by clicking the “Ignore” button, which will continue with the submission being uploaded to the Node, or click the “Cancel” button, which will terminate the upload process.

2.5.3 Generate File

User can use “Generate” button to call solicit to generate xml file.



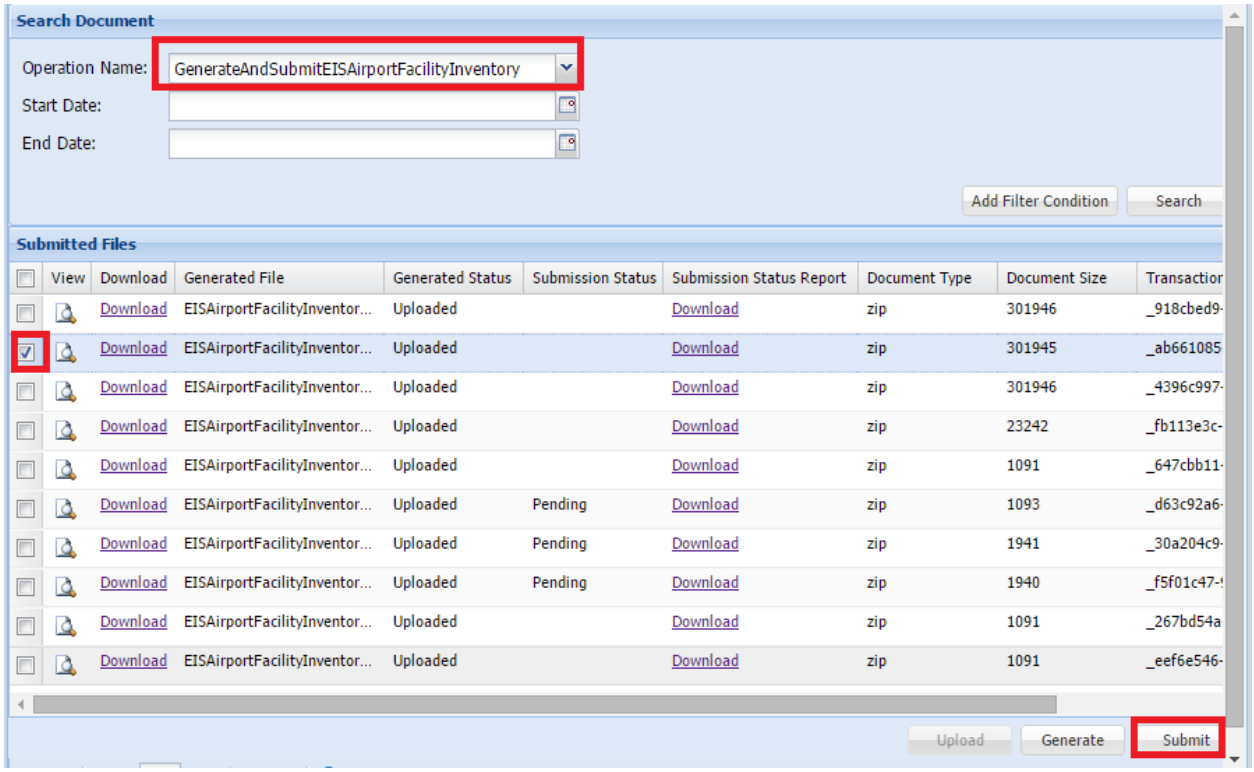
Enter in the parameters values when prompted as shown and click “Generate” button:



2.5.4 Submit File

User can submit any generated file from Node Operation Manager Module. It is simply to

1. Select operation
2. Click the checkbox for this record
3. Click the “Submit” button in the lower right corner to submit to EPA.
4. After the file is submitted, the status will be changed to “Pending”.



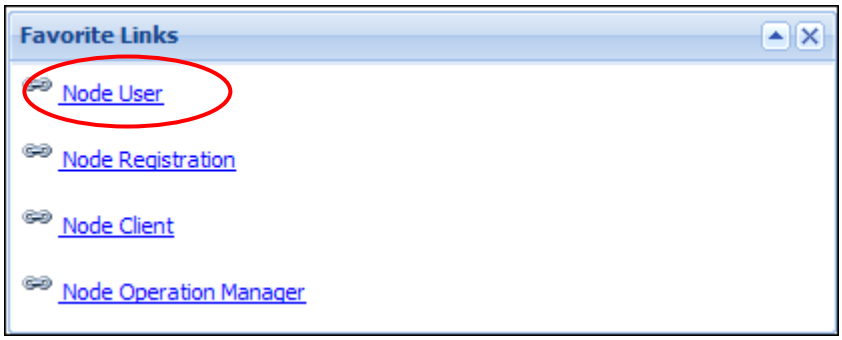
2.6 User Management

The EN-Node manages 3 different types of users:

User Account Types	
Term	**Definition**
Admin Console Users	User who can log in to the Node Administration Console. This consists of: • **Node Admin:** A category of Admin Console User that administers the overall operations of the Node, including Node configuration, user management (Node users and Domain Administrators), and data flow configuration. • **Domain Admin:** A category of Admin Console User that administers one or more domains who can add/update/delete operations within their domain and add/update/remove Node users from their domain.

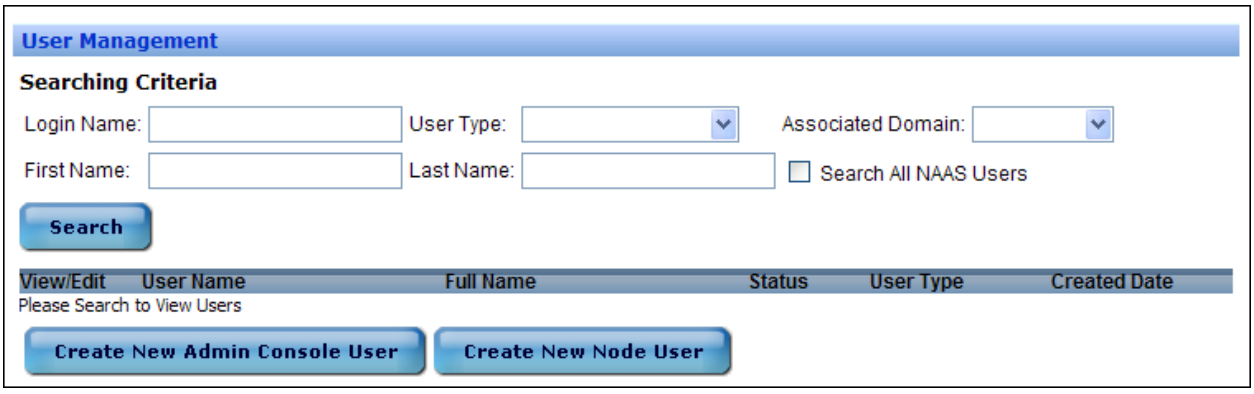
NAAS Users	Users that could use Web Services on the Node and are created and managed through EPA's central NAAS service[2].
Locally Managed Node Users	Users that could use Web Services on the Node and are created and managed at the local Node User who can have privileges to call one or more Web Services on the Node, according to the security policies assigned by the Node and Domain Administrators. Node users can be classified as either NAAS managed Node Users or Locally managed Node Users.

To create or manage the users listed in the table above, access the User Management screen by clicking on the "Node User" link in the Favorite Links Web Part as shown below:



The User management screen allows Administrators to search, view and add users of all the 3 types mentioned above. NAAS users are managed (behind the scenes) directly through NAAS although the User Management screen does provide the ability to create NAAS users. The other users are stored in database table SYS_USER_INFO.

Under user management, Domain Administrators will be able to view only those users associated to their Domains.



The figure above shows the main page for User management. At this page the user can search for users using the following criteria:

[2] NAAS stands for Network Authentication and Authorization Service, and is a service hosted by EPA's Central Data Exchange. It provides a central mechanism for managing Node account information and Node access rights. Domain Administrators are encouraged to use NAAS to manage their Node users.

- Login Name
- User Type: which can be either “Local User”, “NAAS User”, and “Console User”
- Associated Domain (Lists only those Domains that are assigned to the Domain Admin)
- First/Last Name
- Search All NAAS Users: This will show all the NAAS users in the nation, including those who do not have rights on the Node. This is useful when you need to associate an existing NAAS user to your node’s dataflow. Caution: this query sometimes may take some time to return data.

At the User Management Main Page, clicking on the 

button allows the Administrator to create new Node Administrators or Domain Administrators.

A new NAAS node user or local node user can be created by clicking on the 

button.

Clicking the 

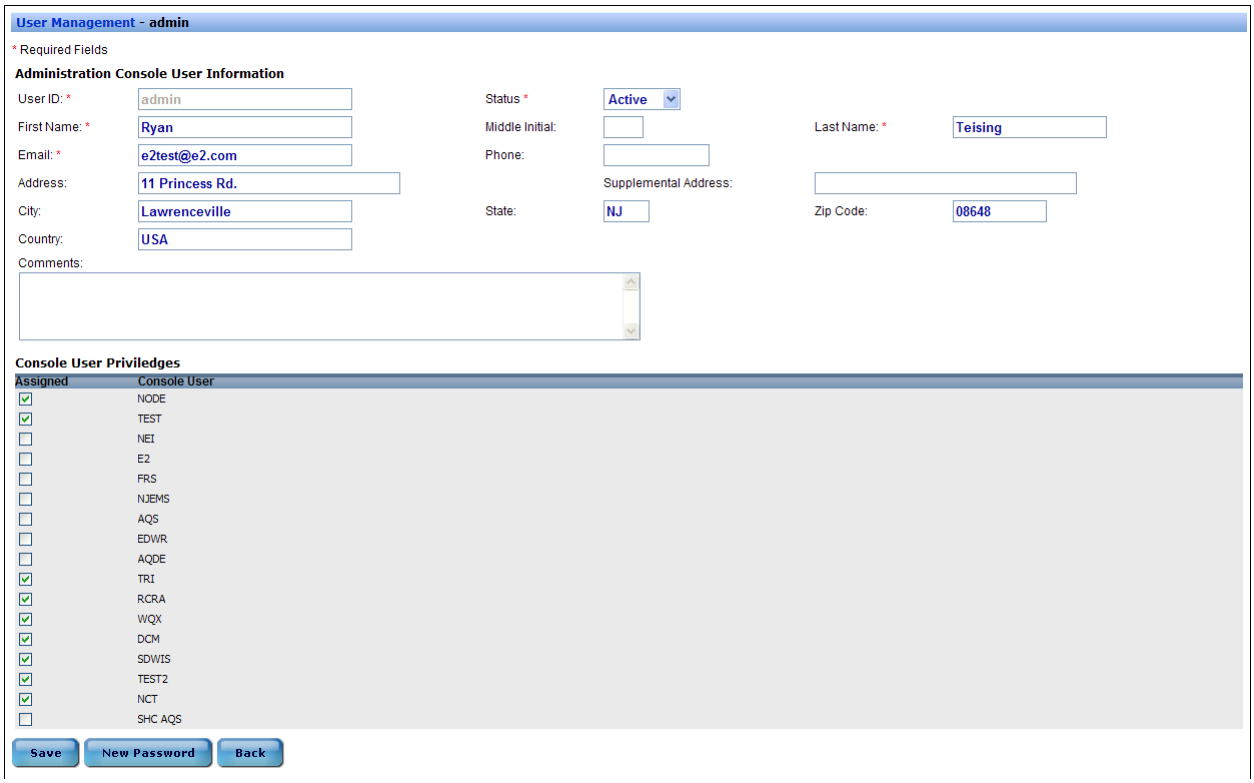
button will return the results meeting the entered criteria. The user can leave search fields blank to return all entries or enter partial strings.

2.6.1 Viewing User Details

The Admin can click on the View link, in the leftmost column to view details of the User and modify it as needed. The details that are displayed and the options that are available upon clicking on the link depend on the category of the user that was specified: Console User, Local Node User, or NAAS Node User.

Console User Details:

The Console User Details screen allows the Administrator to setup details for additional Admin Console Users. General account information can be edited, as shown here:



The Admin could be assigned to one or more domains (that the Domain Admin has rights to) by checking the boxes next to the Domains listed at the bottom of the page. The updated information is saved using the Save button. The user could change their password by clicking the New Password button. Once the button is clicked, a new password is system generated and emailed to the Node/Domain Administrator's Email address. The Administrator may remove the Admin Console User by clicking the Remove User button, which will inactivate the User's account and prevent the User from accessing the EN-Node system.

NAAS and Local Node User Details:

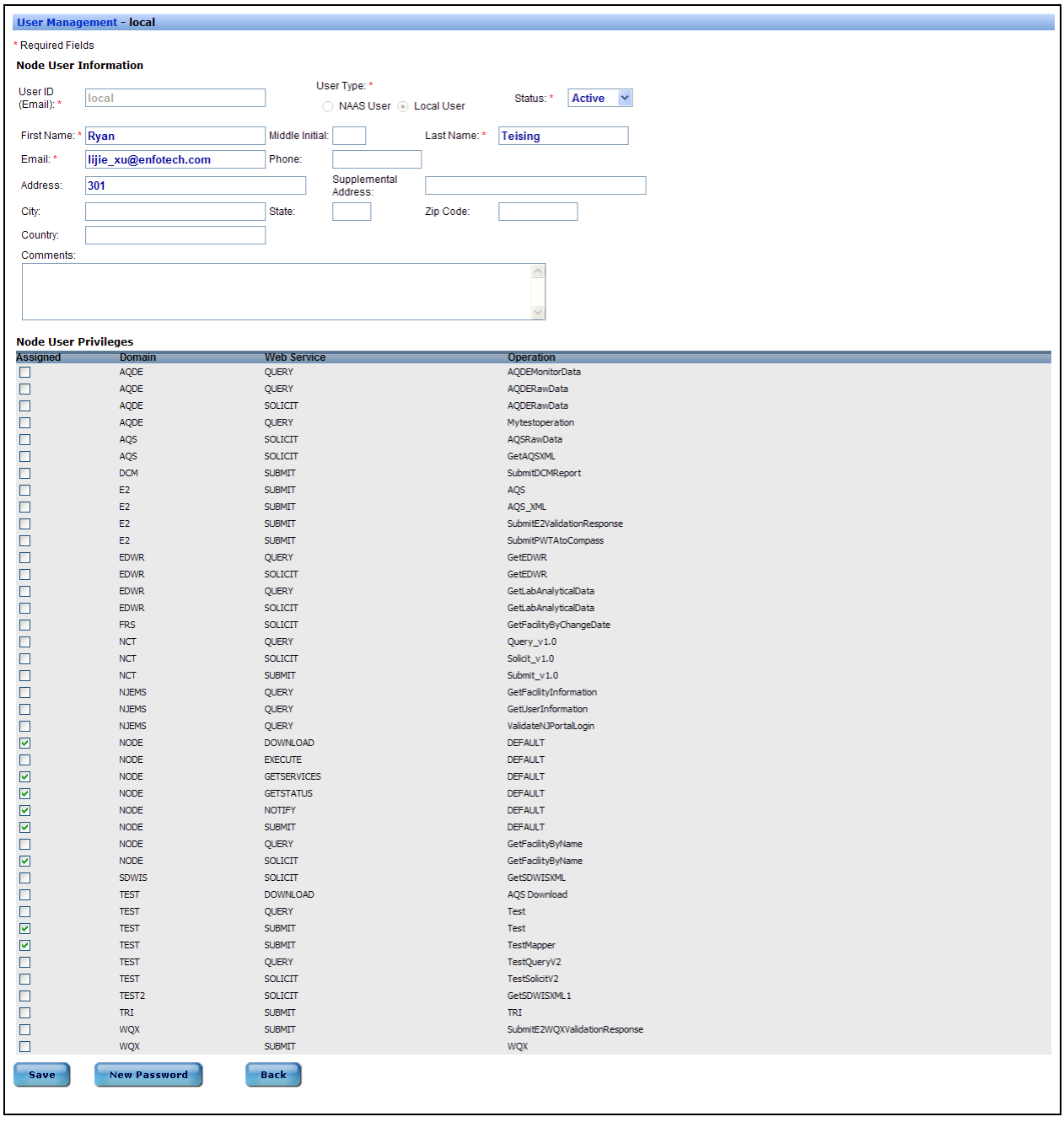
Details for NAAS and locally managed Node users can be managed. In particular you can assign the accounts rights to one or more operations. By default, EN-Node prevents a node user from being able to access a web service until they are explicitly granted rights. There are some differences between Local Node Users and NAAS Node Users, as described here:

Local Node Users:

- Can have more information edited as opposed to NAAS users
- Can be set to active or inactive
- The user could change the password by clicking the New Password button. Once the button is clicked, a new password is generated by the system and sent to the user's Email address. If there is an error happened, user will get "Failed Password Generation, Please check database, Email server or user Email account." message.

NAAS Node users:

- Since the NAAS users are not stored in the local database, the information showed for the NAAS users is only the user name (which is their email address) and the associated operations they have rights to.
- This information is pulled from EPA's NAAS server by performing Web service calls to NAAS. The associated domains and operations could be managed at this screen but as mentioned above, these modifications would be handled via NAAS. This will directly update the information stored on EPA's NAAS server by making a Web service call to NAAS.



User Management - local

* Required Fields

Node User Information

User ID (Email): * local

User Type: * NAAS User / Local User

Status: * Active

First Name: * Ryan Middle Initial: Last Name: * Teising

Email: * lijie_xu@enfotech.com Phone:

Address: 301 Supplemental Address:

City: State: Zip Code:

Country:

Comments:

Node User Privileges

Assigned	Domain	Web Service	Operation
☐	AQDE	QUERY	AQDEMonitorData
☐	AQDE	QUERY	AQDERawData
☐	AQDE	SOLICIT	AQDERawData
☐	AQDE	QUERY	Mytestoperation
☐	AQS	SOLICIT	AQSRawData
☐	AQS	SOLICIT	GetAQSXML
☐	DCM	SUBMIT	SubmitDCMReport
☐	E2	SUBMIT	AQS
☐	E2	SUBMIT	AQS_XML
☐	E2	SUBMIT	SubmitE2ValidationResponse
☐	E2	SUBMIT	SubmitPWTAtoCompass
☐	EDWR	QUERY	GetEDWR
☐	EDWR	SOLICIT	GetEDWR
☐	EDWR	QUERY	GetLabAnalyticalData
☐	EDWR	SOLICIT	GetLabAnalyticalData
☐	FRS	SOLICIT	GetFacilityByChangeDate
☐	NCT	QUERY	Query_v1.0
☐	NCT	SOLICIT	Solicit_v1.0
☐	NCT	SUBMIT	Submit_v1.0
☐	NJEMS	QUERY	GetFacilityInformation
☐	NJEMS	QUERY	GetUserInformation
☐	NJEMS	QUERY	ValidateNJPortalLogin
☑	NODE	DOWNLOAD	DEFAULT
☐	NODE	EXECUTE	DEFAULT
☑	NODE	GETSERVICES	DEFAULT
☑	NODE	GETSTATUS	DEFAULT
☑	NODE	NOTIFY	DEFAULT
☑	NODE	SUBMIT	DEFAULT
☐	NODE	QUERY	GetFacilityByName
☑	NODE	SOLICIT	GetFacilityByName
☐	SDWIS	SOLICIT	GetSDWISXML
☐	TEST	DOWNLOAD	AQS Download
☐	TEST	QUERY	Test
☑	TEST	SUBMIT	Test
☑	TEST	SUBMIT	TestMapper
☐	TEST	QUERY	TestQueryV2
☐	TEST	SOLICIT	TestSolicitV2
☐	TEST2	SOLICIT	GetSDWISXML1
☐	TRI	SUBMIT	TRI
☐	WQX	SUBMIT	SubmitE2WQXValidationResponse
☐	WQX	SUBMIT	WQX

Save New Password Back

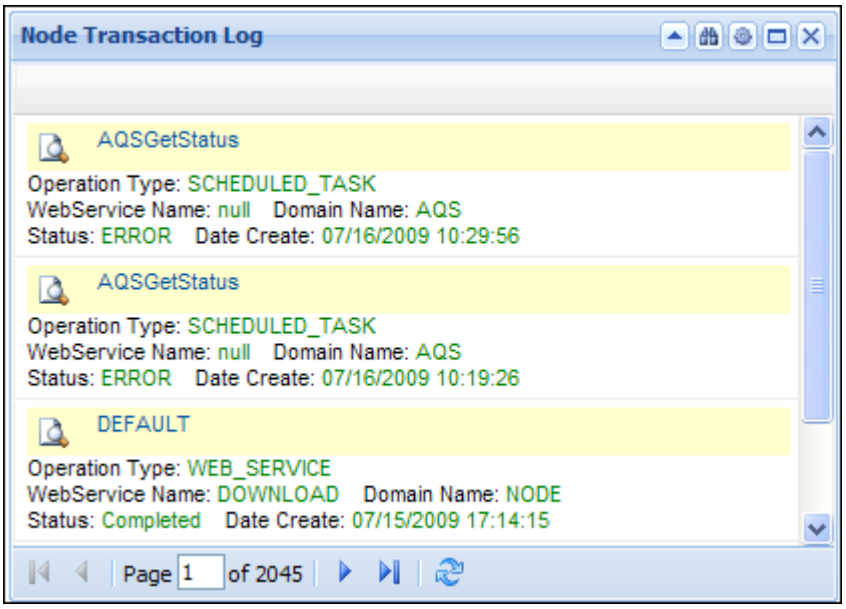
The updated information could be saved using the 

button. The Administrator may remove the User from the system by clicking the 

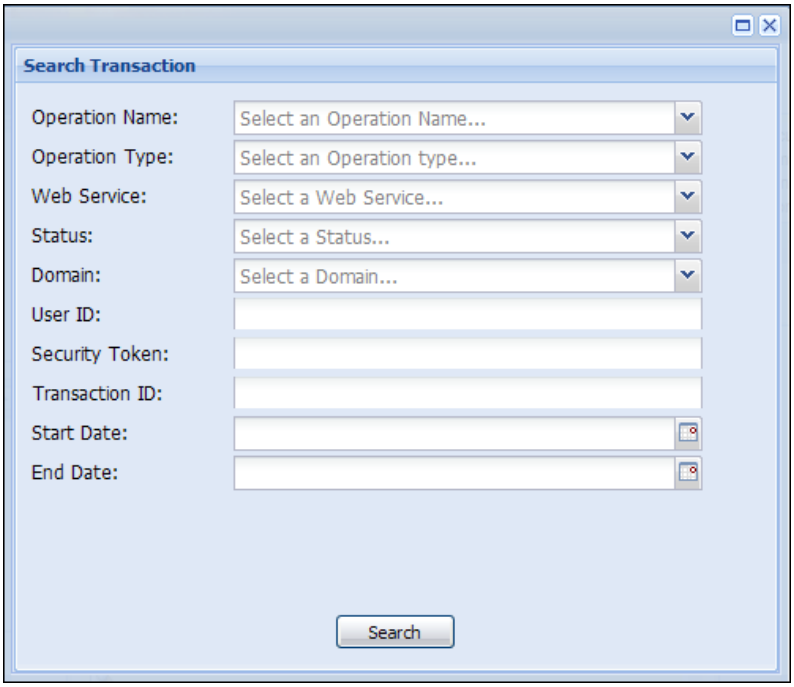
button.

2.7 Viewing the Node Transaction Log

The Node Transaction log stores a history of all instances where the operations of the Node were invoked, or attempted to be invoked. In addition, any task logging that has been logged to the node logging tables can be viewed here. This screen is populated by pulling data from the Node_Operation_Log, Node_Operation_Log_Parameter, and Node_Operation_Log_Status tables.

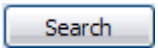


By click on the icon  , the search screen will be shown:

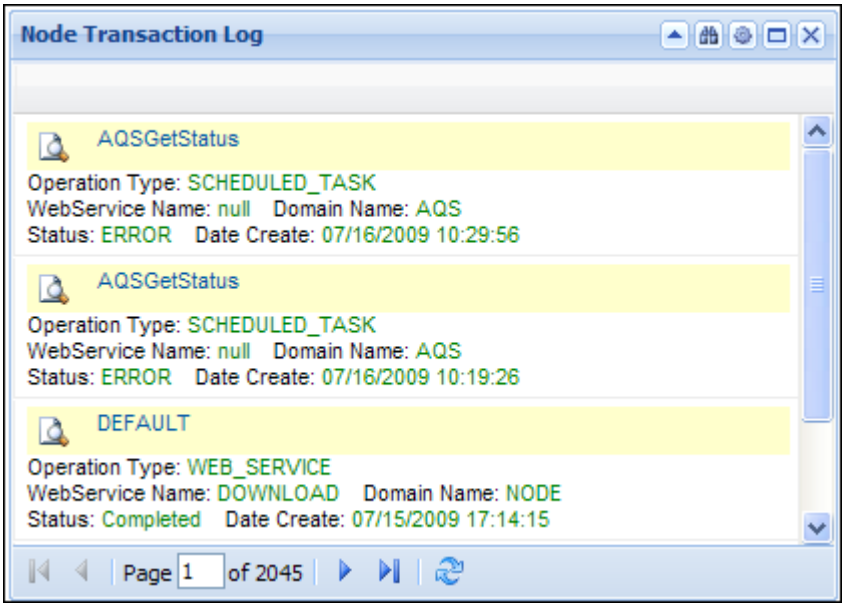


At this screen the user can search the Node Transaction Log based on:

- Operation Name
- Operation Type
- Web Service Name
- Status (corresponds to the current status of an Operation)
- Domain Name (for Domain Administrators, this will be limited to only those Domains controlled by the Admin)
- User ID
- Security Token
- Transaction ID
- Date range

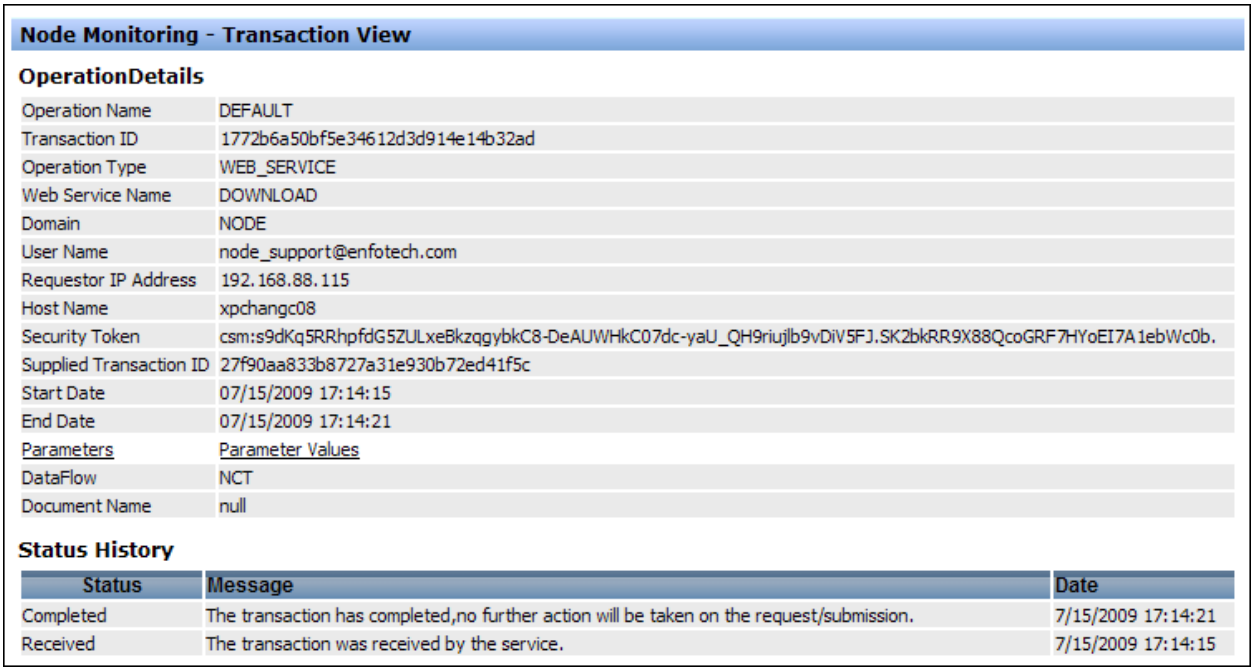
Clicking on the 

button will return the results, meeting the entered criteria, to the Node Transaction Log Web Part. The user can leave all fields blank to return all entries or use the % symbol as a wildcard.



Within the returned list of log results, the user can click on the icon  , to view further details of the operation. The Web Part displays the following attributes:

- Operation Name
- Transaction ID: the unique transaction ID generated by the Node for the specific Node request
- Operation Type: The type of operation, either Web Service, or Scheduled Operation
- Web Service Name: For WEB_SERVICE operation types, the web service that is being invoked.
- Domain: the domain to which the logged operation belongs
- User Name: the name of the user (if known) that is making the web service request
- Requester IP Address: The IP address of the requester
- Host Name: The name of the host (if known) that is making the web service request
- Security Token: The security token issued to the requester during authentication
- Supplied Transaction ID: in some cases such as Download, the transaction ID provided by the requester
- Starting Date / Time for the Web service request: Date / Time a task started (for scheduled operations) or the Date / Time the service request was made
- End Date/Time: Date / Time a task completed (for scheduled operations) or the Date / Time the service request response was made
- Parameters: Parameters associated with web service request
- DataFlow: Dataflow associated with the web service request
- Document Name: The name of the document associated with the web service request
- Node Operation Transaction Status History



Node Monitoring - Transaction View

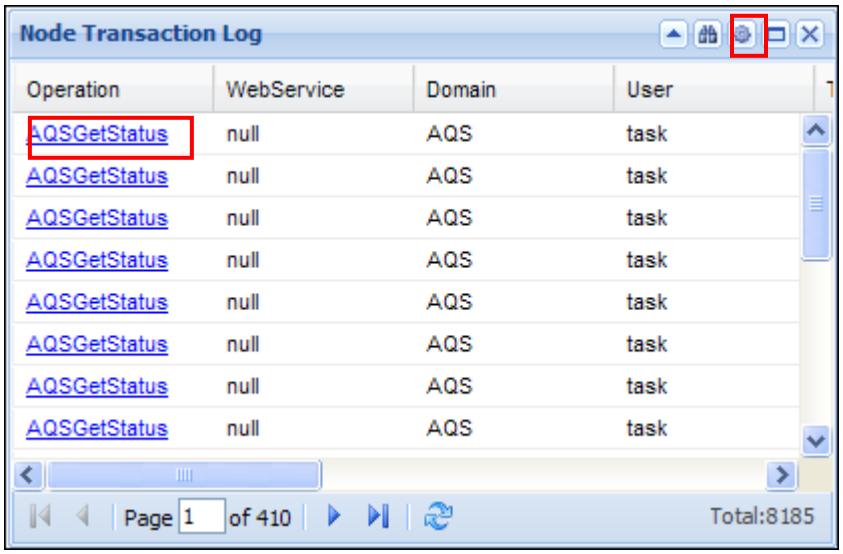
OperationDetails

Operation Name	DEFAULT
Transaction ID	1772b6a50bf5e34612d3d914e14b32ad
Operation Type	WEB_SERVICE
Web Service Name	DOWNLOAD
Domain	NODE
User Name	node_support@enfotech.com
Requestor IP Address	192.168.88.115
Host Name	xpchangc08
Security Token	csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riujlb9vDiV5FJ.SK2bkRR9X88QcoGRF7HYoEI7A1ebWc0b.
Supplied Transaction ID	27f90aa833b8727a31e930b72ed41f5c
Start Date	07/15/2009 17:14:15
End Date	07/15/2009 17:14:21
Parameters	Parameter Values
DataFlow	NCT
Document Name	null

Status History

Status	Message	Date
Completed	The transaction has completed,no further action will be taken on the request/submission.	7/15/2009 17:14:21
Received	The transaction was received by the service.	7/15/2009 17:14:15

The Node Transaction log detail can also be accessed by clicking on the “click to switch view model icon” and then by clicking on the Operation Link. This scenario is shown in the figure below:



Node Transaction Log

Operation	WebService	Domain	User
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task
AQSGetStatus	null	AQS	task

Page 1 of 410 Total:8185

2.7.1 Additional Logging

The majority of Node activity will be tracked under the Node Monitoring section of the Node Administration Console. However, there may be times when a Node administrator needs to check some detailed Exception logs. This may be the case during a troubleshooting exercise. In this case, four additional log files can be retrieved from the Node Server for each component of the Node:

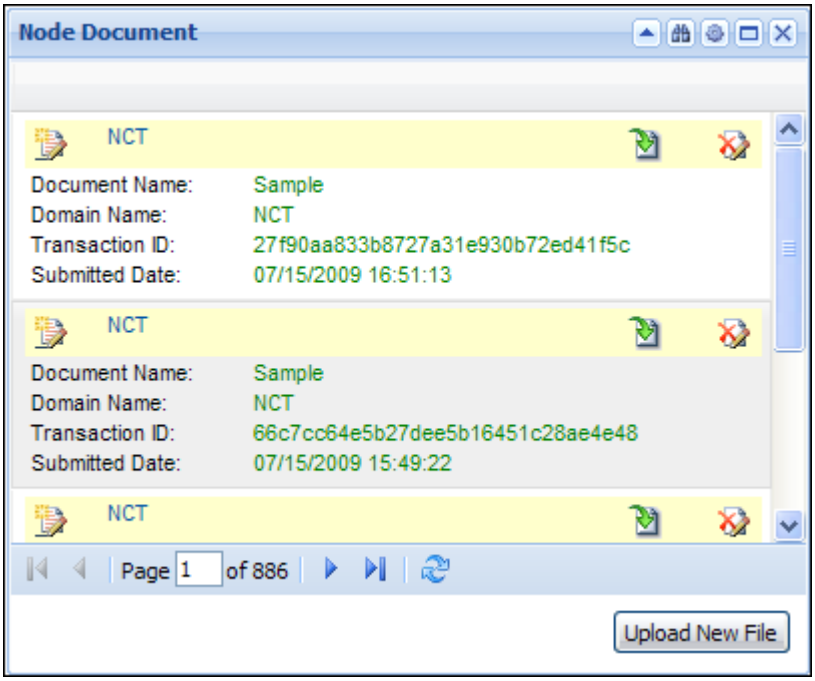
- NodeAdministrationLog.txt
- NodeClientLog.txt
- NodeTaskLog.txt

- NodeWebServicesLog.txt

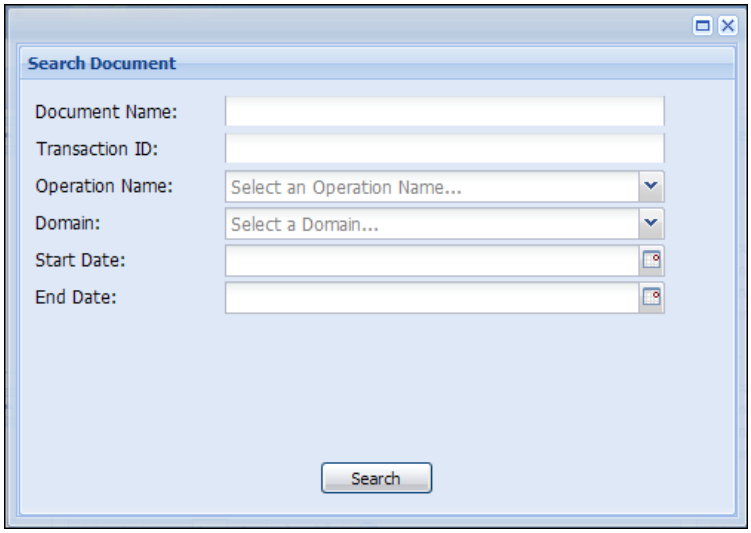
The directory location and filename of these log files could be modified from the application config file present in each component of the Node. Also user can set the log level in “Node Configuration” module

2.8 Document Management

The Document Management screen keeps track of the files stored in the node database. This could include files that have been uploaded to the node from external users, or made available on the node for download.

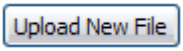


By clicking on the icon  , the following search screen will be shown:

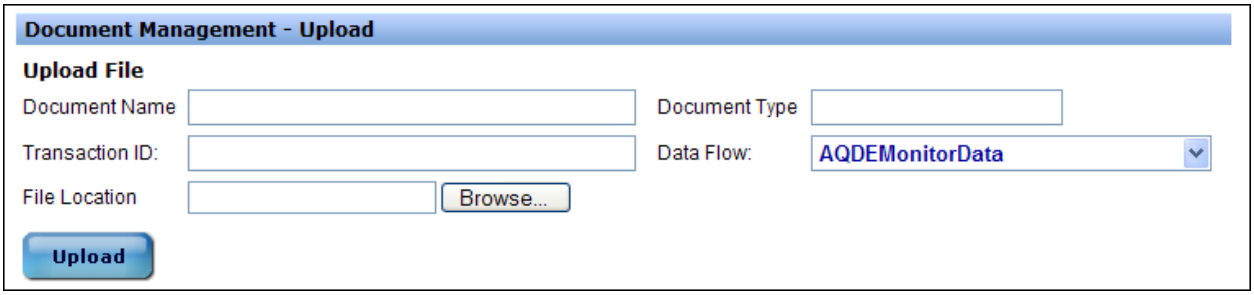


The Document search screen allows the Domain Administrators to search, upload, download, and view details for Node documents. To search for documents, the following search criteria are available:

- Document Name
- Transaction ID
- Operation Name
- Domain Name (Lists only Domains assigned to the Domain Admin)
- Start Date
- End Date

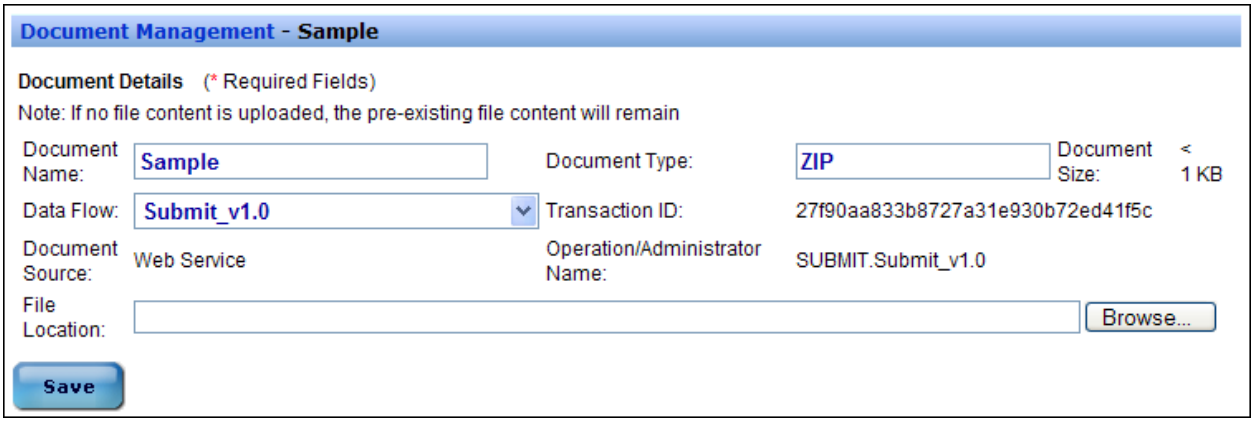
To upload a new document, click on the 

button. This will bring the user to a new page as shown below:



At the Upload screen, Domain Administrators can specify the Document Name, Document Type (xml, text, zip currently supported), Transaction ID, Data flow name (lists only those data flows to which the Domain Admin has rights) and upload the file location using the Browse button. If the Domain Admin is uploading a document to make it available for download by an external user, it is important to specify the transaction ID or document name: the ID or name will need to be supplied by the external user when they are ready to download the document from the EN-Node.

Edit File Properties: When the Domain Admin clicks on the view icon,  , on the Node Document search results grid, a new page is displayed, as shown below.



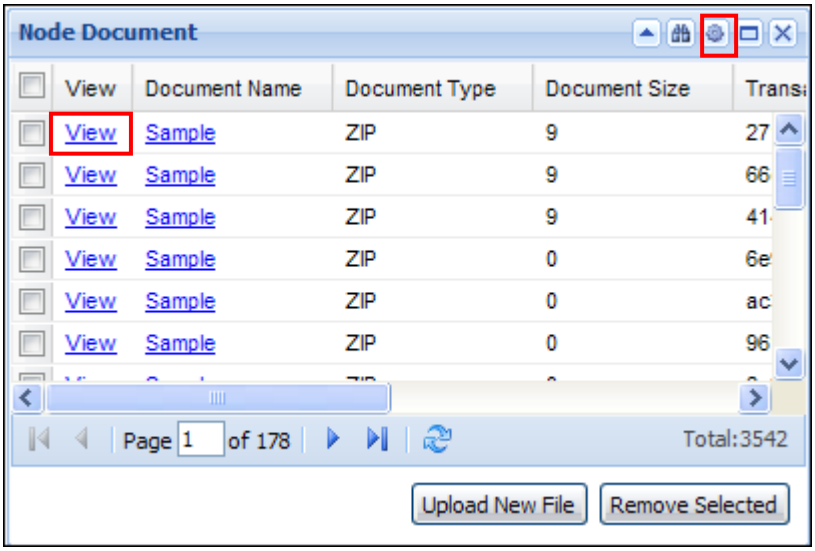
On this screen, the Admin can view/edit the following properties:

- Document Name
- Document Type
- Data Flow
- Transaction ID
- Document source
- Operation/Administrator Name
- File Location (this will replace the existing file)

All the changes made are saved by clicking the 

button.

The file can be downloaded by clicking the Download icon  on the Node Document screen. The document detail also can be accessed by clicking on the “Click to switch view model” icon and then by clicking on the View link as shown in the following figure:

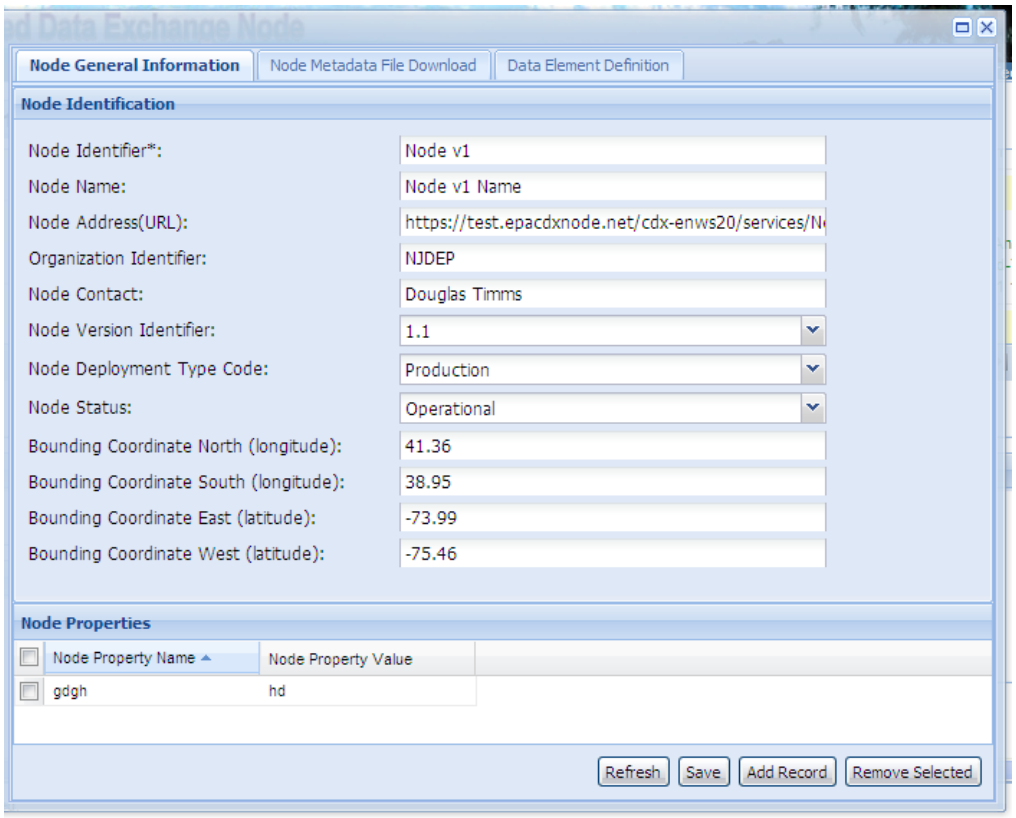


2.9 Node Registration

The Node Registration link allows the Node Administrator to define registration needed to support both the GetServices web method (used to register your Node's service at EPA's ENDS registry) and for generating DEDL XML files, which can also be registered at EPA's ENDS registry.

To clarify what these files are used for:

- **GetServices (also may be referred to as ENDS):** An XML file that describes your Node, Node location, listing of services, and (for Query and Solicit services) lists the service parameters
- **DEDL:** Allows you to further describe your service parameters, by specifying things such as parameter drop-down values, validation logic, human readable parameter labels, defines parameter data type, required indicator, multi-select indicator, etc.. This is especially useful if you want your services to appear in the EN Browser.



This screen provides 3 tabs:

- **Node General Information:** where you configure general information about your node.
  - **Note:** Important to note that the information provided here is only to populate the GetServices XML output, and is not used for other purposes.
  - **Note:** the information on this tab will change depending on whether you are on Tab 1.1 or Tab 2.0 in the Node Administration Console. You should configure this for both tabs if you support Node 1.1 and Node 2.0 servics.
- **Node File Download:** allows you to download an XML file that is the same XML file that would be returned if someone called a GetServices on your Node. This will dynamically join the general node information (from the first tab) with the node's service listing (which is pulled form the Domain/Operation configuration page).
  - **Note:** the file that is downloaded will vary depending on which tab (1.1 or 2.0) that you are on)

- **Data Element Definition:** allows you to define the DEDL information for your services. The following document is helpful in understanding what needs to be filled out here (especially if you are filling this out so it will be displayed in the EN Browser):
  http://www.enbrowser.net/enbrowser/ClientBin/HowToPublishData.pdf

  The PDF is a good source for describing each of the fields on this page. However, there are certain special fields that deserve additional description here:
  - **Dynamic Parameter Values:** you may have parameters that have a valid value list that is highly dynamic. (For example if you have a service that allows a user to filter data based on Project Name, and new projects are added weekly). In this case, it is too time consuming to manually maintain the valid values list. In these cases you can point the Node to a database table or view from which the list of valid values can be pulled. Then, this list can be regenerated and the resulting DEDL file can be submitted to EPA.

    To use dynamic parameter values set:
    - **Data Source Type:** set to "DBMS"
    - **Access Statement:** set to a comma separated string consisting of:
      1. COLUMN NAME of field to display to user
      2. COLUMN NAME of field to pass to Node
      3. TABLE or VIEW NAME
      4. (optional override) COLUMN NAME of field used to filter table or view
      5. (optional override) value of column from #4 above used to filter table or view

    Here is an example: `CHARACTERISTIC_NAME, CHARACTERISTIC_CD, VIEW_WQDE_RESULT, RESULT_STATUS_IDENTIFIER, Final`

    If any Data Element Definitions have these 2 conditions listed above met, then when the DEDL XML file is generated, it will retrieve the parameter lookup values from the database and refresh the DEDL XML file with the latest set of values. (The Access Statement values will then be removed from the DEDL XML prior to submission to EPA to protect against any potential SQL Injection attacks.

2.9.1 How to Register ENDS at EPA

- **Step 1: Supply General Node Information:** In Node Administration Console, click on the "Node Registration" link. Fill in general Node information on the 1st tab
- **Step 2: Specify which operations you wish to publish and provide operation parameter information:** Back on main Node Admin Console page, click through each of the operations by clicking the "Go to Operations" link for each domain, and for each operation click the edit button.
  - If you wish to publish the service, make sure that the "Include in publishing to ENDS" drop-down is set to "Yes", otherwise set to "No"
  - Specify the parameters for each Query or Solicit service
- **Step 3: Download ENDS Registration file:** Back on the main Node Admin Console, click on the "Node Registration" link. Then click on 2nd tab ("Node MetaData File Download") and click the "Download XML File" button. Save this XML file to your local hard drive.
- **Step 4: Submit ENDS XML file to EPA:** Using a Node Client, invoke a Submit to EPA's ENDS node by supplying the following information:
  - Endpoint:
    - Test: https://ends2.epacdxnode.net/Node2WS.svc
    - Production: https://ends2.epa.gov/Node2WS.svc
  - Submit Dataflow: ENDS_v20
  - Submit Flow Operation: Refresh

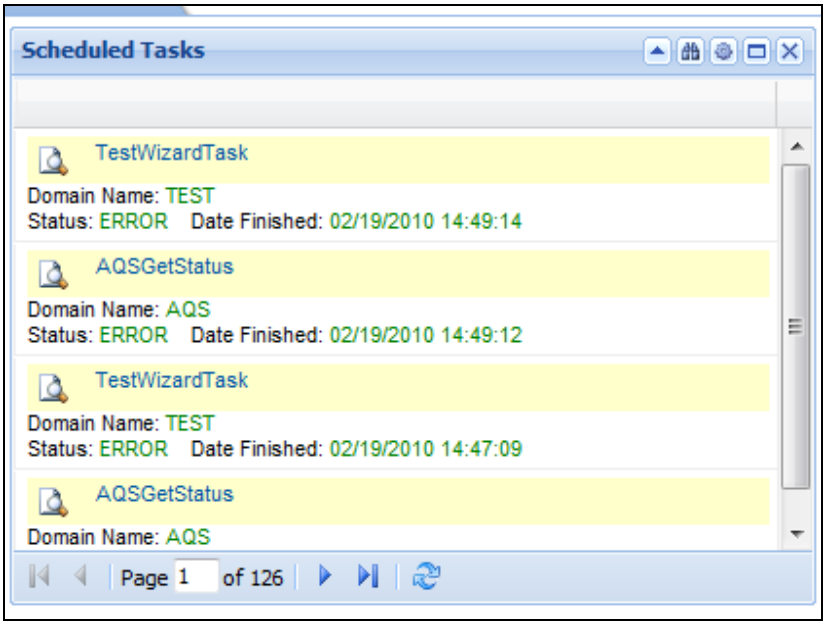
- **Step 5: Verify that the ENDS information is successfully updated in ENDS:** Log into EPA's ENDS application and verify that the ENDS information has been updated successfully:
    - Test Environment: https://ends2.epacdxnode.net
    - Production Environment: https://ends2.epa.gov

2.9.2 How to Register DEDL at EPA

Download the DEDL Plug-In located here (http://code.google.com/p/en-node2/downloads/detail?name=DEDL_PlugIn_StarterKit%28JAVA%29.zip&can=2&q=). Follow the read-me instructions contained in that download package for submitting DEDL information to EPA.

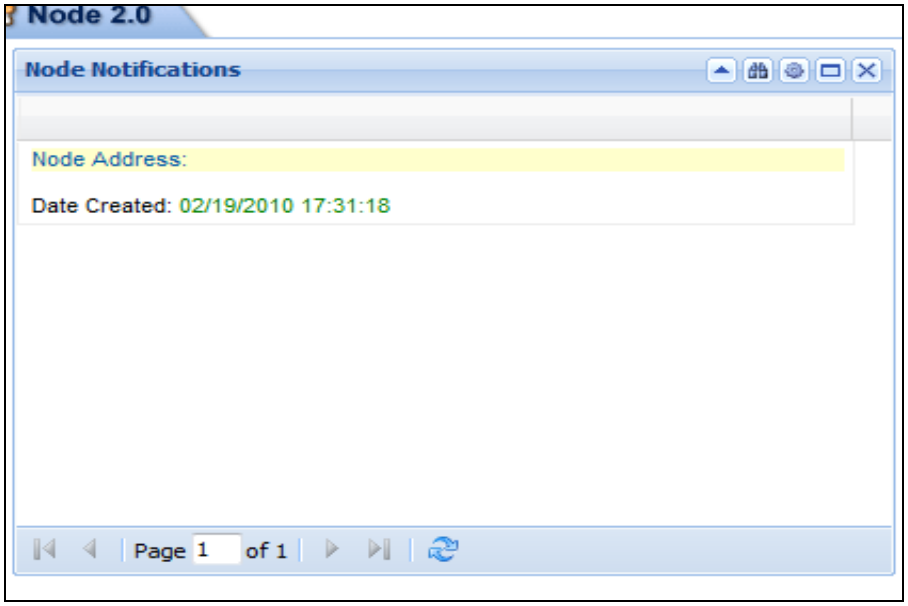
2.10 Scheduled Tasks

The Scheduled Tasks Web Part lists all instances that a Task has run on the Node.



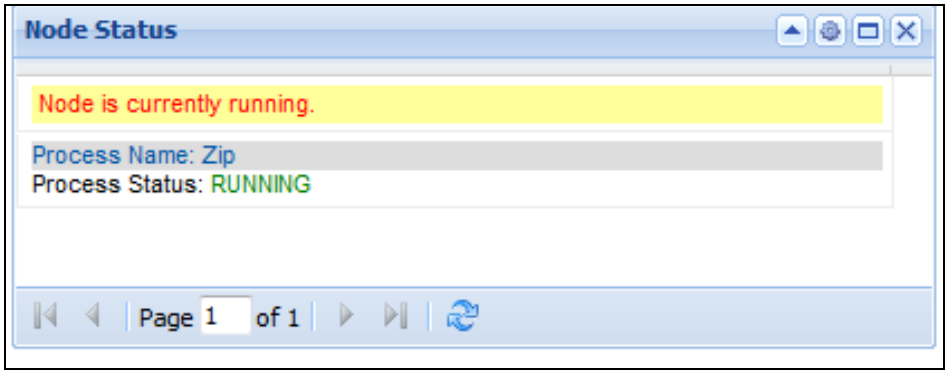
2.11 Node Notifications

The Node Notifications Web Part lists all notifications that have been sent to the node using the notificationURI feature. If a Web service (originating either from your Node or another Node) such as a Submit provides your node's web service endpoint in the notificationURI parameter, your node will be notified. This notification will appear in the Node Notifications Web Part.



2.12 Node Status

The Node Status Web Part shows the status of the Node and also displays all threads that are currently running in the background. This is useful to know if any long-running task or web service is currently running or has timed out or terminated unexpectedly.



3 Node Database Details

3.1 Data Dictionary

Table	Column	Data Type	Description
NODE_ACCOUNT_TYPE			This table stores the various possible account types that could be associated to the user.
	ACCOUNT_TYPE_ID	NUMBER(10)	Primary key for the table
	ACCOUNT_TYPE	VARCHAR2(20)	Account type: Node Admin, Domain Admin, NAAS managed user, locally managed user
	ACCOUNT_DESC	VARCHAR2(50)	Details about the account type
NODE_ACCOUNT_TYPE_XREF			This table cross references the user account type with the User ID and Domain ID.
	ACCOUNT_TYPE_XREF_ID	NUMBER(10)	Primary key for the table
	ACCOUNT_TYPE_ID	NUMBER(10)	Foreign key from NODE_ACCOUNT_TYPE table
	USER_ID	NUMBER(10)	Foreign key from SYS_USER_INFO table
	DOMAIN_ID	NUMBER(10)	Foreign key from NODE_DOMAIN table
NODE_DOMAIN			Stores details about the domain.
	DOMAIN_ID	NUMBER(10)	Unique Domain ID
	DOMAIN_NAME	VARCHAR2(50)	Domain name
	DOMAIN_DESC	VARCHAR2(100)	Description of the domain
	DOMAIN_STATUS_CD	VARCHAR2(10)	Indicates whether the Domain is Running or Stopped
	DOMAIN_STATUS_MSG	VARCHAR2(1000)	Provides a message to Node users regarding the current status of the Domain
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2 (50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2 (50)	Record updated by person's login name
NODE_DOMAIN_WEB_SERVICE_XREF			Acts as a XREF table between the Domain table and Web Service table. This table links

Table	Column	Data Type	Description
			the Web Services that are available to the Domain.
	DOMAIN_ID	NUMBER(10)	Foreign key from Domain table (Part of Composite Primary key)
	WEB_SERVICE_ID	NUMBER(10)	Foreign key from NODE_WEB_SERVICE table (Part of Composite Primary key)
NODE_FILE_CABIN			This table stores files that are used in Node operations like the Submit and Solicit methods.
	FILE_ID	NUMBER(10)	Unique file ID (Primary key for the table)
	TRANS_ID	VARCHAR2(50)	Node transaction ID which uses files
	FILE_NAME	VARCHAR2(200)	File name if it's available
	FILE_TYPE	VARCHAR2(50)	File type if it's available
	STATUS_CD	VARCHAR2(10)	This is for the Solicit method when the URL address is passed to Node. Possible values are R (ready to submit) and S (Submitted)
	DATAFLOW_NAME	VARCHAR2(50)	Indicates the data flow either supplied as a parameter during a Submit, or supplied by the Node administrator during file upload.
	SUBMIT_URL	VARCHAR2(100)	The URL to which documents will be submitted. This is for the Solicit method when the URL address is passed to Node.
	SUBMIT_TOKEN	VARCHAR2(200)	The security token that the application will use to invoke the Submit method. This is for the Solicit method when the URL address is passed to Node.
	SUBMIT_DTTM	DATE	Date time that the document is submitted to URL. This is for the Solicit method when the URL address is passed to Node.
	FILE_CONTENT	BLOB	The content of file in BLOB format.
	FILE_SIZE	NUMBER(10)	Stores the size of the document
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2 (50)	Record updated by person's login name
	DOCUMENT_ID	VARCHAR2 (50)	Reserve for future using
NODE_OPERATION			Stores information related to the operations
	OPERATION_ID	NUMBER(10)	Unique operation ID auto created by the system.
	DOMAIN_ID	NUMBER(10)	Foreign key from Domain table that identifies the unique domain for which this operation is associated.
	WEB_SERVICE_ID	NUMBER(10)	Foreign key from Web_Service table that identifies the unique web service handler for which this operation is associated.
	OPERATION _NAME	VARCHAR2(50)	Name of the operation
	OPERATION_DESC	VARCHAR2(255)	Description of the operation
	OPERATION_TYPE	VARCHAR2(25)	Indicates the operation type as either “Scheduled Task” or “Web Service”
	OPERATION_CONFIG	XMLTYPE	Stores details about operations
	OPERATION_STATUS_CD	VARCHAR2(20)	Indicates the status of the operation.
	OPERATION_STATUS_MSG	VARCHAR2(1000)	Stores the status message provided by the Domain Administrator for the operation.

Table	Column	Data Type	Description
	VERSION_NO	VARCHAR2(10)	Stores the version of Node (VER_11 or VER_20)
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2(50)	Record updated by person's login name
	OPERATION_CONFIG_CLOB	CLOB	Stores details about operations of CLOB version
	PUBLISH_IND	VARCHAR2(1)	Indicate if the web service is published to public
NODE_OPERATION_LOG			Logs in the details of the web service operations.
	OPERATION_LOG_ID	NUMBER(10)	Primary key for the table auto created by the system
	TRANS_ID	VARCHAR2(50)	Unique transaction ID generated for each service request
	OPERATION_ID	NUMBER(10)	Foreign key from Operation table that identifies the unique operation for which this operation log applies (if the system could resolve one based on the service request).
	USER_NAME	VARCHAR2(50)	Identifies the USER_NAME either directly supplied (for Authentication requests), or obtained via security token and previous operation log.
	REQUESTOR_IP	VARCHAR2(30)	IP address of the Node requestor.
	SUPPLIED_TRANS_ID	VARCHAR2(50)	Transaction ID supplied as an input parameter for Submit, GetStatus, and Download service requests.
	TOKEN	VARCHAR2(200)	Security token that is used to make Node request.
	NODE_ADDRESS	VARCHAR2(100)	Nodeaddress parameter supplied as part of Notify service request
	RETURN_URL	VARCHAR2(100)	returnURL parameter supplied as part of Solicit service request
	SERVICE_TYPE	VARCHAR2(50)	ServiceType parameter supplied as part of GetServices service request
	START_DTTM	DATE	Date time a task started (for scheduled operations) or the datetime the service request was made.
	END_DTTM	DATE	Date time a task completed (for scheduled operations) or the datetime the service request response was made.
	HOST_NAME	VARCHAR2(40)	Server name which ran the task (for scheduled operations).
	CREATED_DTTM	DATE	Record created by person's login name
	CREATED_BY	VARCHAR2(20)	Record created date time
	UPDATED_DTTM	DATE	Record updated by person's login name
	UPDATED_BY	VARCHAR2(20)	Record updated date time
NODE_OPERATION_LOG_PARAMETER			Stores each parameter supplied as part of the parameter array for Query and Solicit service requests. Allows for relating asynchronous requests having different tokens.
	OPERATION_LOG_ID	NUMBER(10)	Unique operation log that helps form composite primary key for the table

Table	Column	Data Type	Description
	PARAMETER_NAME	VARCHAR2(50)	Parameter name supplied as part of service request that helps form composite primary key for the table.
	PARAMETER_VALUE	VARCHAR2(4000)	Stores the value for the supplied parameter
NODE_OPERATION_LOG_STATUS			Stores details about the status of the operation logs
	OPERATION_LOG_STATUS_ID	NUMBER(10)	Primary key for the table
	OPERATION_LOG_ID	NUMBER(10)	Foreign key from Node_Operation_Log table
	STATUS_CD	VARCHAR2(20)	Status of the operation
	MESSAGE	VARCHAR2(4000)	Message for the operation
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
NODE_OPERATION_MANAGER			Stores details information of the operation manager
	SUBMIT_ID	NUMBER(10)	Primary key for the table
	OPERATION_NAME	VARCHAR2 (100)	Operation name
	STATUS_CD	VARCHAR2(25)	Operation status
	SUBMITTED_DTTM	DATE	Submit date
	SUBMITTED_URL	VARCHAR2(100)	Submit target end point
	VERSION_NO	VARCHAR2(10)	Node version number
	TRANS_ID	VARCHAR2(100)	Node transaction ID
	SUPPLIED_TRANS_ID	VARCHAR2(100)	EPA transaction ID
	FILE_CONTENT	BLOB	Submit file content
	DATA_FLOW	VARCHAR2(50)	Data flow name
NODE_PROCESS_MONITOR			This table stores background process information.
	PID	INTEGER	Primary key for the table
	TRANSACTION_ID	VARCHAR(100)	Unique transaction ID generated for each service request
	OPERATION_ID	INTEGER,	Foreign key from Operation table that identifies the unique operation for which this operation log applies (if the system could resolve one based on the service request).
	PROCESS_NAME	VARCHAR(300)	Name of Process under threading condition
	PROCESS_STATUS	VARCHAR(100)	Status of Process under threading condition
	UPDATED_DTTM	DATETIME,	Record updated date time
	UPDATED_BY	VARCHAR(100)	Record updated by person's login name
	NODE_ADDRESS	VARCHAR(300)	Record the external URL which is calling by EN-Node
NODE_USER_OPERATION_XREF			Table to manage privileges of the user with respect to the operation
	USER_ID	NUMBER(10)	Foreign key from SYS_USER_INFO table (Part of Composite Primary Key)
	OPERATION_ID	NUMBER(10)	Foreign key from NODE_OPERATION table (Part of Composite Primary Key)
NODE_WEB_SERVICE			Stores details about the available Web Services
	WEB_SERVICE_ID	NUMBER(10)	Primary key for the table
	WEB_SERVICE_NAME	VARCHAR2(50)	Name of the Web Service

Table	Column	Data Type	Description
	WEB_SERVICE_DESC	VARCHAR2(50)	Description of the Web Service
SYS_ADDRESS			Stores info about address for the user accounts
	ADDRESS_ID	NUMBER(10)	Primary key for the table
	ADDRESS	VARCHAR2(100)	User address
	SUPPLEMENTAL_ADDRESS	VARCHAR2(100)	Supplemental address
	LOCALITY_NAME	VARCHAR2(100)	Address locality name
	STATE_CD	CHAR(2)	Address state code
	ZIP_CODE	VARCHAR2(15)	Address zip code
	COUNTRY_CD	VARCHAR2(25)	Address country code
	STATUS_CD	CHAR(1)	Active/Inactive status
	ADDRESS_DESC	VARCHAR2(100)	Description of the address
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2(50)	Record updated by person's login name
SYS_CONFIG			This table stores node system configuration files. Those files are used by application and will effect how the system runs.
	CONFIG_ID	NUMBER(10)	Unique configuration ID
	CONFIG_NAME	VARCHAR2(300)	Configuration file name
	CONFIG_XML	XMLTYPE	Content of configuration file
	CONFIG_TYPE_CD	VARCHAR2(25)	Indicator to differentiate the type of configuration file
	STATUS_CD	CHAR(1)	Configuration file status. Possible values are A (Active) or I (Inactive). This must be set to active “A” for the system to read this file.
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2 (50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2 (50)	Record updated by person's login name
	CONFIG_CLOB	VARCHAR2 (50)	Clob version of content of configuration file
SYS_EMAIL			Stores info about email for the user accounts
	EMAIL_ID	NUMBER(10)	Unique ID for the table
	EMAIL_ADDRESS	VARCHAR2(100)	Email address (ID) for the user account
	EMAIL_TYPE	VARCHAR2(25)	For differentiating Emails when User has more than one Email ID
	STATUS_CD	CHAR(1)	Active/Inactive status
	CREATED_DTTM	DATE	Record created by person's login name
	CREATED_BY	VARCHAR2(50)	Record created date time
	UPDATED_DTTM	DATE	Record updated by person's login name
	UPDATED_BY	VARCHAR2(50)	Record updated date time
SYS_SEQUENCE_NO			Table used to manage automatic numbering of the primary keys for other tables.
	SEQUENCE_ID	NUMBER(10)	Primary key for the table
	TABLE_NAME	VARCHAR2(30)	Table for which the sequence number is created
	COLUMN_NAME	VARCHAR2(30)	Column for which the sequence number is created

Table	Column	Data Type	Description
	MIN_NUMBER	NUMBER(5)	Minimum number for the sequence
	MAX_NUMBER	NUMBER(10)	Max number limit for the sequence
	LAST_USED_NUMBER	NUMBER(10)	Displays the last used sequence number for the column in that specific table
	STATUS_CD	CHAR(1)	Active/Inactive status
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2(50)	Record updated by person's login name
SYS_USER_ADDRESS			Maps user to their Address via SYS_ADDRESS table
	ADDRESS_ID	NUMBER(10)	Unique Address ID (Part of Composite Primary Key)
	USER_ID	NUMBER(10)	Unique User ID from SYS_USER_INFO table (Part of Composite Primary Key)
	ADDRESS_TYPE_CD	VARCHAR2(25)	Specifies the type of address
SYS_USER_EMAIL			Maps users to their Email ID via SYS_EMAIL table
	EMAIL_ID	NUMBER(10)	Unique Email ID (Part of Composite Primary Key)
	USER_ID	NUMBER(10)	Unique User ID from SYS_USER_INFO table (Part of Composite Primary Key)
SYS _USER_INFO			This table stores user account information for using Node administration tool. System will use information from this table to control user login Node administration and security rights of using Node administration tool.
	USER_ID	NUMBER(10)	Unique user ID assigned
	LAST_NAME	VARCHAR2(60)	Node user’s last name
	FIRST_NAME	VARCHAR2(60)	Node user’s first name
	MIDDLE_INITIAL	CHAR(1)	Node user’s middle initial
	LOGIN_NAME	VARCHAR2(50)	Login name for the user
	LOGIN_PASSWORD	VARCHAR2(100)	Login password for administration user account
	USER_STATUS_CD	VARCHAR2(10)	User account status code. This controls if the user can login to the system. Possible values are A (Active) or I (Inactive).
	LAST_4_SSN	VARCHAR2(4)	Last 4 digits of admin’s social security number
	CHANGE_PWD_FLAG	VARCHAR2(1)	A flag which indicates if password need to be changed when user login in. Possible values are Y or N. Flag will be set to Y when user's password has been created for 1st time. After the password has been reset when the user logs in, flag will be set to N.
	PHONE_NUMBER	VARCHAR2(15)	User contact phone number
	COMMENTS	VARCHAR2(255)	Comment on user account
	CREATED_DTTM	DATE	Record created date time
	CREATED_BY	VARCHAR2(50)	Record created by person's login name
	UPDATED_DTTM	DATE	Record updated date time
	UPDATED_BY	VARCHAR2(50)	Record updated by person's login name
	UPDATED_DTTM	DATE	Record updated date time

3.2 Database Diagram