EN-NODE USER'S GUIDE

Version: 1.0

September 26, 2008

Purchase Order #: (if applicable)

enfoTech

11 Princess Road, Unit A
Lawrenceville, New Jersey 08648
www.enfotech.com

Restriction on Use and Disclosure of Document Information

This document includes data that should not be disclosed outside the business entity for which it was intended, indicated as the recipient on this title page. The entire document is copyrighted by enfoTech and is protected under the US copyright law and international treaties. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without express written permission from enfoTech & Consulting Inc.

Copyright © 2001 – 2008 by enfoTech & Consulting Inc. All Rights Reserved.

Revision History

Version	Date	Created By	Reviewed By	Description
1.0	09/26/2008	Charlie Tsai		
	09/10/2010	Charlie Zhang		

Tables of Contents

1 Introduction

1.1 Document Purpose

The purpose of this document is to provide instructions to Node and data flow Users on the use of the EN-Node Client application.

1.2 Overview of Node Components

EN-Node is a Web Services-based data exchange portal. Built using Web services, it allows other trading partners to access your environmental data in a secure and consistent manner. In addition, the EN-Node provides capabilities for submitting data to other Network Nodes, including EPA's Node and the Nodes of other states.

The Node consists of four sub-applications:

- **Node.WebServices:** An engine that controls the logic for responding to Web Service requests on the Node, providing the nine web services outlined in the Node 1.1/2.0 Specifications. When responding to a Web service request, Node.WebServices will execute .NET logic plugged in for a particular data flow.
- **Node.Task:** Provides the capability to execute tasks on a scheduled basis, which allows the state 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 in .NET and are configured by a Domain Admin for a particular data flow.
- **Node.Administration:** A Web interface that allows Node and Domain administrators to configure the Node and manage data flows. The Node.Administration application serves as an interface to configure the Node.WebServices, Node.Task, and Node.Client applications.
- **Node.Client:** A simple Web interface that allows individuals to invoke Web Services on any Node, including the e-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.

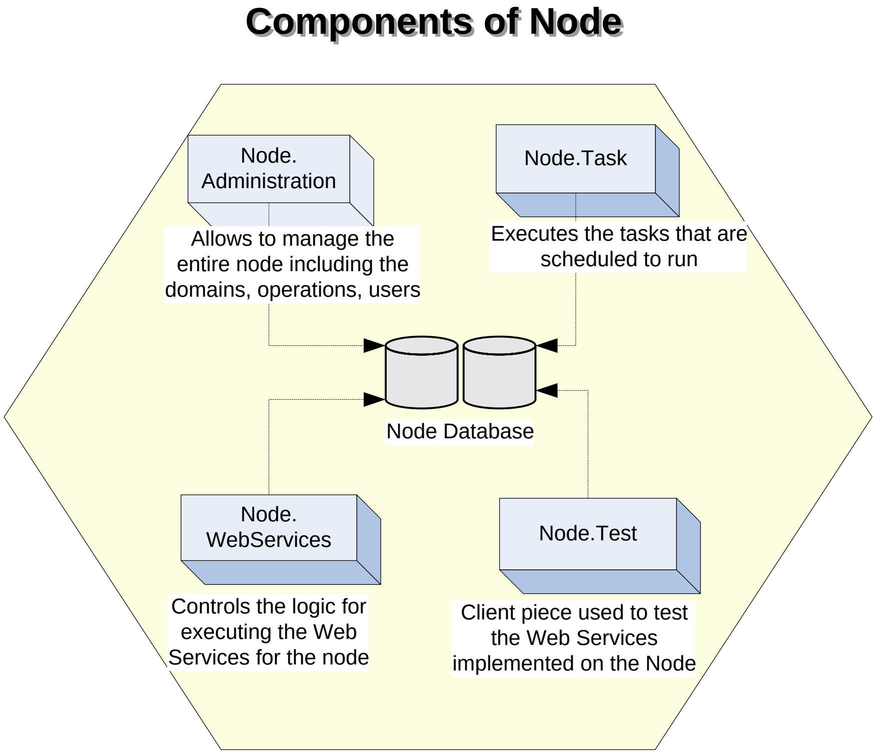

Components of Node
Node. Administration
Allows to manage the entire node including the domains, operations, users
Node.Task
Executes the tasks that are scheduled to run
Node Database
Node. WebServices
Controls the logic for executing the Web Services for the node
Node.Test
Client piece used to test the Web Services implemented on the Node

Figure 1-1 Components of Node

1.3 Available Web Services

The Node comes preconfigured with the 9 standard Web services defined in the Node 1.1/2.0 specifications. These 9 Web services include[1]:

- **Authenticate:** The Authenticate method authenticates a user using a supplied credential.
- **Submit:** The Submit method provides a generic way of sending one or more payloads to a service provider.
- **GetStatus:** GetStatus is a method for transaction tracking. Once submitted, a transaction enters into different processing stages. The GetStatus method offers the client a way of querying the current state of the transaction.
- **Query:** The Query method is a function in the Database interface. The 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 the requested operation in the background or sometimes offline. It is designed especially for queries that may take a long time.
- **Notify:** The Notify method has three (3) intended uses: document notification, event notification, and status notification
- **Download:** The Download method is a function in the Retrieve Interface. After being notified by a submitter, a Node invokes the Download method to retrieve available documents
- **NodePing:** The NodePing method is a function in the Admin interface. It is a utility method for determining whether a Node is accessible.
- **GetServices:** The GetServices method is a function in the Admin interface. It allows requesters to query services provided by a Network Node.

1.4 Overview of How EN-Node Works

EN-Node provides each of the Web services listed in Section 1.4 through the use of “Handlers”. The handler determines the logic/approach by which a Web Service request will be handled. 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.

This is illustrated in Figure 1-2 below:

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

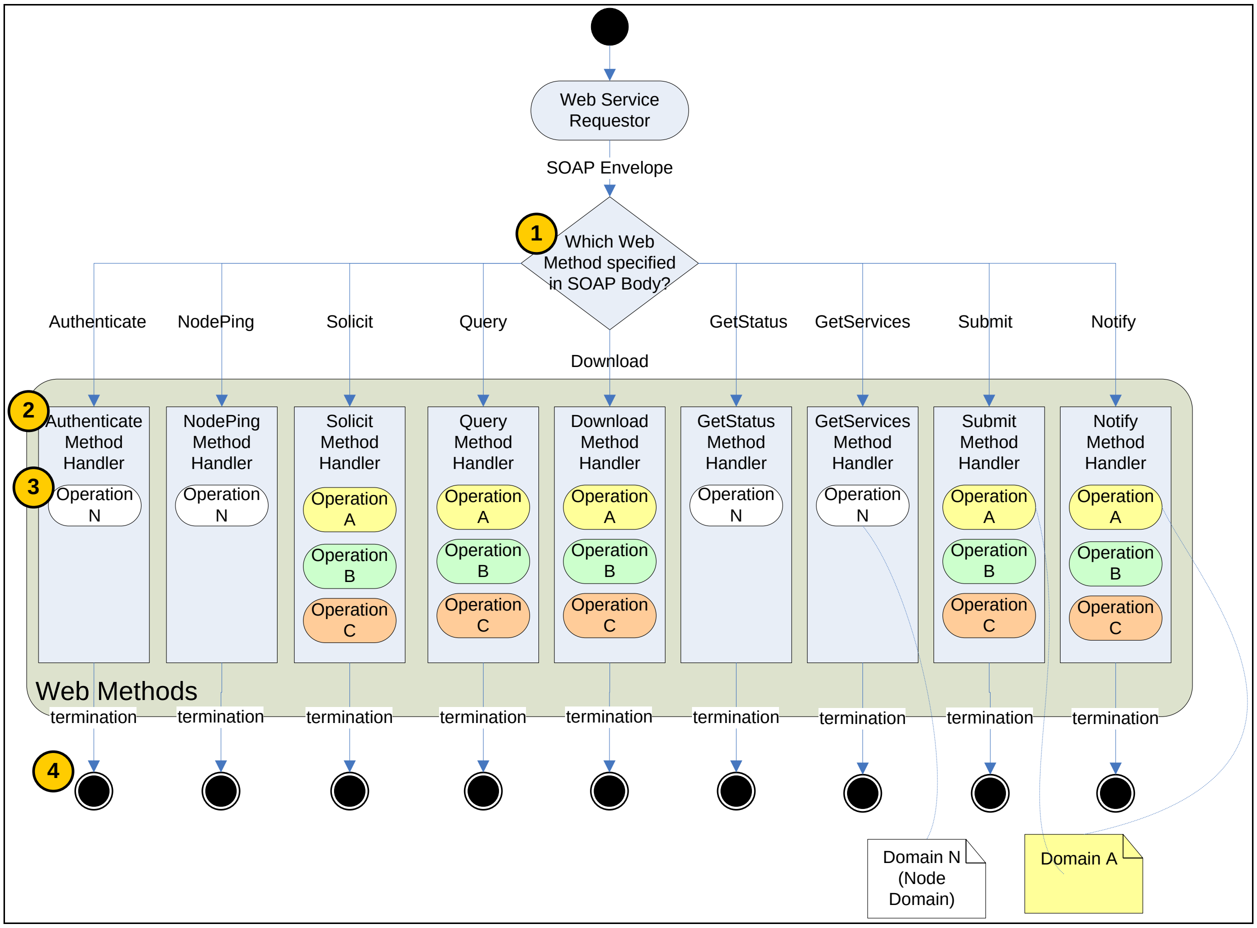


Figure 1-2: EN-Node Overview

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 e-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 in order any registered Pre-Processes for the Operation.
- EN-Node then searches for and 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 searches for and executes in order any Post-Processes registered for the Operation.

2 Using the Node Client

The Node Client is the main interface for a user to access the call Web Services supplied by a target Node. It can be run by loading the following page "http://<<InstallLocation>>/Node.Client/Page/Entry/FrameSet.jsp". After entering the Node Client there is a list of the nine Web Services that can be performed on the left hand side of the screen. To invoke any of the services click the corresponding button in this section of the screen:

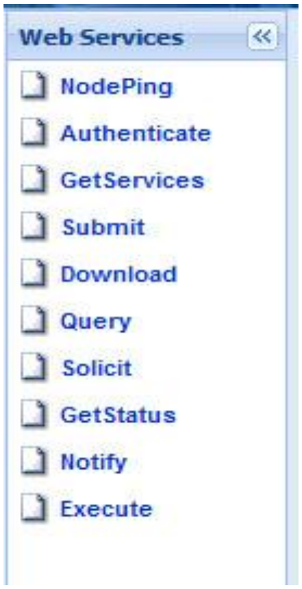


Figure 2-1: Node Client Navigation Panel

Due to fact that eNode200 provide two version of the node service, you can switch the version by click on the tab as screen shown below

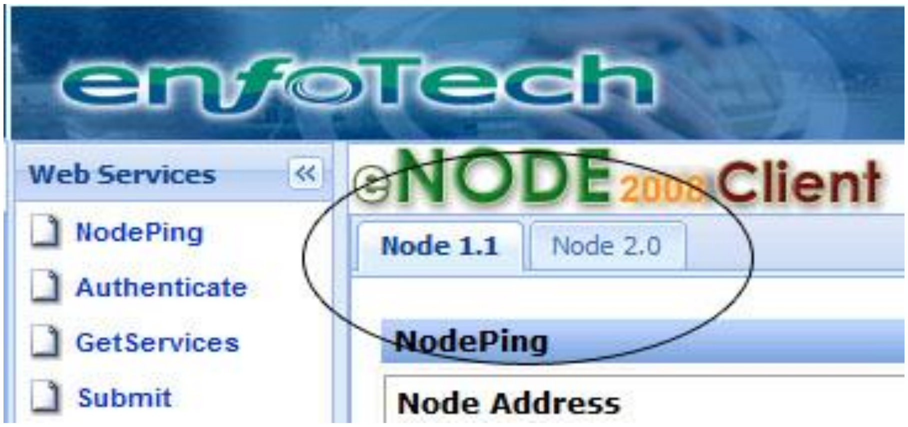


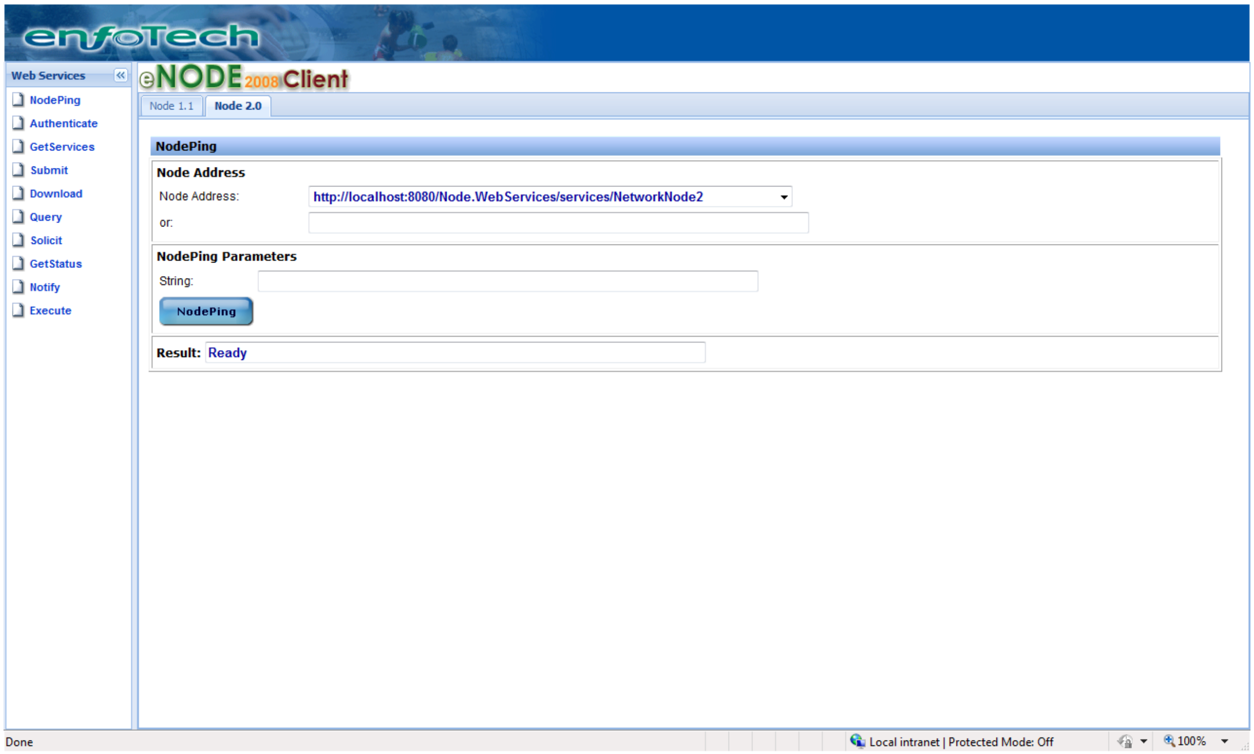
Figure 2-2: Node Version Selection Tab

2.1 NodePing

When first entering the Node Client, the default screen that will appear is the Node Ping screen. This Service allows you to determine if a particular Node is accessible. To call the service, the following information is needed:

- **Node Address**: This is the Node URL of the desired Node. A selection can either be made from the drop-down list of pre-selected Nodes or be entered in the free text field.
- **Node Ping Input Parameter**: Enter any string. This string will then be returned if the Node is functioning.

After entering the required information click the “Node Ping” Button. If the text entered as the input parameter is returned in the “Node Ping Result” box, then the Node that was called is functioning, as shown below:



NodePing

Node Address

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

or:

NodePing Parameters

String:

Result: Ready

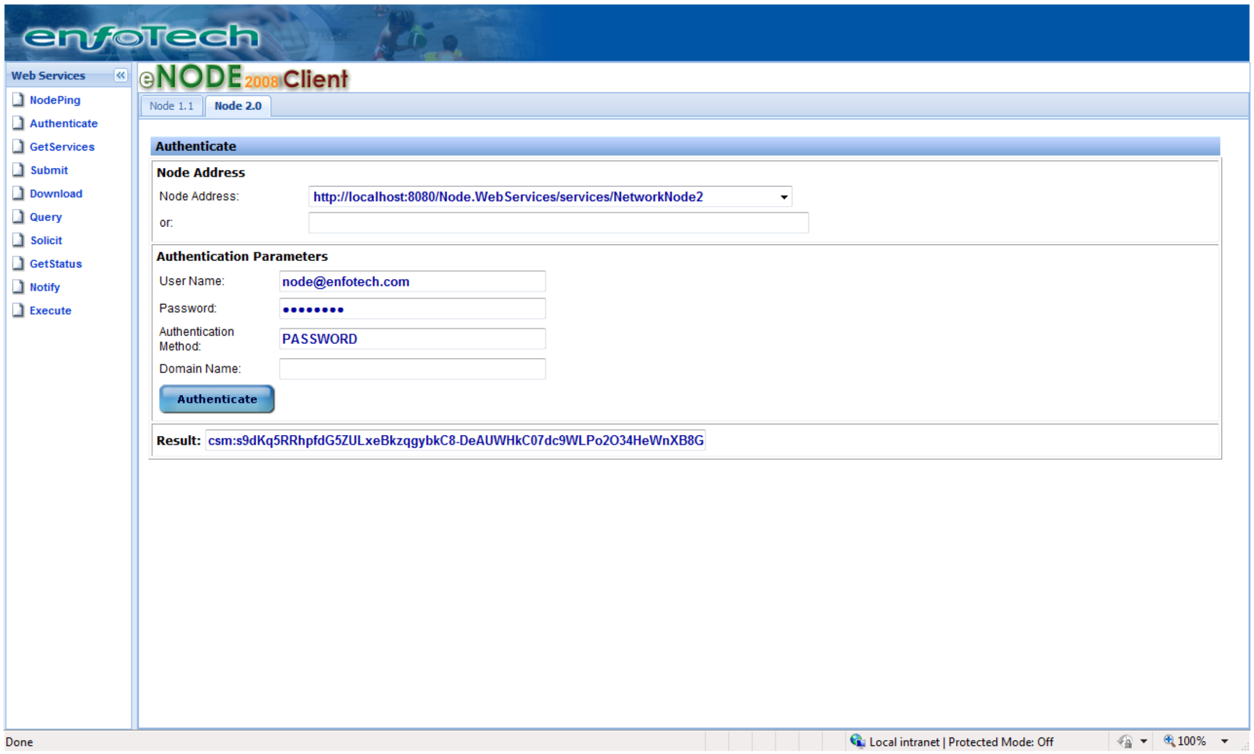
Figure 2-3: Node Ping Web Service Call Screen

2.2 Authenticate

Other than the Node Ping Web Service call, a user must first obtain a security token before making the service request as proof of identity. A security token is an opaque string is only meaningful to the issuer or its trusted peers. To obtain a security token the Authenticate Web Service must be invoked. To call the service, the following user information is needed (typically a NAAS account is required):

- **Node Address**: This is the Node URL of the desired Node. A selection can either be made from the drop-down list of pre-selected Nodes or be entered in the free text field.
- **User ID**: The user's User ID for invoking Web Services (typically the user ID of a NAAS account)
- **Password**: The password for the User ID supplied
- **Authentication Method**: Use the default text "Password"
- **Domain Name**: New Parameter specified in 2.0 Node Spec which is optional parameter

After entering the required information click the "Authenticate" Button. If the user account information is correct the security token will be returned in the "Authenticate Result" box, as shown below:



Authenticate

Node Address

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

or:

Authentication Parameters

User Name: node@enfotech.com

Password: ••••••••

Authentication Method: PASSWORD

Domain Name:

Result: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc9WLPo2O34HeWnXB8G

Figure 2-2: Authenticate Web Service Call Screen

After obtaining a security token the any of the remaining seven Web Service types can be invoked. **Note**: The target Node may need to provide rights for each particular Service offered before they can first be invoked. If this is the case contact the Node Administrator of the target Node.

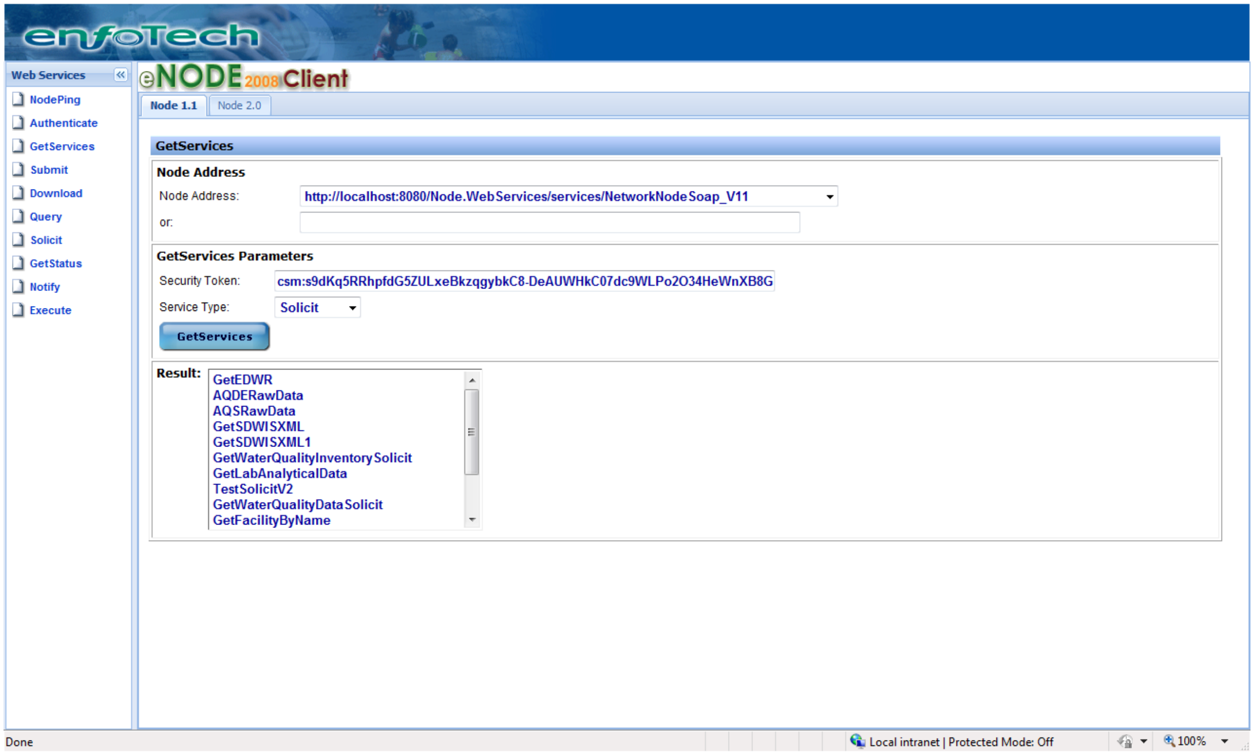
2.3 Get Services

After receiving a security token go to the Get Services screen in order to query the services that are being made available by the desired Node. The type of services that can be queried includes (but is not limited to):

- Interfaces: The Web service interfaces supported by the Node
- Query: Predefined information requests that can be used in the Query method
- Solicit: Predefined information requests that can be used in the Solicit method
- Execute: Predefined procedures that can be used in the Execute method

To call the service, the following information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Service Type**: The type of services that will be returned from the query (this can either be chosen from a drop-down list or entered as free text).



GetServices

Node Address

Node Address: http://localhost:8080/Node.WebServices/services/NetworkNodeSoap_V11

or:

GetServices Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc9WLPo2O34HeWnXB8G

Service Type: Solicit

Result:

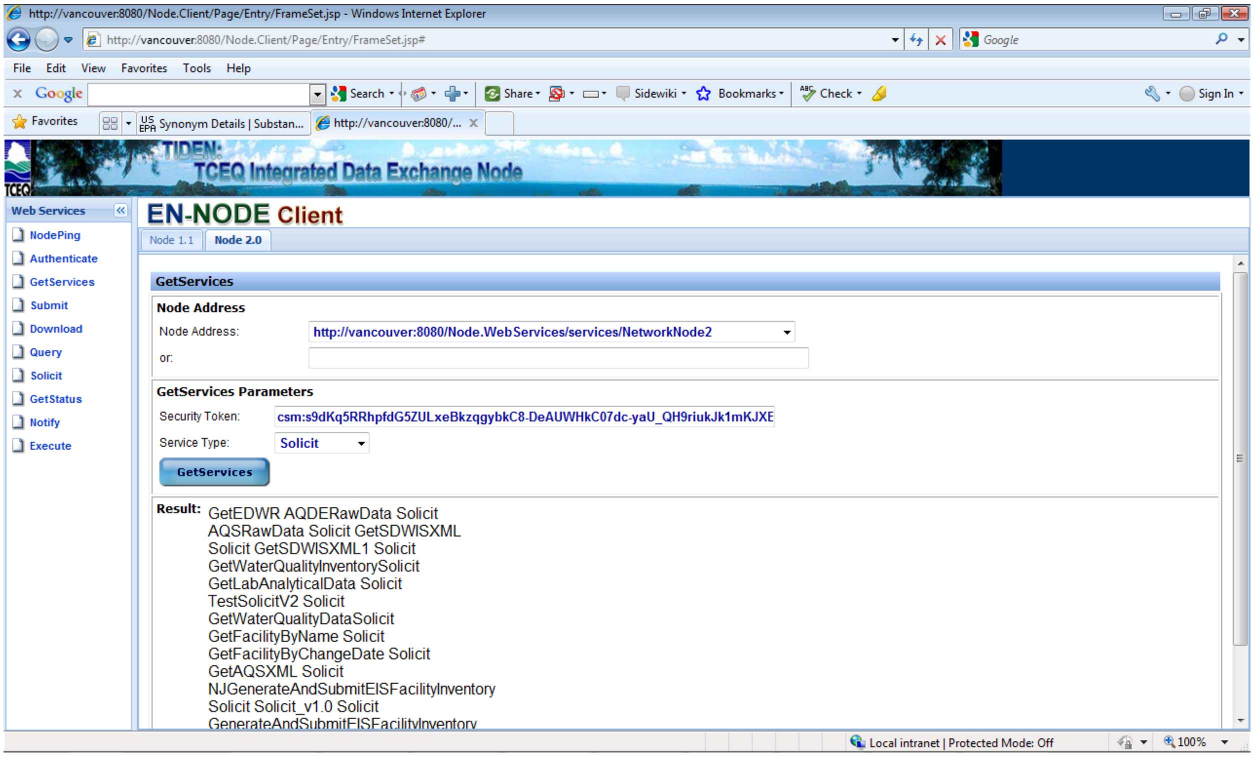
```
GetEDWR
AQDERawData
AQSRawData
GetSDWISXML
GetSDWISXML1
GetWaterQualityInventorySolicit
GetLabAnalyticalData
TestSolicitV2
GetWaterQualityDataSolicit
GetFacilityByName
```

Figure 2-3: Get Services Web Service Call Screen

For 2.0 Node, the type of services that can be queried includes

- AllServices: A list of all service categories supported by the GetService method.
- Query: Predefined data Service that can be used in the Query method
- Solicit: Predefined data service that can be used in the Solicit method
- Execute: Predefined data service that can be used in the Execute method.

After entering the required information click the “Get Services” Button. A query will then be performed and all services matching the Service Type supplied will be returned in the “Get Services Result” box, as shown below:


http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer
http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#
File
Edit
View
Favorites
Tools
Help
Google
Search
Share
Sidewiki
Bookmarks
Check
Sign In
Favorites
Synonym Details | Substan...
http://vancouver:8080/...
TIDEN:
TCEQ Integrated Data Exchange Node
TCEQ
Web Services
NodePing
Authenticate
GetServices
Submit
Download
Query
Solicit
GetStatus
Notify
Execute
EN-NODE Client
Node 1.1
Node 2.0
GetServices
Local intranet | Protected Mode: Off
100%

Node Address

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

or:

GetServices Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Service Type: Solicit

GetServices

Result:

```
GetEDWR AQDERawData Solicit
AQSRawData Solicit GetSDWISXML
Solicit GetSDWISXML1 Solicit
GetWaterQualityInventorySolicit
GetLabAnalyticalData Solicit
TestSolicitV2 Solicit
GetWaterQualityDataSolicit
GetFacilityByName Solicit
GetFacilityByChangeDate Solicit
GetAQSXML Solicit
NJGenerateAndSubmitEISFacilityInventory
Solicit Solicit_v1.0 Solicit
GenerateAndSubmitEISFacilityInventory
```

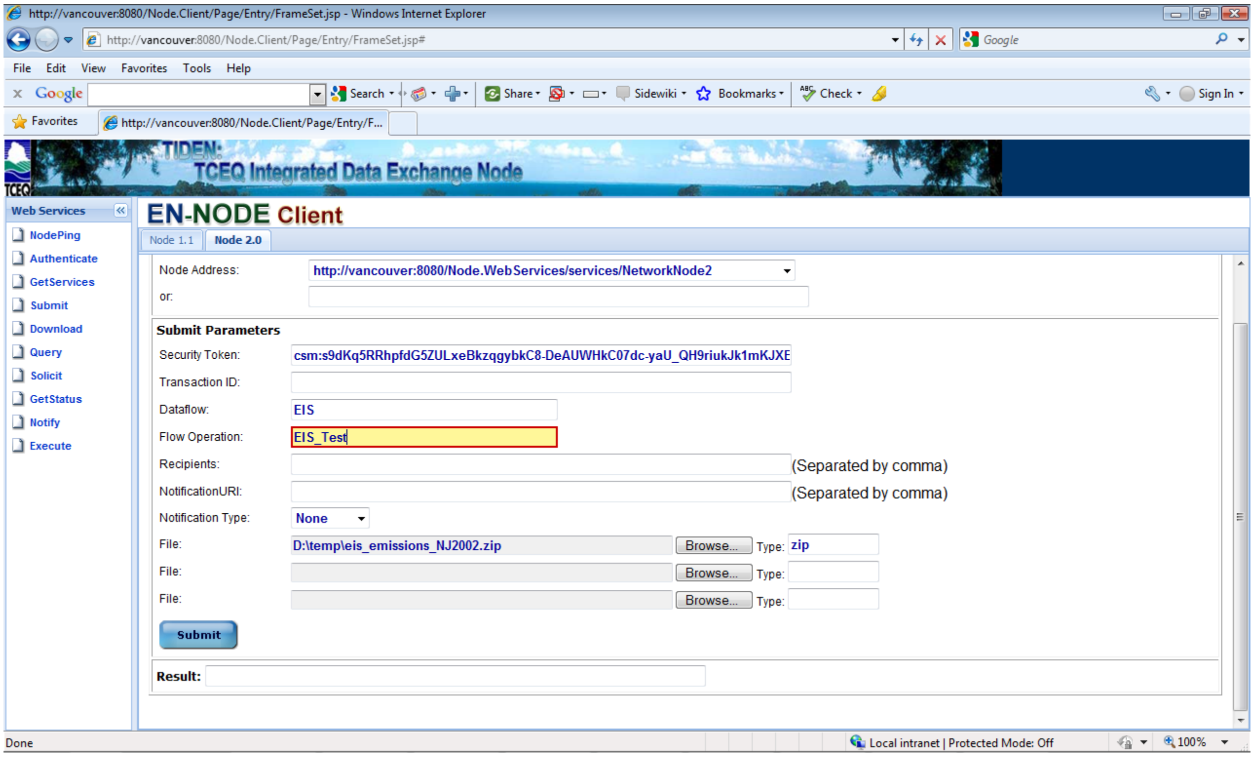
Figure 2-5: Node 2.0 Get Services Web Service Call Screen

2.4 Submit

After receiving a security token go to the Submit screen in order to send a payload to the target Node. This is a generic way to send one or more payloads to a service provider. To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Transaction ID (optional)**: A transaction ID for the submission if the operation is a result of an asynchronous operation. It should be the transaction Id associated with a previous solicited operation (See the Solicit method) if any. It should be empty if the Submit operation is independent.
- **Dataflow**: The name of the target dataflow. A dataflow is a logical collection of certain kinds of documents, understandable to the sender and the ultimate receiver. Therefore, a dataflow can also be understood as a tag of the ultimate receiver of the payload.(For Node default submit, here is "NODE")
- **Documents**: The user can navigate to the files they wish to upload by clicking the "Browse" button.
- **Document Type**: Next to each file specify the type of file being submitted (txt, xml, zip, other)
- **Flow Operation**: Defined by 2.0 Node Spec., It indicate the specific processing for the document.(For Node default submit, here is "default")
- **Recipient**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction ID of the submission.
- **Notification URI**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction id, status code, and statusDetail of the submission. If it contains a Node URI, the Notify method will be called on the URI, including the transaction ID and transaction status information

After entering the required information click the "Submit" Button. The files will then be submitted to the target Node and a transaction ID will be returned in the "Submit Result" box, as shown below:


http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer
http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#
File Edit View Favorites Tools Help
Google
Search
Share
Sidewiki
Bookmarks
Check
Sign In
Favorites
http://vancouver:8080/Node.Client/Page/Entry/F...
TIDEN:
TCEQ Integrated Data Exchange Node
TCEQ
Web Services
NodePing
Authenticate
GetServices
Submit
Download
Query
Solicit
GetStatus
Notify
Execute
EN-NODE Client
Node 1.1
Node 2.0
Browse...
Browse...
Browse...
Submit
Done
Local intranet | Protected Mode: Off
100%

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

or:

Submit Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Transaction ID:

Dataflow: EIS

Flow Operation: EIS_Test

Recipients: (Separated by comma)

NotificationURI: (Separated by comma)

Notification Type: None

File: D:\temp\eis_emissions_NJ2002.zip Type: zip

File: Type:

File: Type:

Result:

Figure 2-6: Submit Web Service Call Screen

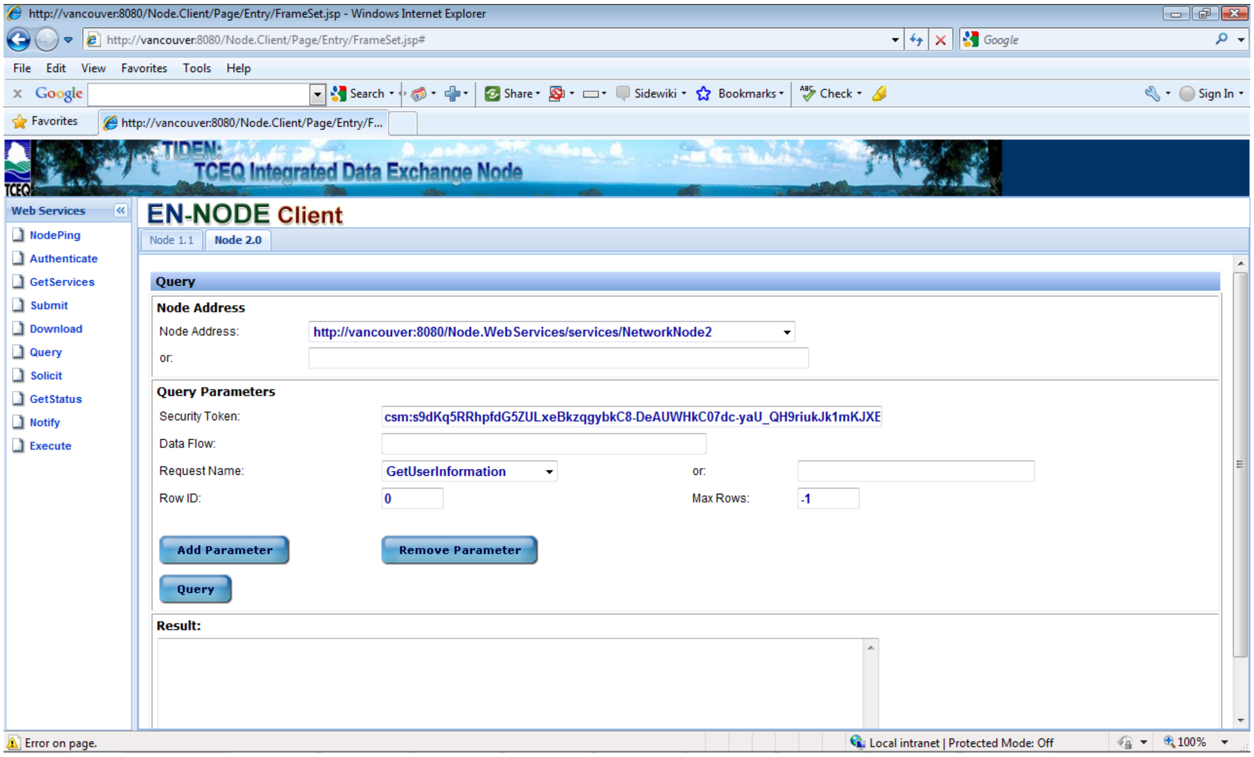
The transaction ID can be used later to check the status of the submission by calling the GetStatus Service (see Section 2.7)

2.5 Query

After receiving a security token go to the Query screen in order to perform a synchronous query of data that is offered by the target Node. A Query will require certain input parameters to be supplied that are used to filter the data returned. These parameters must be entered in a particular order and in some cases may restrict what values can be supplied. To determine what the input parameters are for a particular query, consult the Flow Configuration Document for the particular data flow that is being requested. To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Request**: The name of the particular query that is being requested (this can either be chosen from a drop-down list or entered as free text).
- **Row ID**: A query will return a number of a certain number of records. To limit which records are returned the user can enter the starting row ID. If the user would only like records after row 10 to be returned in the resulting XML file, then the value 10 should be supplied. In order to return all rows the value 0 should be supplied.
- **Max Rows**: In order to restrict the number of rows returned by the query, the user can enter how many rows they would like returned. In order to return all available rows enter a value of -1.
- **Parameter List**: Enter the values for each of the input parameters of the query. Remember these parameters are predefined and must be entered in a particular order. If a parameter should be left blank click the “Null” check box. **Note**: after selecting which request will be performed the specific input parameters for the request will be displayed.
- **Flow Operation**: Defined by 2.0 Node Spec., It indicate the specific processing for the document
- **Recipient**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction ID of the submission.
- **Notification URI**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction id, status code, and statusDetail of the submission. If it contains a Node URI, the Notify method will be called on the URI, including the transaction ID and transaction status information

After entering the required information click the “Query” Button. A query will then be performed and all the text of the returned XML file will be displayed in the “Query Result” box, as shown below:



http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer
http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#
File Edit View Favorites Tools Help
Google
Search
Share
Sidewiki
Bookmarks
Check
Sign In
Favorites
http://vancouver:8080/Node.Client/Page/Entry/F...
TIDEN:
TCEQ Integrated Data Exchange Node
Web Services
NodePing
Authenticate
GetServices
Submit
Download
Query
Solicit
GetStatus
Notify
Execute
EN-NODE Client
Node 1.1
Node 2.0
Add Parameter
Remove Parameter
Query
Error on page.
Local intranet | Protected Mode: Off
100%

Query

Node Address

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

or:

Query Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Data Flow:

Request Name: GetUserInformation or:

Row ID: 0 Max Rows: -1

Result:

Figure 2-7: Query Web Service Call Screen

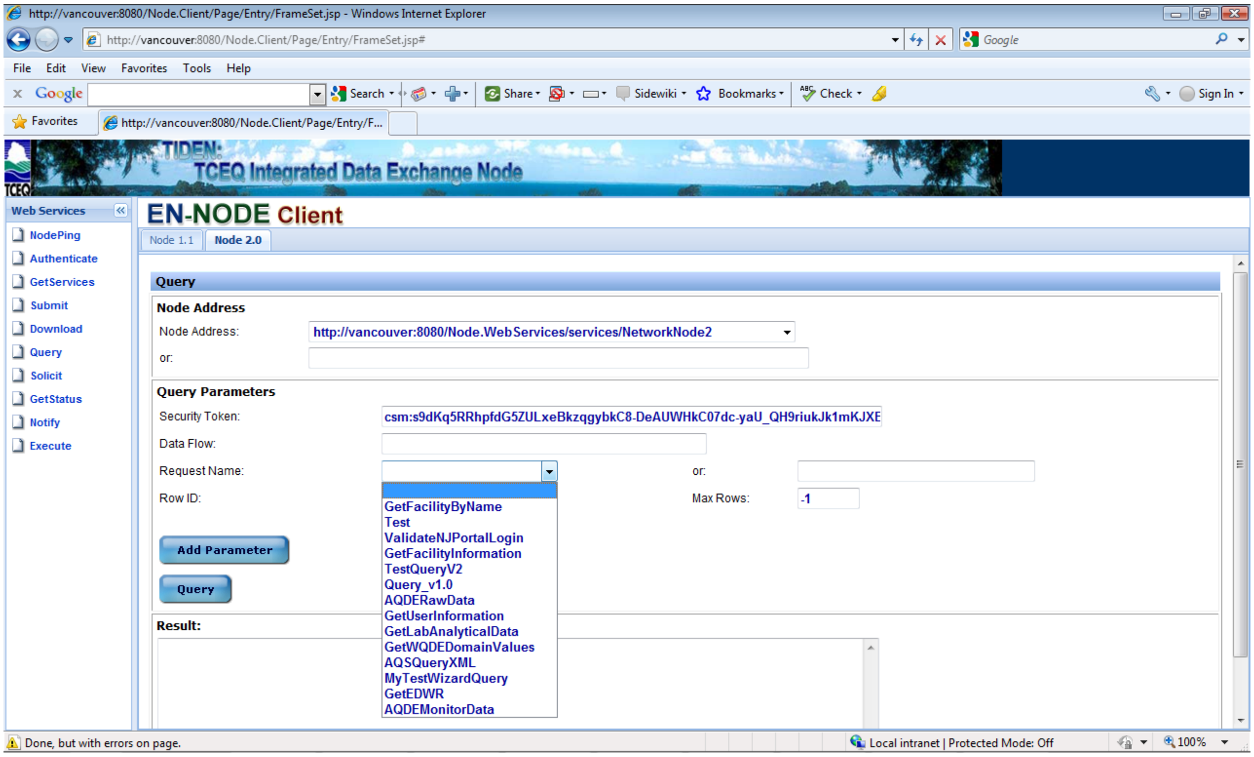
In order to Save, or Open the returned XML file, click the “Save Result” button. A dialog box will then open asking the user if they would like to Save the file or Open the file.

2.6 Solicit

After receiving a security token go to the Solicit screen in order to perform an asynchronous query of data that is offered by the target Node. A Solicit will require certain input parameters to be supplied that are used to filter the data returned. These parameters must be entered in a particular order and in some cases may restrict what values can be supplied. To determine what the input parameters are for a particular solicit, consult the Flow Configuration Document for the particular data flow that is being requested. To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Return URL**: Because the Solicit is an asynchronous query the results will not be returned immediately. By entering a Return URL the Providing Node will invoke a Submit on the Node who's URL is provided here to return the result of the Solicit. If the return URL is left blank the data requester must perform a subsequent "Download" Service (see section 2.8) after confirming the status of the transaction has changed to "Completed" by invoking the GetStatus Service (see section 2.7). The transaction ID that is returned as a result of the Solicit will be needed for these to services.
- **Request**: The name of the particular query that is being requested (this can either be chosen from a drop-down list or entered as free text).
- **Parameter List**: Enter the values for each of the input parameters of the Solicit. Remember these parameters are predefined and must be entered in a particular order. If a parameter should be left blank click the "Null" check box. **Note**: after selecting which request will be performed the specific input parameters for the request will be displayed.
- **Recipient**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction ID of the submission.
- **Notification URI**: Defined by 2.0 Node Spec., if it contains valid email address, EN-Node will send out the email with transaction id, status code, and statusDetail of the submission. If it contains a Node URI, the Notify method will be called on the URI, including the transaction ID and transaction status information

After entering the required information click the "Solicit" Button. The request will then be sent to the target Node and a transaction ID will be returned in the "Solicit Result" box, as shown below:



http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer

http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#

Query

Node Address

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

or:

Query Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Data Flow:

Request Name:

or:

Row ID:

Max Rows: -1

Result:

Done, but with errors on page.

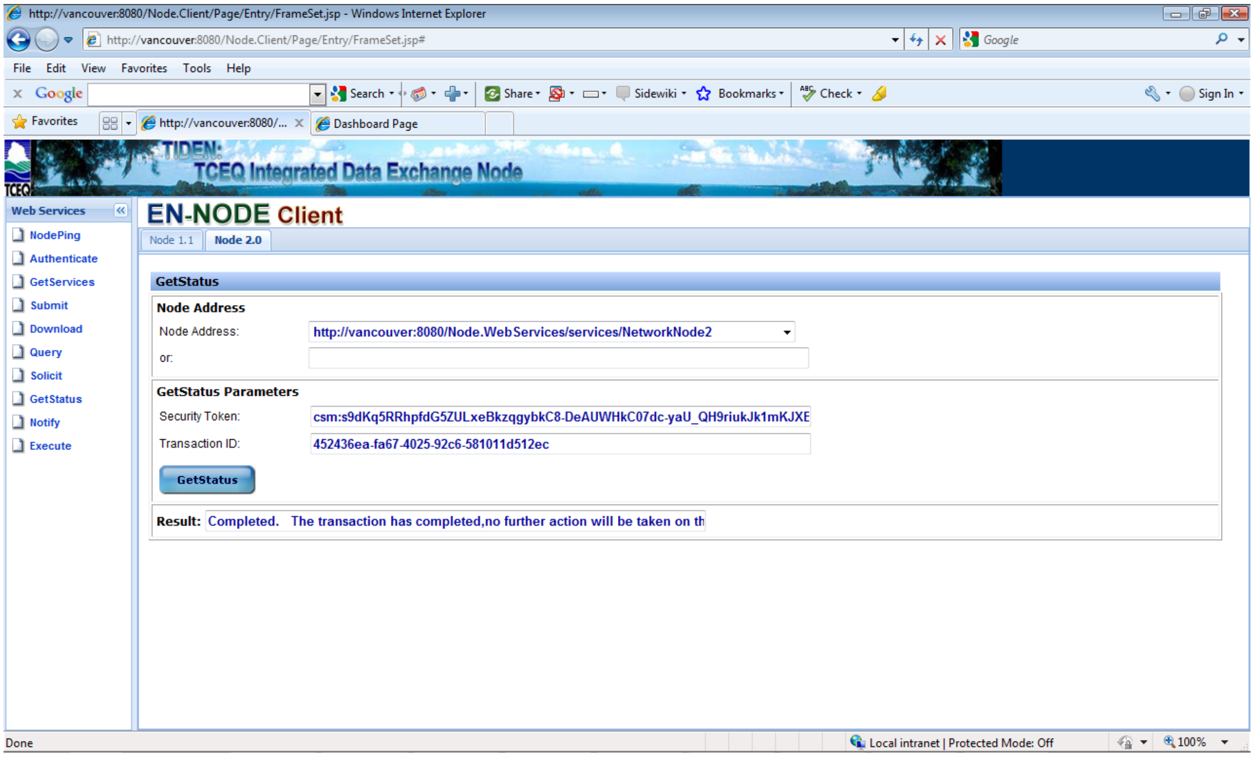
Figure 2-8: Solicit Web Service Call Screen

2.7 Get Status

After receiving a security token go to the Get Status screen in order to check the status of any outstanding transactions that have been made to the target Node (this may include any Submits or Solicits that have been made). To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Transaction ID**: This is the ID that was returned from the transaction that the user would like to check the status of.

After entering the required information click the “Get Status” Button. The request will then be sent to the target Node and the status of the desired transaction will be returned in the “Solicit Result” box, as shown below:



Node Address

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

or:

GetStatus Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Transaction ID: 452436ea-fa67-4025-92c6-581011d512ec

Result: Completed. The transaction has completed,no further action will be taken on th

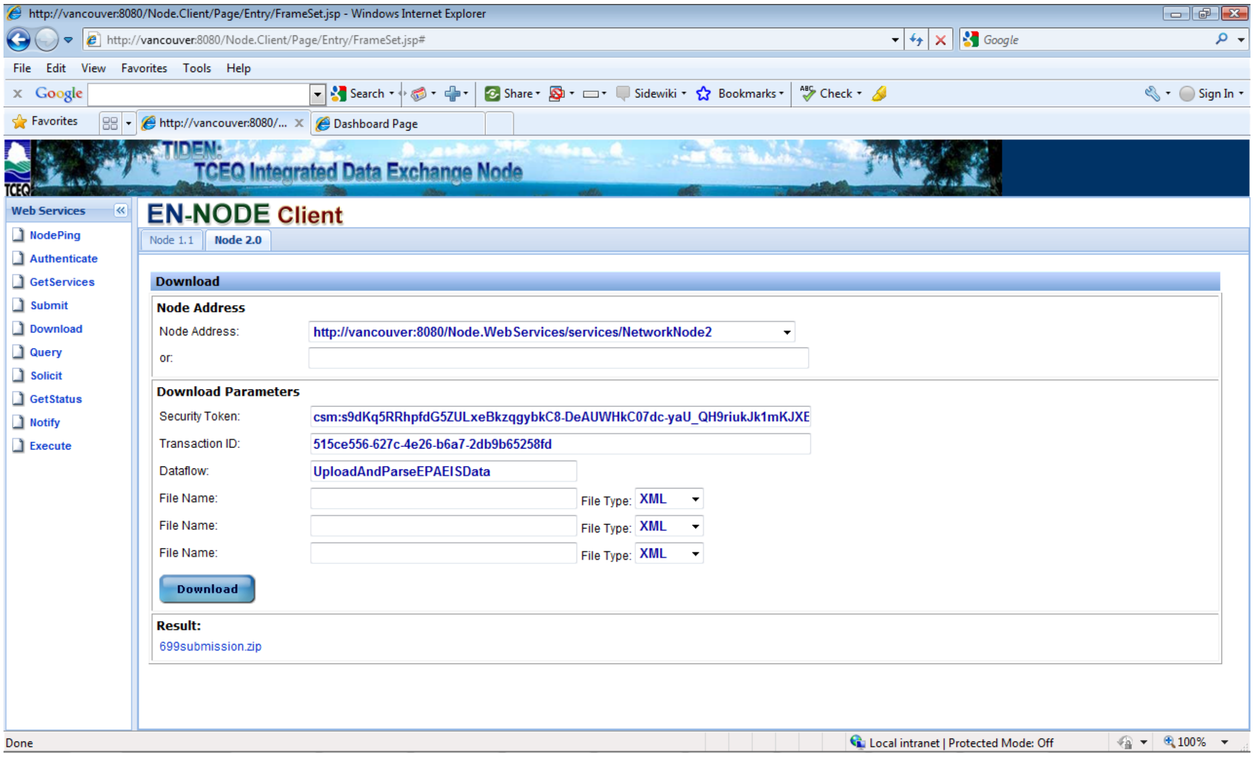
Figure 2-9: Get Status Web Service Call Screen

2.8 Download

After receiving a security token go to the Download screen in order to download files from the target Node (this service may be needed to download the results of Submit requests). To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Transaction ID (optional)**: This is the ID that was returned from the transaction that the user would like to download a file from.
- **Data Flow (optional)**: The name of the data flow the user would like to download files from.
- **File Name (optional)**: If the file name is known it can be entered.
- **File Type (optional)**: If the file name is entered the file type can also be entered.

After entering the required information click the “Download” Button. The request will then be sent to the target Node and a hyperlink to the returned files will be provided in the “Download Result” box, as shown below:


http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer
http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#
File Edit View Favorites Tools Help
Google Search Share Sidewiki Bookmarks Check Sign In
Favorites http://vancouver:8080/... Dashboard Page
TIDEN: TCEQ Integrated Data Exchange Node
Web Services
NodePing
Authenticate
GetServices
Submit
Download
Query
Solicit
GetStatus
Notify
Execute
EN-NODE Client
Node 1.1
Node 2.0
Download
Done
Local intranet | Protected Mode: Off
100%

Node Address

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

or:

Download Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

Transaction ID: 515ce556-627c-4e26-b6a7-2db9b65258fd

Dataflow: UploadAndParseEPAEISData

File Name: File Type: XML

File Name: File Type: XML

File Name: File Type: XML

Download

Result:

699submission.zip

Figure 2-10: Get Status Web Service Call Screen

In order to Save, or Open the file returned click the hyperlink. A dialog box will then open asking the user if they would like to Save the file or Open the file.

2.9 Notify

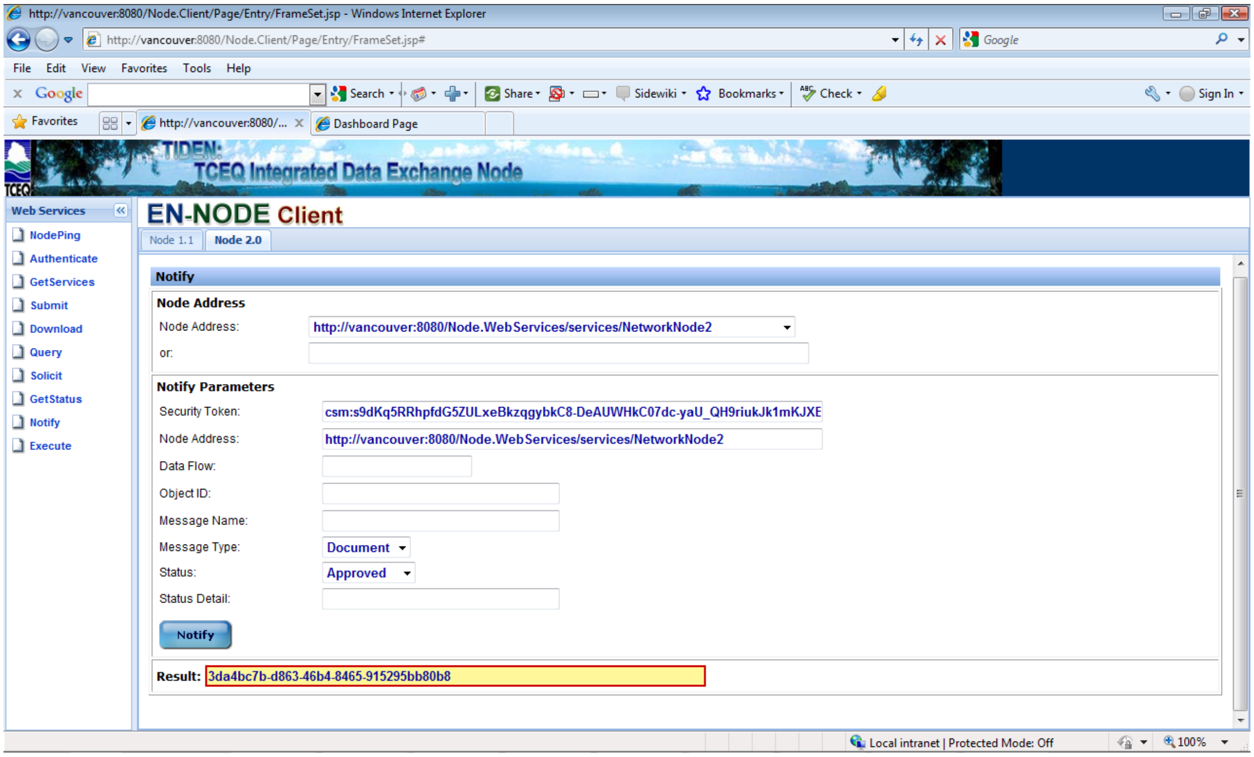
After receiving a security token go to the Notify screen in order to perform a Notify Web Service. The Notify method has three (3) intended uses: document notification, event notification, and status notification described as follows:

- Document notification: A Node or client notifies a service provider about availability of some documents (soliciting). The service provider can retrieve the documents anytime.
- Event notification: A Node sends, or possibly broadcasts, an event that is of interest to other parties. Event messages can be security alerts, shutdown notices, and other Network management notes.
- Status notification: A service provider sends a message to a requester to provide the current status of a submission or service request. The Notify Service To call the service, the following user information is needed:

To call the service, the following user information is needed:

- **Node Address**: This will be automatically populated based on which Node was used in the Authentication step
- **Security Token**: This field will be automatically populated with the security token received from the Authenticate service call.
- **Node Address**: For document notification, the parameter contains a Network Node address where the document can be downloaded. It should contain the initiator's Node address, or be empty if not applicable, for event and status notifications.
- **Dataflow**: The target dataflow that identifies an event or status if the value is http://www.exchangenetwork.net/node/event or http://www.exchangenetwork.net/node/status.
- **Documents**: For Document notification list an array of documents. To navigate to a document click the "Browse" button.
- **File Type**: If the file name is entered the file type can also be entered.

After entering the required information click the "Notify" Button. A transaction ID for the Document Notification will then be returned in the "Notify Result" box, as shown below:


http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp - Windows Internet Explorer
http://vancouver:8080/Node.Client/Page/Entry/FrameSet.jsp#
File Edit View Favorites Tools Help
Google Search Share Sidewiki Bookmarks Check Sign In
Favorites
http://vancouver:8080/...
Dashboard Page
TIDEN: TCEQ Integrated Data Exchange Node
TCEQ
Web Services
NodePing
Authenticate
GetServices
Submit
Download
Query
Solicit
GetStatus
Notify
Execute
EN-NODE Client
Node 1.1
Node 2.0
Local intranet | Protected Mode: Off
100%

Notify

Node Address

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

or:

Notify Parameters

Security Token: csm:s9dKq5RRhpfdG5ZULxeBkzqgybkC8-DeAUWHkC07dc-yaU_QH9riukJk1mKJXE

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

Data Flow:

Object ID:

Message Name:

Message Type: Document

Status: Approved

Status Detail:

Notify

Result: 3da4bc7b-d863-46b4-8465-915295bb80b8

Figure 2-4: Notify Web Service Call Screen

For 2.0 Node, there are additional information is needed

- **Object ID**: The unique ID associated with the notification Object. It should be transaction ID for transaction notification, Document ID for document notification, and Event Name for event notification.
- **Message Name**: The name of notification message.
- **Message Category**: This is a notification type of either: Event, Document or Transaction.
- **Status**: The current status of the object

The transaction ID returned can be used to query the status of the submission by using the GetStatus method (see Section 2.7)