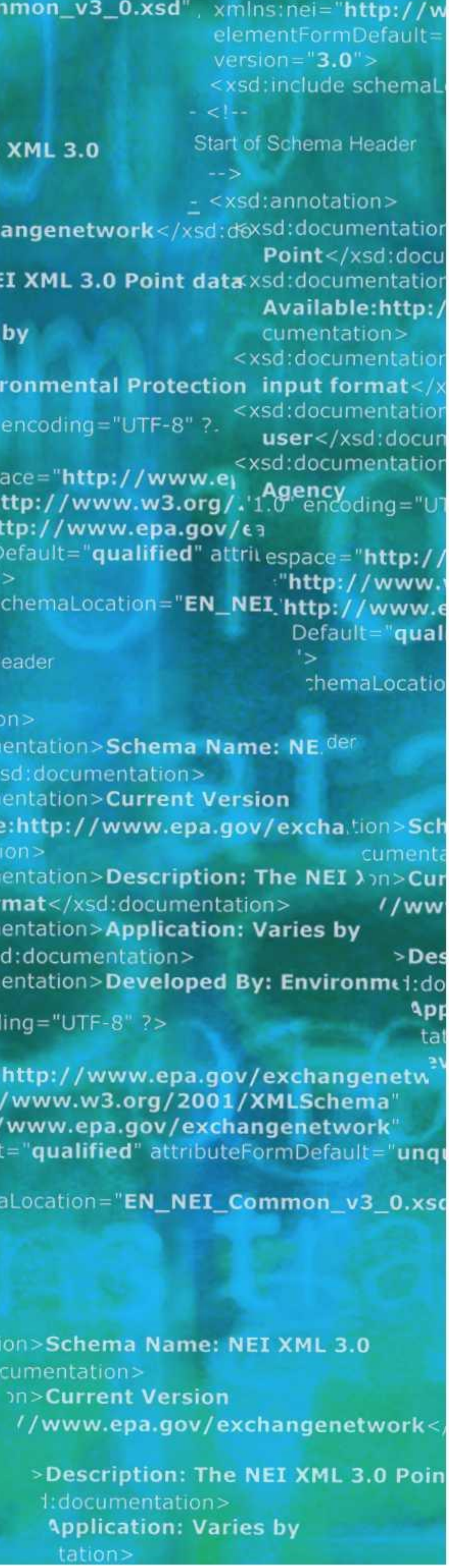


OpenNode2

Institutional Controls (IC) 1.0 Data Exchange Implementation Guide (Java)

Revision Date: 9/20/2013

Prepared By:

WINDSOR SOLUTIONS

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

Environmental Information eXchange Network

Revision History

Date	Author	Changes	Version
9/20/2003	Windsor	Initial version	1.0

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Data Exchange Overview

The purpose of this document is to provide detailed instructions for the installation and configuration of the Exchange Network Institutional Controls (IC) data exchange on the Java implementation of the Exchange Network OpenNode2 (OpenNode2).

The IC data exchange supports data publishing services, providing peer-to-peer sharing of IC data between State or Tribal agencies, and private industry. At this time, there is no centralized data repository (e.g., an EPA data repository).

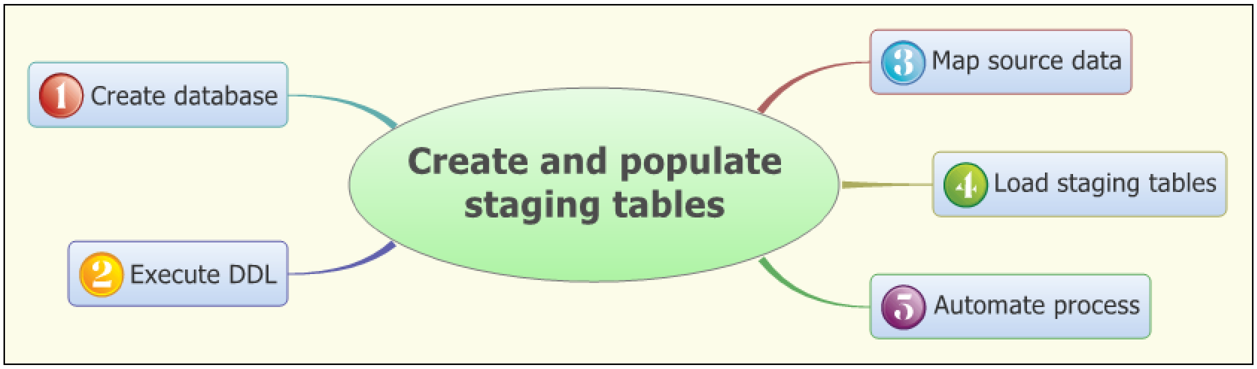
The IC data exchange configuration process involves two main steps: 1) create and populate the IC staging tables and 2) install and configure the IC data exchange plugin and services. The remainder of this document will describe these two processes in detail.

Further information about the IC data exchange is available at http://www.exchangenetwork.net/data-exchange/ic/.

Create and Populate the IC Staging Tables

OpenNode2 uses a plugin-based architecture to support data exchanges with Exchange Network partners. Data must first be loaded into a set of staging tables before it can be extracted by the plugin and shared through the IC data exchange. This section outlines the steps required to set up the IC data exchange database staging tables.

The following figure illustrates these steps:

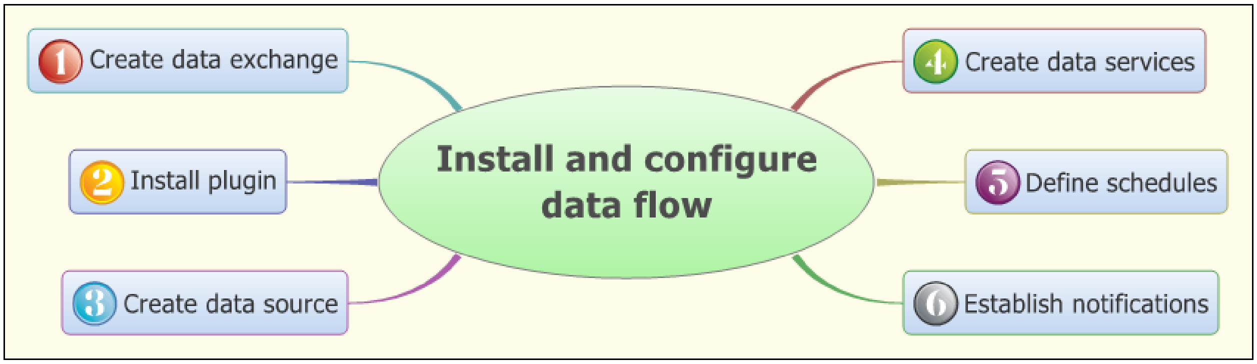


1. The first step is to create the staging database itself if one has not already been established to support another data exchange (typically named NODE_FLOW).
2. Once the staging database is created, a Database Definition Language (DDL) script included in the IC plugin package can be executed to create the staging tables that will be used to store the data being made available through the IC data exchange.
3. With the staging environment established, data must now be mapped from the source database to the equivalent fields in the IC staging tables. The staging tables closely reflect the structure and naming of the IC XML schema, and it is recommended that the Data Exchange Template (DET) published at http://www.exchangenetwork.net/data-exchange/ic/ be used to facilitate this mapping.
4. Once the mapping is complete, a database extract, transform, and load (ETL) routine should be developed to populate the tables in the staging database using the mapping prepared during the earlier step. This should be a repeatable process that will empty and replace all of the data in the staging tables, or a procedure that will incrementally add, update and remove data as it changes in the source system.
5. Once the data extract process has been developed, it should be automated to execute on a regular schedule as appropriate to the needs of the organization and its Exchange Network partners.

Install and Configure the IC Data Exchange

This section describes the steps required to install and configure the IC data exchange on the Java implementation of the OpenNode2 using the Node Administration Web application (Node Admin).

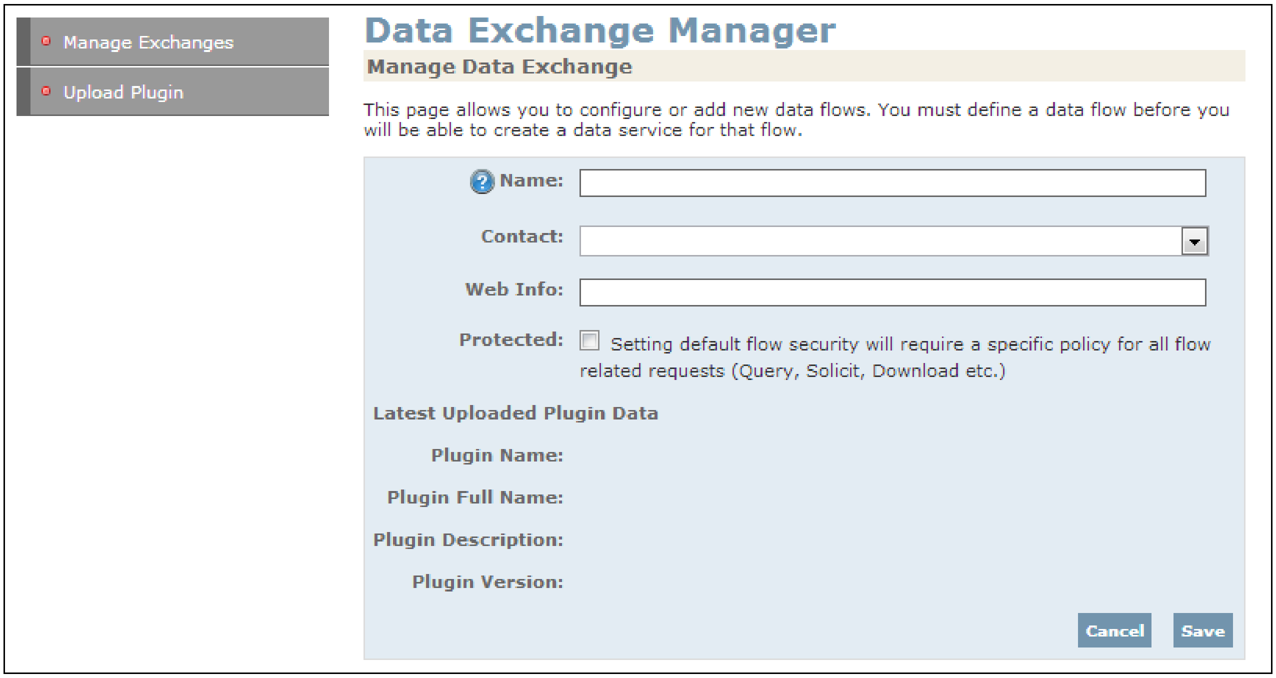
The following figure illustrates these steps:



Create the IC Data Exchange

The first step is to create the IC data exchange using the Node Admin Web application.

1. After logging into the Node Admin, click the **Exchange** tab on the top navigation bar.
2. Click the **Add Exchange** button. The Manage Data Exchange screen will be displayed as follows:



Data Exchange Manager

Manage Data Exchange

This page allows you to configure or add new data flows. You must define a data flow before you will be able to create a data service for that flow.

Name:

Contact:

Web Info:

Protected: □ Setting default flow security will require a specific policy for all flow related requests (Query, Solicit, Download etc.)

Latest Uploaded Plugin Data

Plugin Name:

Plugin Full Name:

Plugin Description:

Plugin Version:

3. Type “IC” in the **Name** field.

4. Select a user account name from the **Contact** drop down box. Contacts are populated with all accounts that have been set up on the OpenNode2. See the **Security** tab for a list of available accounts.

5. In the **Web Info** field, enter a URL where more information can be found about the IC data exchange. It is recommended that the following URL be used for this purpose http://www.exchangenetwork.net/data-exchange/ic/.

6. It is recommended that the **Protected** box not be checked. This will allow authorized users access to the IC services without the need for an OpenNode2 administrator to specifically grant permissions to the user. If the **Protected** box is checked, accounts and account security policies must be set up on the OpenNode2 in order for those accounts to access the IC data services.

   Please see the OpenNode2 Administration User Guide for more information on managing accounts and account policies.

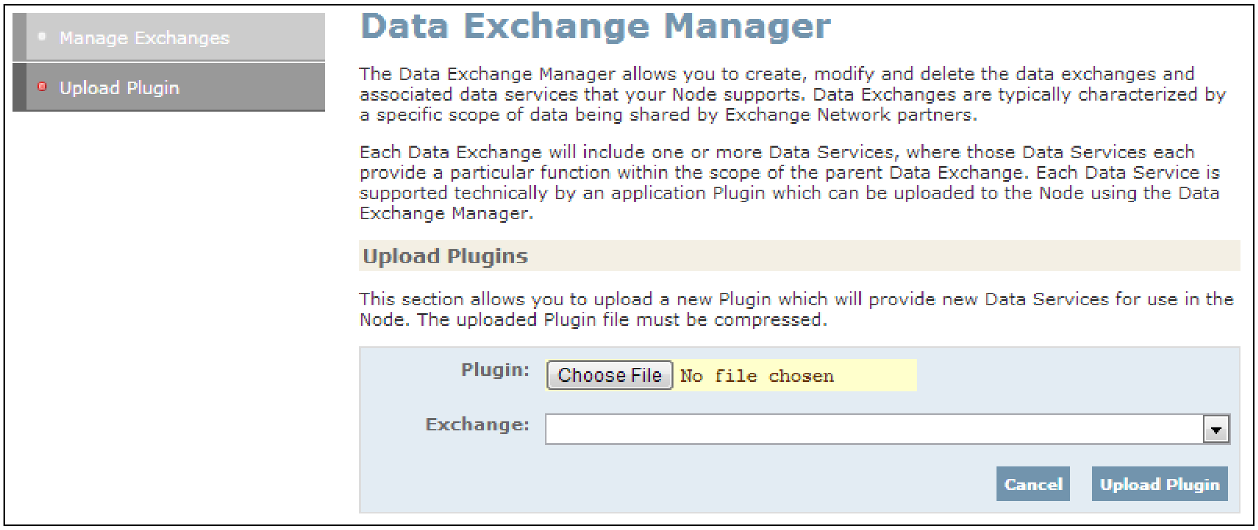
7. Click the **Save** button to save the data exchange to the OpenNode2 metadata database.

Install the IC Plugin

Once the data exchange has been created, the next step is to upload the IC plugin into the OpenNode2 plugin repository.

1. Click the **Exchange** tab on the top navigation bar.

2. Click the **Upload Plugin** section on the left navigation bar. The Upload Plugin screen will be displayed as follows:



Data Exchange Manager

The Data Exchange Manager allows you to create, modify and delete the data exchanges and associated data services that your Node supports. Data Exchanges are typically characterized by a specific scope of data being shared by Exchange Network partners.

Each Data Exchange will include one or more Data Services, where those Data Services each provide a particular function within the scope of the parent Data Exchange. Each Data Service is supported technically by an application Plugin which can be uploaded to the Node using the Data Exchange Manager.

Upload Plugins

This section allows you to upload a new Plugin which will provide new Data Services for use in the Node. The uploaded Plugin file must be compressed.

Plugin: Choose File No file chosen

Exchange:

3. Click the **Browse** button which is located to the right of the **Plugin** field.

4. Locate and select the compressed (zipped) file containing the code component for the IC plugin you obtained with the OpenNode2 installation package or separately from the OpenNode2 download site.

5. Select the data exchange name “IC” created during the previous step from the **Exchange** dropdown box.

6. Click the **Upload Plugin** button to upload the plugin.

The newly uploaded plugin code will be placed in the OpenNode2 plugin repository. Any previous plugin versions will be retained in the repository but won't be accessible through the Node Admin. Only the latest version of any one plugin is made available during the next step to establish data services.

Create the IC Data Services

Data services are distinct functions provided by a plugin to support a given data exchange. In the case of the IC flow, there are three specific data services provided by the plugin:

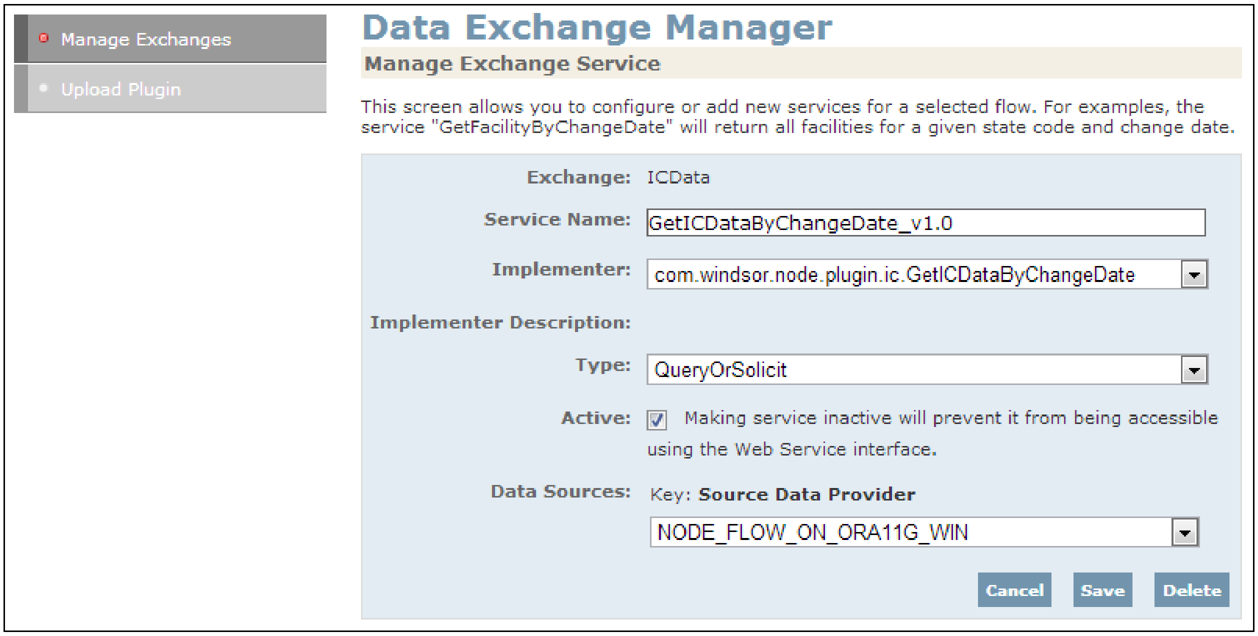
- GetICDataByChangeDate
- GetICDataByBoundingBox
- GetICDataByParameters

Each of these data services must be created and configured before they can be accessed through the OpenNode2.

GetICDataByChangeDate

This service is used to support the creation and maintenance of a replica set of institutional control data across partners (i.e., data synchronization). This data service will retrieve records from the staging database according to the Change Date parameter value supplied in the request and will generate an XML file matching the criteria supplied in the query request.

1. From the **Exchange** tab, scroll down the list of installed data exchanges until the IC exchange is located.
2. Click the **Add Service** button located just beneath the IC data exchange heading. The following page will be displayed to allow a new data service to be added.



This screen allows you to configure or add new services for a selected flow. For examples, the service "GetFacilityByChangeDate" will return all facilities for a given state code and change date.

Exchange: ICData

Service Name: GetICDataByChangeDate_v1.0

Implementer: com.windsor.node.plugin.ic.GetICDataByChangeDate

Implementer Description:

Type: QueryOrSolicit

Active: ☑ Making service inactive will prevent it from being accessible using the Web Service interface.

Data Sources: Key: **Source Data Provider**

NODE_FLOW_ON_ORA11G_WIN

3. In the **Service Name** field, type "GetICDataByChangeDate_v1.0".
4. From the **Implementer** drop down box, select the appropriate value for this data service.

Note: When the implementer is selected and the ***Next*** *is clicked, several additional arguments will appear. The Node Admin application will obtain these properties directly from the IC plugin.*

5. From the **Type** drop down box, select how you wish to make the services available. The options available will be obtained from the plugin by the Node Admin. Select “QueryOrSolicit”.
6. Enable the service by checking the **Active** checkbox.
7. Based on the selection made from the implementer drop-down menu, the Node Admin will determine what argument and data source requirements the plugin has and will display the relevant data entry fields as follows:
    a. **Data Sources**: Set the **Source Data Provider** to the data source that connects to the IC staging tables by selecting from the dropdown box.

       Please see the OpenNode2 Administration User Guide for information on setting up data sources.
8. Click the **Save** button to save the service.

GetICDataByBoundingBox

This service returns all IC data within a bounding box (area) specified by maximum and minimum latitude/longitude coordinate pairs. This data service retrieves records from the staging database according to the input runtime parameters and will generate an XML file containing the data that matches the criteria supplied in the query request.

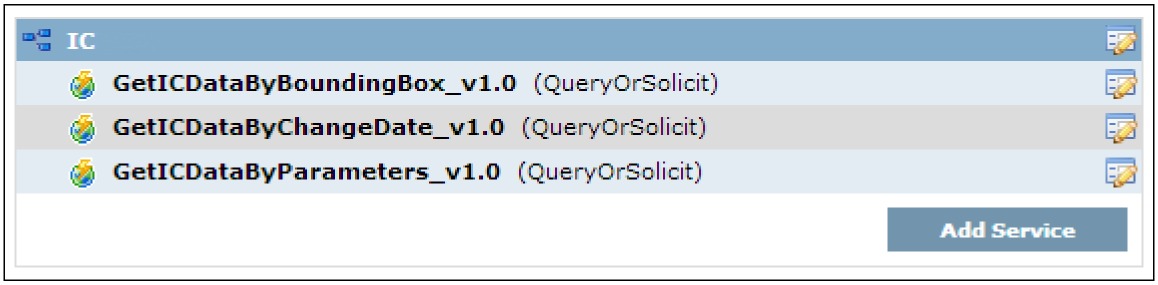
1. Repeat the steps above to set up the service, setting the **Service Name** field to “GetICDataByBoundingBox_v1.0”, and selecting the appropriate value from the **Implementer** drop down box.
2. Click the **Save** button to save the service.

GetICDataByParameters

Allows for a variety of filter criteria to be supplied when retrieving institutional control data. This data service will retrieve records from the staging database according to the input runtime parameters and will generate an XML file containing the data that matches the criteria supplied in the query request.

1. Repeat the steps above to set up the service, setting the **Service Name** field to “GetICDataByParameters_v1.0”, and selecting the appropriate value from the **Implementer** drop down box.
2. Click the **Save** button to save the service.

The **Manage Exchanges** page for the IC data exchange should appear as follows:



IC

GetICDataByBoundingBox_v1.0 (QueryOrSolicit)

GetICDataByChangeDate_v1.0 (QueryOrSolicit)

GetICDataByParameters_v1.0 (QueryOrSolicit)

Define Data Exchange Schedules

Because the IC data exchange will support only data publishing services, it is not required to define scheduled jobs in order to support the data exchange.

However, scheduled jobs can be configured, if desired, for:

- Testing the flow. A schedule can be created and run to execute one of the IC Services. The plugin will create an XML response and save it to the OpenNode2 database. The XML response can then be downloaded from OpenNode2 Admin and analyzed to ensure the expected records were returned.
- Scheduling/executing IC data exchange Query or Solicit services to retrieve IC data from Exchange Network partners who have implemented the IC data exchange on their Exchange Network node.

Please see the OpenNode2 Administration User Guide and the IC Flow Configuration Document (FCD) for information on setting up scheduled jobs using the IC data publishing services.

Establish Email Notifications

If desired, the Node administrator may create NAAS accounts for one or more staff members and create notifications for the any OpenNode2 events related to the IC data exchange.

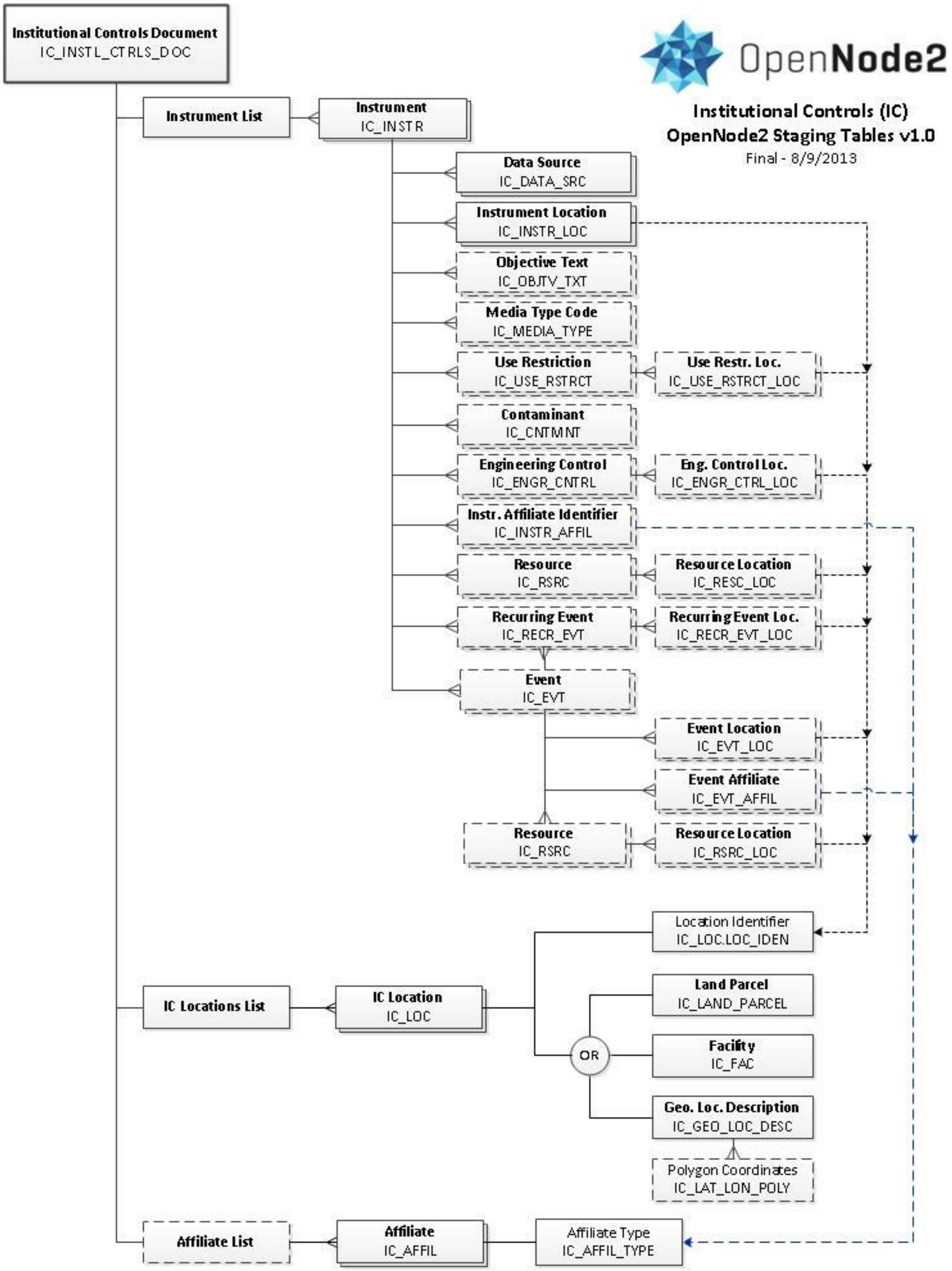
Please see the OpenNode2 Administration User Guide for more information on setting up notifications.

Monitor Flow Activity

The OpenNode2 will track all IC data exchange activity and can be accessed to monitor and debug related flow activities.

Please see the OpenNode2 Administration User Guide for more information on accessing and searching the available OpenNode2 activity reports.

Appendix A – IC Staging Table Diagram



Appendix B – REST Service URL Query Syntax:

The implementation of REST services and associated URL query syntax are described in detail within the OpenNode2 REST Service Specification (.Net and Java) located at :

https://code.google.com/p/opennode2/downloads/detail?name=OpenNode2_REST_Specification.pdf&can=2&q=

The following syntax was derived from this specification:

The OpenNode2 REST URL syntax is as follows:

> *NodeURL*/*RestEndpoint*/**Query?Dataflow**=*dataflow***&Request**=*request*[**&Username**=*username* **&Password**=*password*][**&token**=*token*][**&ZipResults**=*TrueOrFalse*][**&Params**=*params*]

Optional parameters are shown in [brackets]. Static portions of the URL are shown in **bold** face.

Parameter	**Description**	**Example Value**
NodeURL	The base URL of the OpenNode2 instance	https://www.myagency.gov/node
RestEndpoint	The URL path to the REST endpoint. While the OpenNode2 installation guide recommends an endpoint application name of “REST”, agencies may choose to use a different application name.	REST
dataflow	The Exchange Network data flow to query. The name must match exactly with the standard flow identifier prescribed in the FCD for the exchange. OpenNode2 administrators may choose	FacID_v3.0
request	The name of the query to invoke. The value provided should match the query service name defined in the FCD for the given data exchange (dataflow).	GetFacility_v3.0
username	The NAAS username to use to authenticate to OpenNode2.	me@myagency.gov
password	The NAAS password to use to authenticate to OpenNode2.	MyP@ssword1!
token	In lieu of explicitly supplying a username and password, a valid NAAS-issued token can be provided. This allows a client to authenticate to NAAS directly instead of by proxy via the OpenNode2 REST query.	FTOVMp9-D_L7LadLZmT0…
TrueOrFalse	Set to True if zipped response is to be returned from OpenNode2. The default is false if this parameter is not supplied in which case the response will be XML. Note that	True
Params	The query parameters (filters) to apply. The	Facility Name|ACME%;

	parameters supported are specific to the data flow and service. Please see the section below for the syntax requirements for defining parameters.	Zip Code|48123

Example

The following REST service URL's are a sample of queries that may be constructed for this data exchange and are based upon the Institutional Control Data Publishing Queries outlined in the Flow Configuration Document (http://www.exchangenetwork.net/data-exchange/ic/).

Common escape characters used when specifying parameters.

%3B = ;

%7C = |

%20 = single white space character

%25 = wildcard character (%)

ASCII Codes

http://www.december.com/html/spec/asciiall.html

GetICDataByParameters

No parameters specified.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=

Get where InstrumentIdentifier = INST-BBB

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=InstrumentIdentifier%7CINST-BBB

Get where InstrumentIdentifier = INST-BBB or INST-CCC

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=InstrumentIdentifier%7CINST-BBB%7CINST-CCC

Get where InstrumentIdentifier = INST-BBB or INST-CCC and ChangeDate = 2013-08-27.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=InstrumentIdentifier%7CINST-BBB%7CINST-CCC%3BChangeDate%7C2013-08-27%7C

Get where InstrumentIdentifier = INST% (use wildcard character as specified in FCD)

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=InstrumentIdentifier%7CINST%25

Get where FacilityIdentifier = SMITH.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilityIdentifier%7CSMITH

Get where FacilityIdentifier = SMITH or JONES

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilityIdentifier%7CSMITH%7CJONES

Get where FacilityIdentifier = SMI% (use wildcard character as specified in FCD)

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilityIdentifier%7CSMI%25

Get where FacilityIdentifier LIKE SMI% (use wildcard character as specified in FCD) and FacilitySiteName = Smith Lead Pencil Factory.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilityIdentifier%7CSMI%25%3BFacilitySiteName%7CSmith%20Lead%20Pencil%20Factory

Get where FacilitySiteName = Smith Lead Pencil Factory. Should return INST-AAA.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilitySiteName%7CSmith%20Lead%20Pencil%20Factory

Get where FacilitySiteName = Smith Lead Pencil Factory or Jones Paperworks.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilitySiteName%7CSmith%20Lead%20Pencil%20Factory%7CJones%20Paperworks

Get where FacilitySiteName = Smith% (use wildcard character as specified in FCD).

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=FacilitySiteName%7CSmith%25

Get where UseRestrictionTypeCode = Restrict Excavation.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=UseRestrictionTypeCode%7CRestrict%20Other

Get where UseRestrictionTypeCode = Restrict Excavation or Restrict Other.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=UseRestrictionTypeCode%7CRestrict%20Excavation%7CRestrict%20Other

Get where UseRestrictionTypeCode = Restrict Excavation or Restrict Other and where bounding box of lat 40-50 and long -90 to -80.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=UseRestrictionTypeCode%7CRestrict%20Excavation%7CRestrict%20Other%3BBoundingCoordinateNorth%7C50%3BBoundingCoordinateSouth%7C40%3BBoundingCoordinateWest%7C-90%3BBoundingCoordinateEast%7C-80

Get where UseRestrictionTypeCode = Restrict% (use wildcard character as specified in FCD).

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=UseRestrictionTypeCode%7CRestrict%25

Get where bounding box of lat 40-50 and long -90 to -80. Should return INST-AAA and INST-BBB.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=BoundingCoordinateNorth%7C50%3BBoundingCoordinateSouth%7C40%3BBoundingCoordinateWest%7C-90%3BBoundingCoordinateEast%7C-80

Get where bounding box of lat 40-45 and long -88.5 to -80.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=BoundingCoordinateNorth%7C45%3BBoundingCoordinateSouth%7C40%3BBoundingCoordinateWest%7C-88.5%3BBoundingCoordinateEast%7C-80

Get where ChangeDate as of 2013-08-27.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2013-08-27%7C

Get where ChangeDate < all last update dates.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2012-07-30%7C

Get where ChangeDate is > all last updates dates.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByParameters_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2013-09-05%7C

GetICDataByBoundingBox

Get where bounding box of lat 40-50 and long -90 to -80.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByBoundingBox_v1.0&Username=cdx&Password=test&ZipResults=false&Params=BoundingCoordinateNorth%7C50%3BBoundingCoordinateSouth%7C40%3BBoundingCoordinateWest%7C-90%3BBoundingCoordinateEast%7C-80

GetICDataByChangeDate

Get where ChangeDate as of 2013-08-27.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByChangeDate_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2013-08-27%7C

Get where ChangeDate < all last update dates.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByChangeDate_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2012-07-30%7C

Get where ChangeDate is > all last updates dates.

http://jct.windsor.com:9080/wnrest/services/Query?Dataflow=ICData&Request=GetICDataByChangeDate_v1.0&Username=cdx&Password=test&ZipResults=false&Params=ChangeDate%7C2013-09-05%7C