OpenNode2

RCRAInfo 5.10 Data Exchange Implementation Guide (.NET)

Revision Date: 03/22/2021

Prepared By:

WINDSOR SOLUTIONS

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

Environmental Information exchange Network

Revision History

Date	Author	Changes	Version
4/21/2009	Windsor	Initial version	1.0
5/29/2009	Windsor	Revised to reflect standardization of plugin arguments.	1.1
9/15/2009	Windsor	Revised to include CME data service	1.2
7/27/2012	Windsor	Updated to include all v5.0 plugin implementers	1.3
10/9/2013	Windsor	Revised cover page	1.4
7/9/2014	Windsor	Updated Install Plugin section to describe pre-bundled plugin process starting with OpenNode2 v2.6	1.5
3/16/2016	Windsor	Updated documentation for ExecuteRCRAExtract implementer	1.6
4/6/2016	Windsor	Add note about notifications on v1.1 vs. v2.0 endpoints	1.7
7/18/2016	Windsor	Added Permitting module diagram to Appendix A	1.8
4/13/2018	Windsor	Updated to reflect version 5.6 of schema (namely Appendix A – Staging table diagrams)	1.9
6/17/2019	Windsor	Minor updates to accommodate schema version 5.7.	2.0
10/5/2020	Windsor	Updated version number to reflect 5.9. No substantive content changes.	2.1
03/22/2021	Windsor	Updated version number to reflect 5.9. Includes new Handler Episodic Project table and new Spatial fields for ReportUniverse and Handler.	2.2

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 Resource Conservation and Recovery Act information system (RCRAInfo) inbound data exchange on the Microsoft .NET implementation of the Exchange Network OpenNode2 (OpenNode2).

The RCRAInfo data exchange offers two data services that are used to prepare and submit data from the State program system to the EPA RCRAInfo system.

Further detail about the RCRAInfo data exchange is available in the Flow Configuration Document (FCD) published at exchangenetwork.net.

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

Terminology

Inbound data flow refers to the ability for a partner to push data to another partner. In the case of EPA, the data is going from the State, and data is coming Inbound into the EPA.

Outbound data flow refers to the ability to obtain (solicit, query) data from the EPA. In other words, it is data outbound from the EPA.

> **Note:** This document describes the RCRA INBOUND data flow. To obtain RCRA Info data from the EPA, please see the RCRAInfo v5.10 **Outbound** – Plugin Implementation Guide.docx

Note about v5.7 Updates

Schema version 5.7 introduced several important changes, however most impact the outbound data flow. Minor impacts for the inbound data flow include new attributes that have been added to the Handler payload including Acknowledge Flag Indicator, Include In National Report Indicator, LQHUW Indicator and HD Report Cycle Year. Also, the GIS xsd file was renamed, so in order to flow GS data, you must upgrade to 5.7.

Plugin Architecture

The diagram below shows the architecture of a typical OpenNode2 plugin and how services that access the plugin's functionality are configured by a node administrator.
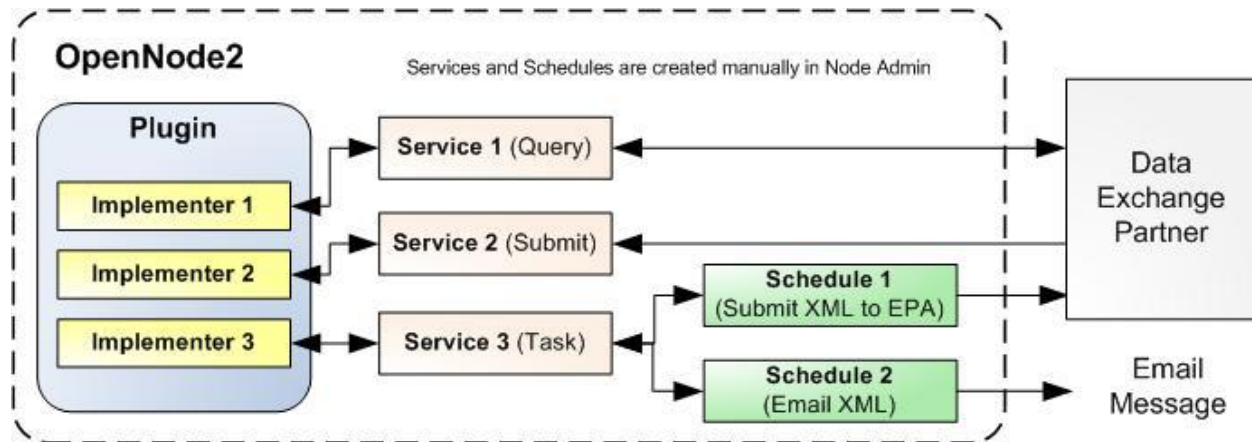
A plugin contains one or more **implementers**. Implementers are canned functionality that are specific to the data exchange. An implementer performs some task, such as composing XML from a series of staging tables.

A node administrator exposes the functionality in an implementer by creating **services**. When a service is created, an implementer must be chosen. Each service may have one or more configuration arguments, defined by the implementer. For example, the service may require that a database connection or node partner URL be provided. Services can be made available to external partners in the form of a query or solicit or as an inbound submission processor. "Task" services are internal only and are accessed via a **schedule**. Schedules also can have configuration arguments which are used by the plugin implementer assigned to the schedule.

RCRA Plugin Implementers

This section describes the different implementers available in the RCRA plugin, the arguments they require, and how they operate.

Extract and Submission Implementers

Implementer Name:	**RCRACMEExtractAndSubmission,** **RCRACorrectiveActionExtractAndSubmission,** **RCRAFinancialAssuranceExtractAndSubmission,** **RCRAGeographicInformationExtractAndSubmission,** **RCRAHandlerExtractAndSubmission,** **RCRAPermitExtractAndSubmission**
Description/Usage:	The extract and submission implementers execute the pre-defined stored procedure to extract data from the source system and loads the data into the appropriate staging tables. The implementer will then create the XML file using the arguments defined and then submit the XML file to the selected exchange partner. A record in the RCRA_SubmissionHistory table is created for each submission data type. The last successful submission date is then used by

the extract procedure the next time it is executed to determine the date range for the incremental load. Note: Only one pending submission per data type is allowed at any given time. See the GetRCRASubmissionStatus, and ClearPendingRCRASubmission implementer descriptions below on how to update or change the status from "Pending".

Note: The stored procedure must define the following parameters:

- **p_run_parm** – can be used for any purpose by the procedure, or not used at all, but it must be present. The plugin sets this parameter to "NORMAL" by default (VARCHAR datatype).

- **p_runtime_txt** – Output parameter returning a message such as "ETL Completed Successfully" (VARCHAR datatype).

- **p_runtime** – Output parameter returning the current date stamp when the ETL completed (DATE datatype)

Submission Implementers

Implementer Name:	**RCRAGetHazardousWasteCMEData, RCRAGetHazardousWasteCorrectiveActionData, RCRAGetFinancialAssuranceData, RCRAGetGISData, RCRAGetHazardousWasteHandlerData, RCRAGetHazardousWastePermitData**
Description/Usage:	The implementer will create the XML file from the data already loaded into the RCRA staging tables using the arguments defined. Once the XML file has been created, it will be submitted to the selected exchange partner.
	A record in the RCRA_SubmissionHistory table is created for each submission data type. Note: Only one pending submission per data type is allowed at any given time. See the GetRCRASubmissionStatus, and ClearPendingRCRASubmission implementer descriptions below on how to update or change the status from "Pending".

Submission Processer Implementers

Implementer Name:	**RCRASubmissionProcessor, FinancialAssuranceSubmissionProcessor, GISSubmissionProcessor, HazardousWasteCMESubmissionProcessor, HazardousWasteCorrectiveActionSubmissionProcessor, HazardousWasteHandlerSubmissionProcessor, HazardousWastePermitSubmissionProcessor**
Description/Usage:	The implementer will take a XML file from an exchange network partner and load the data into the RCRA staging tables.
	Note: the **RCRASubmissionProcessor** implementer will process any of the data types but expects a properly formatted exchange header in the XML file.

ClearPendingRCRASubmissions Implementer

Implementer Name:	**ClearPendingRCRASubmissions**
Description/Usage:	The implementer updates the status to "Failed' in the RCRA_SubmissionHistory table for all submissions with a "Pending" status.

ExecuteRCRAExtract Implementer

Implementer Name:	**ExecuteRCRAExtract**
Description/Usage:	The implementer executes a stored procedure to extract data from a source system and load it into the RCRA staging tables. This implementer is used in conjunction with the Submission Implementers listed above (e.g. RCRAGetHazardousWasteCMEData, RCRAGetGISData, etc.)

SubmissionStatus Implementer

| Implementer Name:	**GetRCRASubmissionStatus**
Description/Usage:	The implementer retrieves the submission status from the network partner, and updates the RCRA_SubmissionHistory table with the status.

SubmitProxy Implementer

Implementer Name:	**RCRASubmitProxy**
Description/Usage:	The implementer was created for a specific client and not applicable for most users. For more information, please contact Windsor Solutions for more information.

Create and Populate the RCRAInfo Staging Database

OpenNode2 uses a plugin-based architecture to support data exchanges with EPA and other 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 RCRAInfo data exchange. This section outlines the steps required to set up the RCRAInfo data exchange database staging tables.

The following figure illustrates these steps:
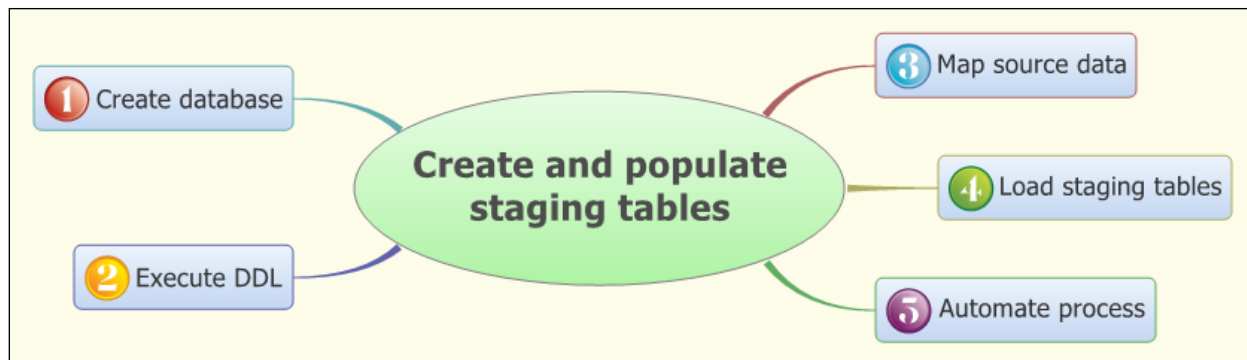
1. The first step is to create the staging database itself if one has not already been established to support another data exchange (typically named NODE_FLOW).

2. Once the staging database itself is created, a Database Definition Language (DDL) script included in the OpenNode2 deployment package can be executed to create the staging tables themselves that will be used to store the data being made available through the RCRAInfo data exchange.

3. With the staging environment established, data must now be mapped from the source database to the equivalent fields in the RCRAInfo staging tables. The staging tables closely reflect the structure and naming of the RCRAInfo XML schema, and it is recommended that the Data Exchange Template (DET) published at exchangenetwork.net be used to facilitate this mapping.

4. Once the mapping is complete, a database 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 for submissions to EPA.

Install and **Configure the RCRAInfo Data Flow**

This section describes the steps required to install and configure the RCRAInfo data exchange on the Microsoft .NET and Java implementations of the OpenNode2 using the Node Administration Web application (Node Admin).
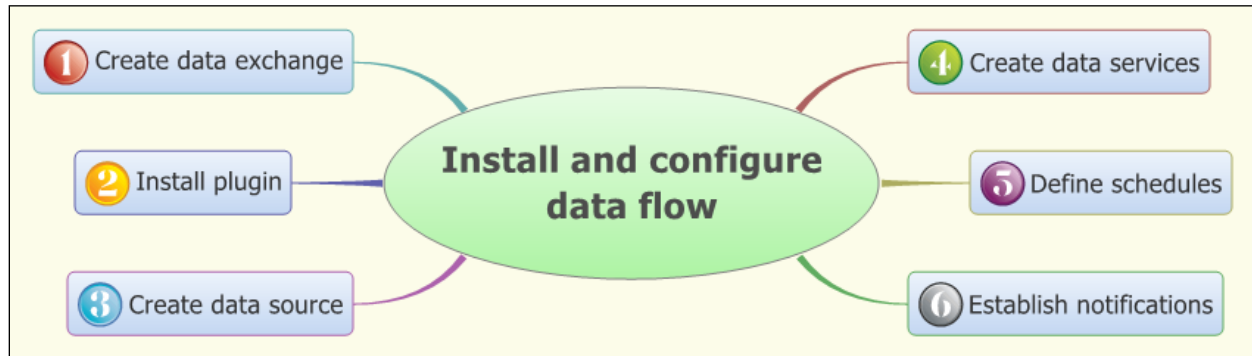
Create the RCRAInfo Data Exchange

The first step is to create the RCRAInfo data exchange using the OpenNode2 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:

3. Type "RCRA" in the **Name** field.

4. Type a short description in the **Description** field.

5. Select a user account name from the **Contact** drop down box. Contacts are populated with all accounts that have been set up on the Node 2008. See the **Security** tab for a list of available accounts.

6. The **Web Info** field can be left blank.

7. It is recommended that the **Protected** box be checked. This will require special flow specific security permissions for this data flow. External access should not be required at this time given the current purpose of this flow is solely as a means of data submission to EPA.

8. Click the **Save** button to save the data exchange to the OpenNode2 repository.
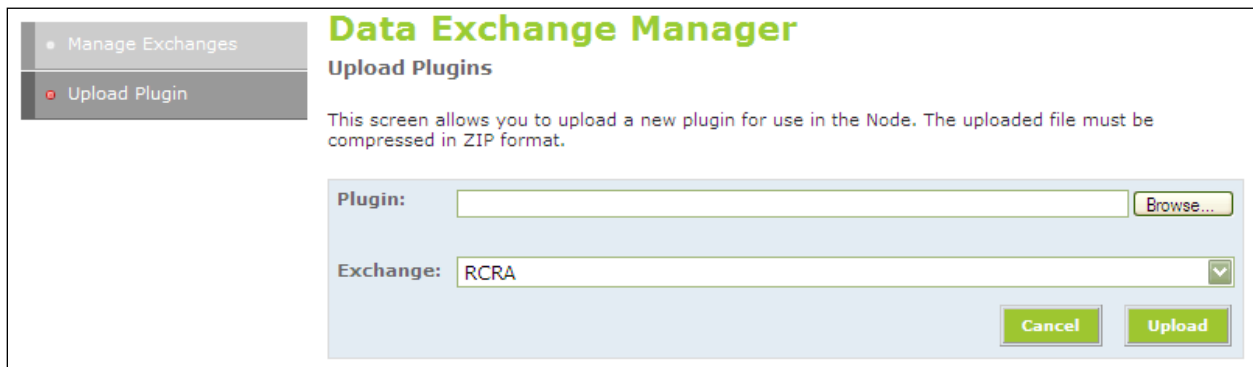
Install the RCRAInfo Plugin

Once the data exchange has been created, the next step is to upload the RCRAInfo plugin provided by Windsor into the OpenNode2 plugin repository.

Note: If you are using OpenNode2 v2.6 or higher, this step is not necessary. Starting with v2.6, all plugins are pre-installed with the OpenNode2 software installation package. By creating the exchange above, the plugin will automatically be loaded and associated with the exchange. To validate that the plugin was installed automatically, follow the steps below:

1. From the **Exchange** tab, scroll down the list of installed data exchanges until the WQX exchange is located.

2. Click the **Add Service** button located just beneath the WQX data exchange record. If the Implementer drop down box is not empty, then the plugin has been installed successfully.

If the steps above reveal that the plugin is not installed, perform the following steps to install it.

1. Navigate to the plugin directory in the **Plugins\[Flow Name]\[version number]** directory included with the OpenNode2 installation files.

2. Create a new zip file containing the two Windsor.Node2008.WNOSPlugin.[Flow name].dll and .pdb files.

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

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

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

6. Locate and select the compressed RCRAInfo zip file you created in step 2 above.

7. Select the data exchange name "RCRA" that you created during the previous step from the **Exchange** dropdown box.

8. Click the **Upload** 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 RCRAInfo Data Service

Data services are distinct functions provided by a plugin to support a given data exchange. In the case of the RCRAInfo data exchange, there are 23 data services provided by the plugin:

- ClearPendingRCRASubmissions

- GetRCRASubmissionStatus

- ExecuteRCRAExtract

- RCRASubmitProxy (service created for unique installation)

  Extract and Submit Data

- RCRACMEExtractAndSubmission

- RCRACorrectiveActionExtractAndSubmission

- RCRAFinancialAssuranceExtractAndSubmission

- RCRAGeographicInformationExtractAndSubmission

- RCRAHandlerExtractAndSubmission

- RCRAPermitExtractAndSubmission

  Submit Data

- RCRAGetHazardousWasteCMEData

- RCRAGetHazardousWasteCorrectiveActionData

- RCRAGetFinancialAssuranceData

- RCRAGetGISData

- RCRAGetHazardousWasteHandlerData

- RCRAGetHazardousWastePermitData

  Process XML and Load into RCRA Staging Tables

- RCRASubmissionProcessor

- RCRAFinancialAssuranceSubmissionProcessor

- RCRAHazardousWasteCMESubmissionProcessor

- RCRAHazardousWasteCorrectiveActionSubmissionProcessor

- RCRAGISSubmissionProcessor

- RCRAHazardousWasteHandlerSubmissionProcessor

- RCRAHazardousWastePermitSubmissionProcessor

Note: Above is the comprehensive list of all services provided, however only the services for the submission type that you implement will need to be setup.

Any or all of the data services must be created and configured before they can be accessed through the OpenNode2 endpoints.

1. From the **Exchange** tab, locate the RCRA data exchange in the list of available exchanges.

2. Click the **Add Service** button located just beneath the RCRA exchange entry. The following page will be displayed to allow a new data service to be added.

3. In the **Name** field, type the name of the service that is being installed.

4. From the **Implementer** drop down box, select either the associated Implementer for the data service provided by the plugin[1].

   Note: When the implementer is selected, several arguments and data sources will appear. The Node Admin application will obtain these properties directly from the RCRAInfo plugin. The steps are the same for all of the data services.

---

[1] This assumes that the user is deploying the RCRAInfo data flow plugin provided freely with OpenNode2 by Windsor. If a different plugin is being deployed, then these configuration instructions will need to be modified as appropriate.

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. It is recommended that the service allow "Solicit".

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 refresh the page to display the relevant data entry fields as follows:

    i. In the argument labeled **Add Header**, enter *true* if the data service is to be used to provide data to the EPA FRS system or any other Exchange Network partner who may invoke this data service. Otherwise, enter *false*.

        Alternatively if a global variable has been set up to provide this value, check the **Use global value** checkbox and select the appropriate variable name from the drop down box that appears in place of the textbox.

    ii. In the argument labeled **Author**, type the name of the developer of the data service.

    iii. In the argument labeled **Contact Info**, type the name of the person who should be contacted regarding any submission created from the data service. Also include the person's email address and phone number. For example, enter *John Smith, (999) 999-9999, john@smith.com*, etc.

        Alternatively if a global variable has been set up to provide this value, check the **Use global value** checkbox and select the appropriate variable name from the drop down box that appears in place of the textbox.

    iv. The argument **NAAS User Mapping File Path** should be left blank.

    v. In the argument labeled **Notifications**, type the email addresses to which notifications should be sent by the receiving partner following processing of a file extract from this data service. If multiple addresses are entered they should be separated with a comma.

    vi. In the argument labeled **Organization**, type the name of the organization that is providing submissions created from the data service. For example, enter *Smith, Inc.*, etc.

        Alternatively if a global variable has been set up to provide this value, check the **Use global value** checkbox and select the appropriate variable name from the drop down box that appears in place of the textbox.

    vii. In the argument labeled **Payload Operation**, enter the appropriate Operation and Module parameter values. For example, enter *RCRA-Transactional/HD*. Please refer to the RCRA Flow Configuration Document for additional information about the Payload Operation Attribute, and allowed values (http://www.exchangenetwork.net/exchanges/waste/rcra.htm).

    viii. Set the **RCRAInfoStateCode** to the two character RCRAInfo state code for the data being submitted.

    ix. Set the **RCRAInfoUserID** to the three character RCRAInfo system user id with appropriate security to update the data for this State.

    x. The argument **Submission Partner Name** should be left blank. The submission partner will be configured in the Schedule instead.

        > **Important:** If a Submission Partner Name is populated, the plugin will also expect a NAAS User Mapping File Path to be supplied. The use of this functionality is out of scope for this document. If a NAAS User Mapping File Path is not supplied, the plugin will produce an error when executed.

    xi. In the argument labeled **Title**, type *GetRCRAData*.

xii.   In the argument labeled **Validate XML,** enter a value of "True" or "False."

8. Set the **SourceProvider** to the data source that connects to the RCRAInfo staging tables.

9. Click the **Save** button to save the service.

Repeat Steps 2 thru 9 if the other RCRAInfo Service is implemented.

The **Manage Exchanges** page for the RCRAInfo flow should now appear as follows if all services are installed:
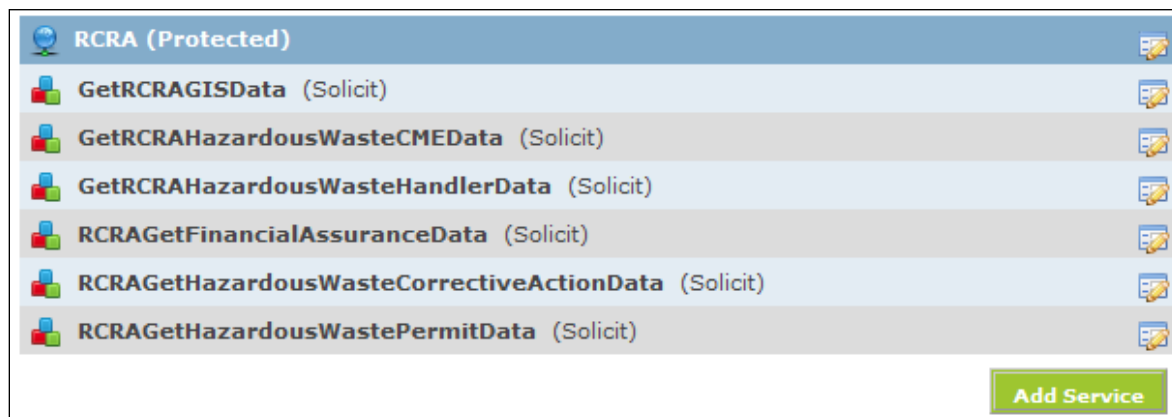
Define Data Exchange Schedules

Scheduled jobs can be configured to perform automated tasks such as submitting data to external partners or processing received files.

The RCRAInfo data exchange requires a schedule to initiate the submission of data from the staging tables to the EPA RCRAInfo database.

1. From the **Schedules** tab, click the **Add Schedule** button.

2. Type the name of the data service in the **Name** field depending on the service being scheduled.

3. Enable the schedule by clicking the **Active** checkbox.

4. Select "RCRA" from the **Exchange** dropdown list.

5. Set the start date to the first date when you wish the schedule to run. If the date is equal to the current or a past date, the schedule will execute immediately upon saving.

6. Set the end date to some point in the distant future.

7. Set the frequency to the data submission interval agreed between State program staff and the EPA Office of Solid Waste and Emergency Response (OSWER). Typically this is monthly.

8. In the **Data Source** area, check the radio button labeled **Results of local service execution**.

9. In the **From** dropdown box, select the appropriate data services value. This informs the schedule to use the selected RCRAInfo service as the data source for the submission.

10. In the **Result Process** area, check the radio button labeled **Submit result to an Exchange Network partner.**

11. Select the value corresponding to the name of the EPA CDX Node endpoint from the **To** dropdown box.

> **Note:** In order to receive notification emails from CDX, you must specify a CDX Node v1.1 endpoint. This is due to how notifications are handled differently at CDX for v1.1 and v2.0 endpoints. OpenNode2 does not use the NotificationURI submit parameter in Node 2.0. Instead, the NotificationURI parameter in the XML Header is used which requires submitting to a v1.1 endpoint in order to work. This is behavior is described in the official RCRA FCD.

12. In the **Exchange** textbox, type "RCRA".

13. In the **Operation** textbox, type "default". The field will not be displayed for exchange endpoints before version 2.0.

14. Click the **Save** button to save the schedule.

Contact CDX to Establish Exchange Settings

Contact the EPA CDX Node helpdesk and ask them to perform the following tasks:

1. Authorize the OpenNode2 runtime (operator) NAAS account to submit to the RCRAInfo data exchange on the EPA systems.

2. Map the OpenNode2 runtime NAAS account to the CDX Web user account that currently administers EPA RCRAInfo data for the organization.

Set Up 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 RCRAInfo data exchange. Please see the Node Administration Guide for more information on setting up notifications.

Monitor Flow Activity

The OpenNode2 will track all RCRAInfo 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: Staging Table Diagrams

The diagrams below show the relationship between the major RCRA schema components and their corresponding OpenNode2 staging table name.

Handler (HD)
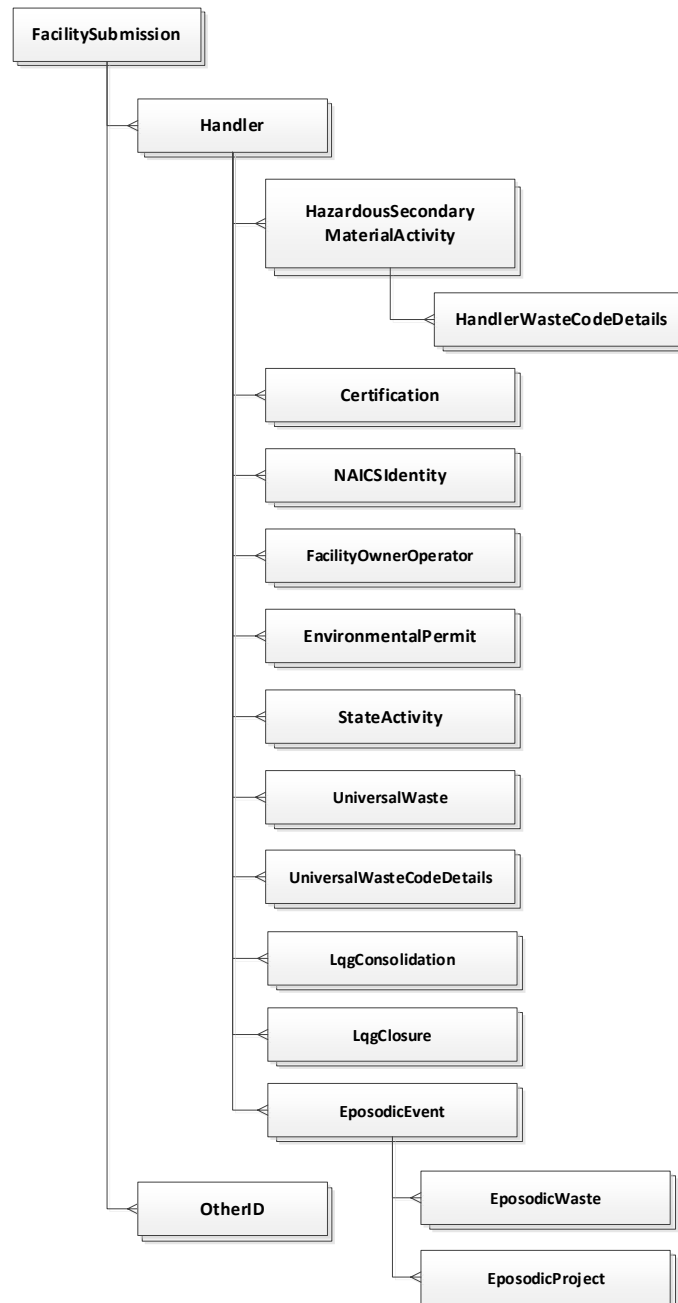
Compliance Monitoring and Enforcement (CME)

Financial Assurance (FA)

```
┌─────────────────────┐
│ RCRA_FA_SUBM        │
└─────────────────────┘
         │
         │  ┌──────────────────────────┐
         └──│ FAFacilitySubmission     │
            │ RCRA_FA_FAC_SUBM         │
            └──────────────────────────┘
                     │
                     │  ┌──────────────────────────┐
                     ├──│ CostEstimate             │
                     │  │ RCRA_FA_COST_EST         │
                     │  └──────────────────────────┘
                     │           │
                     │           │  ┌──────────────────────────┐
                     │           └──│ CostEstimate             │
                     │              │ RelatedMechanism         │
                     │              │ RCRA_FA_COST_EST         │
                     │              │ _REL_MECHANISM           │
                     │              └──────────────────────────┘
                     │
                     │  ┌──────────────────────────┐
                     └──│ Mechanism                │
                        │ RCRA_FA_MECHANISM        │
                        └──────────────────────────┘
                                 │
                                 │  ┌──────────────────────────┐
                                 └──│ MechanismDetail          │
                                    │ RCRA_FA_MECHANISM        │
                                    │ _DETAIL                  │
                                    └──────────────────────────┘
```
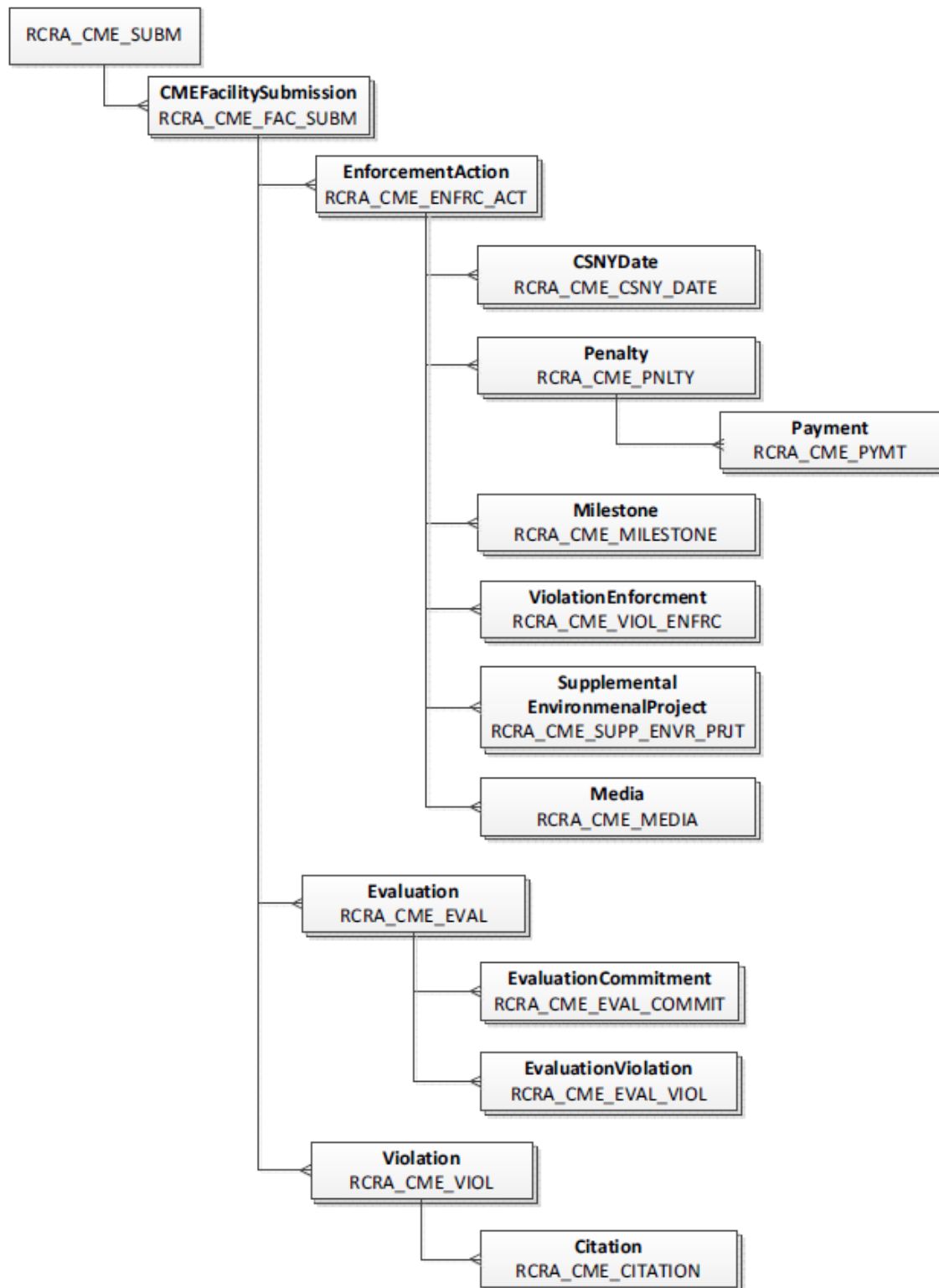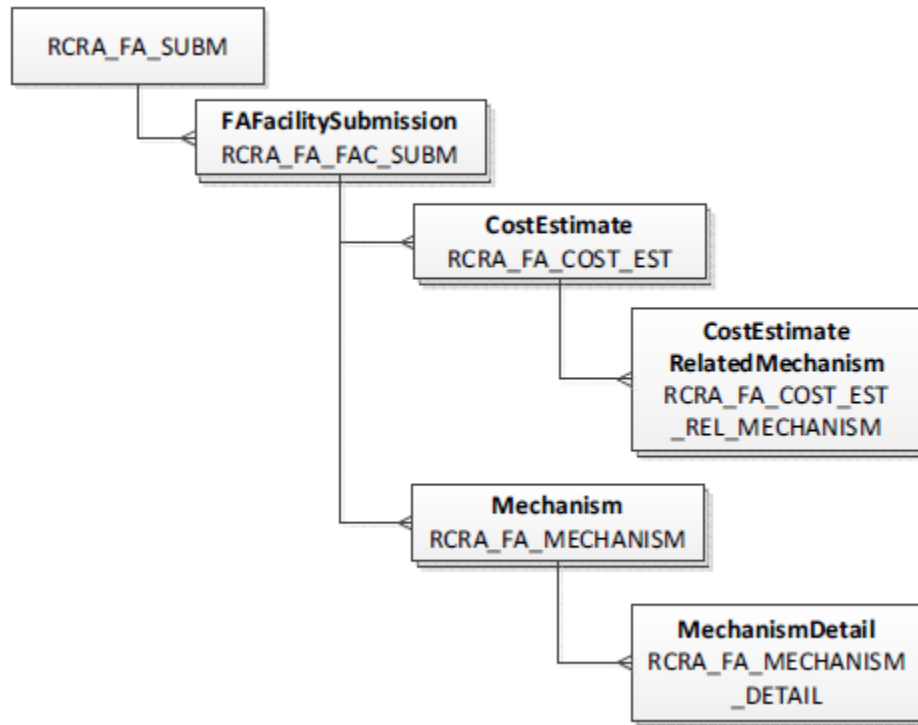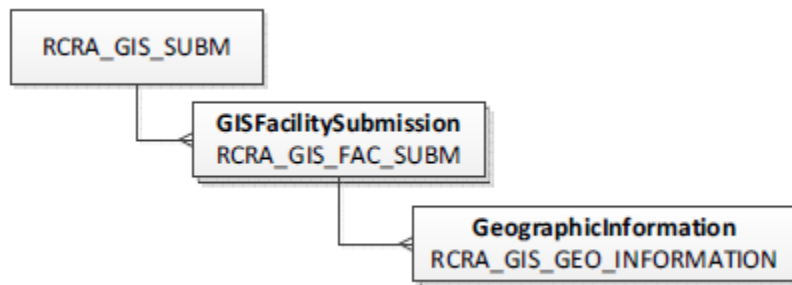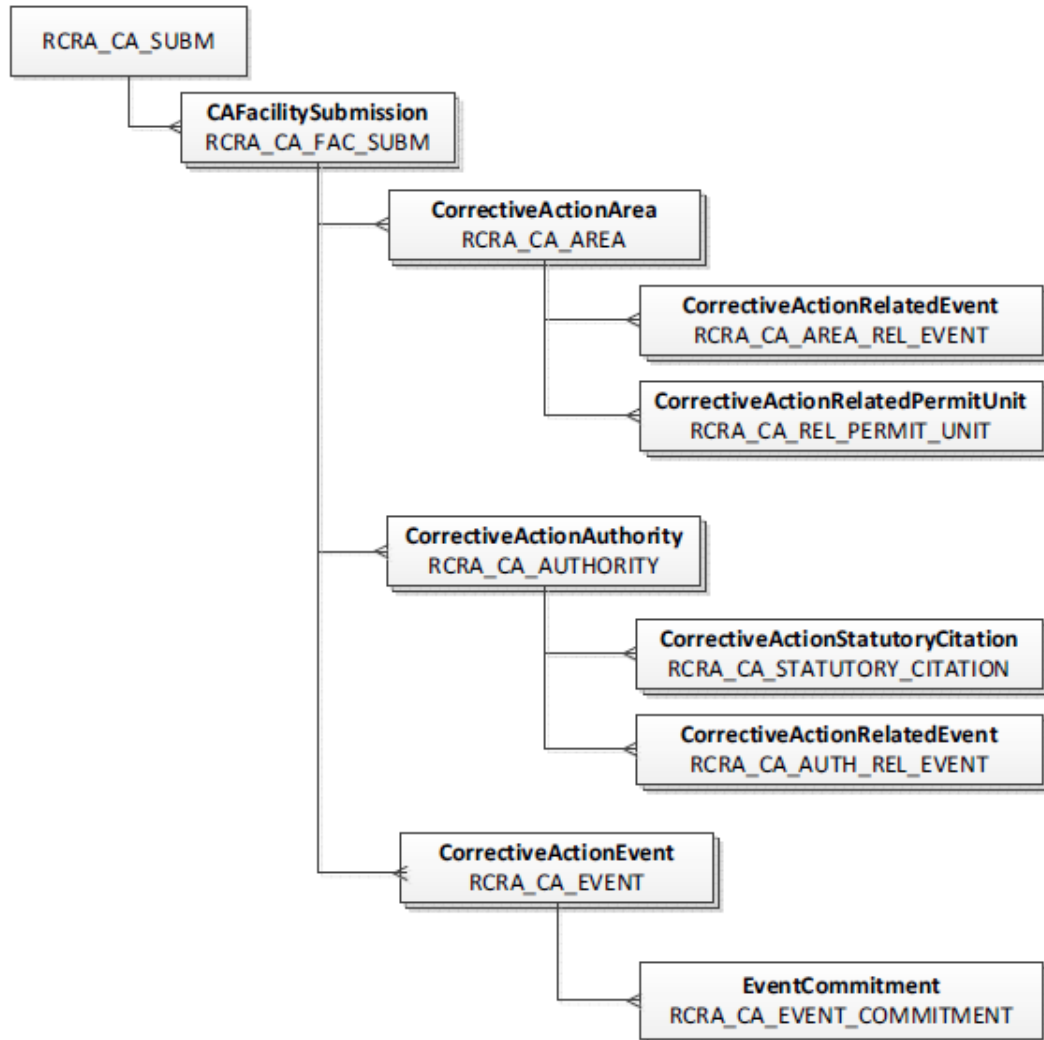
Geospatial Information (GIS)

```
┌─────────────────────┐
│ RCRA_GIS_SUBM       │
└─────────────────────┘
         │
         │  ┌──────────────────────────┐
         └──│ GISFacilitySubmission    │
            │ RCRA_GIS_FAC_SUBM        │
            └──────────────────────────┘
                     │
                     │  ┌──────────────────────────┐
                     └──│ GeographicInformation    │
                        │ RCRA_GIS_GEO_INFORMATION │
                        └──────────────────────────┘
```

Corrective Action (CA)

Permitting (PM)