

**RCRAInfo Network Exchange**

Flow Configuration Document

Version: 5.11

Revision Date: July 30, 2021

THIS PAGE INTENTIONALLY LEFT BLANK

**Table of Contents**

[1 Introduction 4](#_Toc272847654)

[1.1 Flow Identification 4](#_Toc272847655)

[1.2 Background 5](#_Toc272847656)

[1.3 Flow Configuration Document Scope 5](#_Toc272847657)

[1.4 Data Flow Overview 6](#_Toc272847658)

[1.5 Flow Access and Security 6](#_Toc272847659)

[1.6 Flow-level Business Rules 6](#_Toc272847660)

[1.7 Additional Flow Tools and Resources 7](#_Toc272847661)

[2 Submission Composition 8](#_Toc272847662)

[2.1 Implementation of the Header/Payload for the RCRAInfo Network Exchange 8](#_Toc272847663)

[2.2 Payload Operations 12](#_Toc272847664)

[3 Data Processing 15](#_Toc272847665)

[3.1 Submit Processing to EPA CDX 15](#_Toc272847666)

[3.2 Submission Processing and Feedback 18](#_Toc272847667)

[4 Data Access 20](#_Toc272847668)

[4.1 Data Access Services using the Exchange Network 20](#_Toc272847669)

[4.2 Data Publishing Services using REST 23](#_Toc272847670)

[5 Data Validation 25](#_Toc272847671)

[5.1 Understanding the RCRAInfo Schemas 25](#_Toc272847672)

[5.2 Schema Implementation 33](#_Toc272847673)

[Appendix A - RCRAInfo Flow Implementation and Testing Checklist 34](#_Toc272847674)

[Submission Test Cases 36](#_Toc272847675)

[Appendix B – RCRAInfo Data Access Services 38](#_Toc272847676)

**Component Alignment**

**Flow Component Versions Currently Supported**

|  |  |  |
| --- | --- | --- |
| Component | Version(s) Supported | Explanation (optional) |
| FCD | **5.0, 5.1, 5.2** | **Replacement of previous versions. Minor update.** |
|  |  |  |
| Schema | **5.0, 5.1, 5.2** | **Replacement of previous versions. Minor update.** |
| DET | **5.0, 5.1, 5.2** | **Replacement of previous versions. Minor update.** |
| RCRA | **5.0, 5.1, 5.2** | **Submit, Query, Solicit data services. Minor update.** |
| RCRA | **5.3** | **Replacement of previous version. Minor update.** |
| RCRA | **5.4** | **Replacement of previous version. Minor update.** |
| RCRA | **5.5** | **Replacement of previous version. Minor update.** |
| RCRA | **5.6** | **Replacement of previous version. Minor update.** |
| RCRA | **5.7** | **Replacement of previous version. Adding new module (e-Manifest) for outbound services.** |
| RCRA | **5.8** | **Replacement of previous version. Minor update.** |
| RCRA | **5.9** | **Replacement of previous version. Minor update. Add new service GetCEDeletedDataByState** |
| RCRA | **5.10** | **Replacement of previous version. Minor update.** |
| RCRA | **5.11** | **Replacement of previous version. Minor update.** |

# Introduction

## Flow Identification

**Flow Name:**

RCRAInfo

**Flow Owner:**

David Charbonneau, Chief

USEPA / Information Collection and Analysis Branch Office of Resource Conservation and Recovery

**Flow Owner Contact Information:**

Jimmy Yin

USEPA/ Information Collection and Analysis Branch Office of Resource Conservation and Recovery

Phone: (703) 347-0273

Email: [yin.jianchu@epa.gov](mailto:yin.jianchu@epa.gov)

## Background

RCRAInfo is an information system constructed and maintained by EPA to support the national hazardous waste program as defined by the Resource Conservation and Recovery Act (RCRA). The system is used by States and Regional/Headquarter EPA, to gain insight into the management of the hazardous waste program at both the state and national level.

In recent years RCRA data management tools have undergone a modernization effort, with user interfaces being upgraded to support web-based data entry forms with robust data validation capabilities. Web based reporting interfaces have also been constructed to support the States in the implementation of the program (implementers). Databases have been migrated to a relational Oracle database with associated technical architecture, allowing States to query the database directly to support the management of the program. An XML based translation module with robust data validation rules has been implemented allowing States the ability to electronically submit their hazardous waste data to RCRAInfo.

States interact with RCRAInfo in a variety of ways depending upon the implementation of the individual programs and the maturity of their IT infrastructure. Some States use RCRAInfo as their primary hazardous waste information system, performing data entry directly into the national system. Some States perform dual data entry into RCRAInfo and a corresponding state system that manages RCRA data at the state level. The remaining States rely entirely on their state system and periodically submit data to RCRAInfo for incorporation of the data into the national system.

## Flow Configuration Document Scope

This revision of the RCRAInfo Exchange Network Flow Configuration Document (FCD) is intended to bring the Exchange Network up to date with the Version 6 changes for all modules within RCRAInfo.

The modifications to RCRAInfo were only minor and subsequently the data flow will only increment to Version 5.10. The changes focus on edits for existing xsds for Financial Assurance services. The changes can be found in the schema change log. However, there are required FA fields in this release.

*Appendix A – RCRAInfo Network Exchange Implementation and Testing Checklist*:This appendix presents Trading Partners with a checklist of decisions for consideration, flow configuration actions and testing steps necessary to submit data to RCRAInfo using the Exchange Network.

## Data Flow Overview

The RCRAInfo Network Exchange provides data services that can be used by trading partners to submit data to EPA’s RCRAInfo system. Data is submitted by authorized States to EPA to meet program requirements. The data flow supports a series of transaction models for interacting with RCRAInfo and is equivalent in functionality and quality assurance business rules used by many States. Data access services are also available. These outbound services provide access to data in RCRAInfo following the same schema as the inbound submit services.

The RCRAInfo data services are provided through the Exchange Network ([www.exchangenetwork.net](http://www.exchangenetwork.net)) and are accessed through EPA’s Central Data Exchange Node (the “CDX node”) using the Exchange Network’s node web service specifications. Version 5.11 of the RCRAInfo data services supports version 2.0 of the Exchange Network’s node web service specifications. Version 5.11 of the RCRAInfo data services are also backward compatible with version 1.1 of the Exchange Network’s node web service specifications; however, it is strongly encouraged that trading partners utilize Exchange Network node client or node technology that supports version 2.0 of the specifications.

The RCRAInfo Network Exchange is not backwards compatible with prior major schema versions, and will only support version 5.9 or greater of the schema and this FCD. Partners who have already mapped to version 5.9, and 5.10 will still be able to submit data to version 5.11 barely being impacted by this change.

## Flow Access and Security

All service requests must be accompanied by a valid NAAS security token per the Exchange Network’s Node specifications. All partners must be authorized to NAAS and receive a valid security token before any of the RCRAInfo data services can be invoked.

If partners choose to use direct NAAS authentication, Node 1.1 implementations must authenticate against NAAS 2.0. For Node 2.0 implementations, users must authenticate against NAAS 3.0. Alternatively, partners may choose to use delegated authentication, passing a username and credential to the CDX node. In this case, the CDX Node will automatically authenticate against the correct NAAS version endpoint.

In addition to having a valid NAAS account, the submitter must request that CDX authorize the NAAS account to invoke the Submit, GetStatus and Download operations for the RCRA data exchange. Furthermore, the submitter must request that CDX pair the user’s NAAS account with the RCRAInfo User ID that will be provided in the header of the submission.

In addition, RCRAInfo requires a valid user ID with associated permissions to transact in the system. Permission is granted at the module level (e.g., Handler, CME) and correlates to the areas of the program for which the State has authorization. The RCRAInfo User ID is passed to CDX as a value in the XML submission file. See the Header/Payload discussion in Section 2 of this document for more information.

## Flow-level Business Rules

**Current Business Rules:** The Data Exchange Template (DET) contains the list of data elements along with their respective business rules as defined in the RCRAInfo *DED*. It is recommended that the user familiarize themselves with the *DED* and then use the DET as a quick reference to understand the RCRAInfo XML structure, and to understand the business rules that are being applied.

**Fault Follow-up Actions:** There are two primary failure points in the RCRAInfo data flow; 1) receipt by the CDX node with associated schema validation and 2) loading and validating the data within the RCRAInfo data system. In either case in the event of an error condition, the data can be resubmitted for processing. The section titled *Submission Processing and Feedback*, in this document discusses error processing and messaging in more detail.

## Additional Flow Tools and Resources

This FCD is intended to define the supported data services, as well as the approaches and processes that are used to exchange information. This FCD is intended to be used in conjunction with the following support documents:

*RCRAInfo Documentation / DED*

RCRAInfo has retired the flat file translation process.

**It is critical that Partners embarking on translation to RCRAInfo through the Exchange Network familiarize themselves with the *DED***. All RCRAInfo data sent through the Exchange Network must meet the data quality rules specified in the *DED*.

The Data Exchange Template (DET) provides all of the relevant schema data elements for each of the RCRAInfo modules. EPA has also included all of the business rules from the *DED* as they apply to Exchange Network Partners. The DET is designed to make submittal of data to RCRAInfo easier and more understandable for Exchange Network Partners.

This document and all other relevant documentation can be found at the Exchange Network website (<http://www.exchangenetwork.net/data-exchange/rcrainfo/>).

# Submission Composition

## Implementation of the Header/Payload for the RCRAInfo Network Exchange

### Overview

The RCRAInfo Network Exchange will support a document structure consisting of a single header with a single payload. The RCRAInfo XML schemas have been designed to support processing at the module level. As a result, each separate submission will constitute a processing operation for a complete module of RCRAInfo.

The RCRAInfo Network Exchange continues to use Header v0.9 for both Node v1.1 and Node v2.0 implementations. While the new Header v2.0 is intended to be used by all Node v2.0 exchanges, the existing Header has been retained for consistency.

### Header/Payload Relationship

[The Exchange Network Frequently Asked Questions](http://test.epacdxnode.net/faq/ch03s12.html) provides the following explanation of the header and payload relationship:

“The document header provides information to identify the contents of a data payload. It was developed to further automate the data exchange process so that data can be more readily identified during transport and at its processing destination…

The document header can describe what a data payload contains, who submitted it, when it was submitted, as well as instructions on processing the payload contents, such as whether the contents are additions, deletions, or updates. The header is independent of payload contents, so no data schema changes are necessary…”

The header serves as a wrapper to the individual XML instance documents (payloads). It is used to describe the document, providing basic metadata for the submission.

The following diagram describes the basic Exchange Network Document Structure and the relationship of the header to payload.

Exchange Network Document

Header v0.9

Exchange Network Document

Header v0.9

Header v0.9

Payload

Operation – RCRA-Transactional|HD

Payload

Operation – RCRA-Transactional|CE

Submission #2

Submission #1

Figure 1 – RCRA Submission Format Diagram

Table 1 describes the document Header elements and how they are utilized for the purpose of RCRAInfo submissions.

Table 1

| **Header Element** | **Description** | **Example Value** | **Required** | **Notes** |
| --- | --- | --- | --- | --- |
| Author | First and Last Name of Individual Generating the XML Document. | John Smith | Yes | Reference: not used directly by either the CDX or RCRAInfo processes.  For the purposes of the RCRAInfo Network Exchange, this is the submitter or responsible person contact. |
| Organization | Name of company or environmental agency or individual generating the XML document. | State X Department of Environmental Quality | Yes | Reference. |
| Title | Type of Submission | RCRAInfo Submission | Yes | Reference to the flow. |
| Creation Time | Date/Time when the document was generated. | 2017-01-01T12:12:12 | Yes | Used by CDX Processes for meeting RCRAInfo submission requirements. |
| Comments | Open text |  | No | Reference |
| Data Service | Name of Service Request. |  | No | This component is not used for data submissions. |
| Contact Info | Area Code and Telephone Number, e-mail address of contact (author). | 555-555-5555, [Joe@deq.statex.gov](mailto:Joe@deq.statex.gov) | Yes | Reference |
| Notification | A URI where result/report can be sent. | [Joe@deq.statex.gov](mailto:Joe@deq.statex.gov) | No | This component is not used for data submissions. Use the Property Name/Value pairs described below for notifications. |
| Sensitivity | Level of document sensitivity. | Unclassified, Confidential | No | Reference |
| **Property** |  |  |  |  |
| *Name = RCRAInfoUserID*  *Value = (UserID)* | Name Value Pair, representing the RCRAInfo User ID to which RCRAInfo system rights and audit trails are tied. | ABC or  Smith.[Joe@deq.statex.gov](mailto:Joe@deq.statex.gov) | Yes | Required by both RCRAInfo and the CDX processes.  See *Appendix A - RCRAInfo Flow Implementation and Testing Checklist f*or specifics on obtaining and authorizing this User ID. |
| *Name = RCRAInfoStateCode*  *Value =  (StateID)* | Name Value Pair, two character RCRAInfo state code. | MI | Yes | Required by both RCRAInfo and the CDX processes. |
| *Name = NotificationURI*  *Value =*  *(email address)* | Name/Value pair, indicating where automatic email notifications should be sent when RCRAInfo processing is complete. | [Joe@deq.statex.gov](mailto:Joe@deq.statex.gov) | No | This should be used for Node v1.1 submissions only. Node 2.0 submissions should use the NotificationURI construct in the Submit operation to specify email recipients. |
| Payload Operation Attribute | Operation to be performed on the payload. | **Operation|Module** *Operation Parameter*  RCRA-Transactional,  *Module Parameter* HD=Handler CE=CME  PM=Permitting  CA=Corrective Action  FA=Financial Assurance  GS=GIS  Delimiter: Pipe (|)  *Example*  RCRA-Transactional|HD | Yes | Required by both the CDX processes and RCRAInfo.  There is one payload operation per payload. There may be only one payload per submission |
| Payload XML Document | The root element of XML instance file contained in the payload must match the root element in one of the following schemas:  RCRA\_HazardousWasteCMESubmission\_v5.11.xsd  RCRA\_HazardousWasteHandlerSubmission\_v5.11.xsd  RCRA\_HazardousWastePermitSubmission\_v5.11.xsd  RCRA\_HazardousWasteCorrectiveActionSubmission\_v5.11.xsd  RCRA\_FinancialAssuranceSubmission\_v5.11.xsd  RCRA\_GeographicInformaitonSubmission\_v5.11.xsd  RCRA\_HazardousWasteReportUniv\_v5.11.xsd  RCRA\_HazardousWasteEmanifests\_v5.11.xsd | | Yes |  |

## 

## Payload Operations

The payload operation attribute is used to denote the module processing for submissions. There is one acceptable value: **RCRA-Transactional.** For a list of allowable operations by module, see Table 2.

**Table 2. Allowable Operations by Module**

|  |  |  |
| --- | --- | --- |
| **Module Parameter** | **Module Name** | **Operations Supported** |
| HD | Handler | RCRA-Transactional |
| CE | Compliance Monitoring and Enforcement | RCRA-Transactional |
| PM | Permitting | RCRA-Transactional |
| CA | Corrective Action | RCRA-Transactional |
| GS | GIS | RCRA-Transactional |
| FA | Financial Assurance | RCRA-Transactional |

For a more detailed discussion of the processing from CDX to RCRAInfo loading please see the discussion *Submission Processing and Feedback* in this document.

The following sections describe the payload operations that the CDX processes will use to derive the data processing mode supported by RCRAInfo. In addition, this discussion outlines the challenges that may be encountered with each operation for the implementer’s consideration in selecting and configuring their preferred payload operation.

### Payload Operation: Transactional (RCRA-Transactional| HD, CE, CA, PM, FA, GS)

#### Overview

RCRAInfo translation processes have been developed to support transactional record processing, where implementers can add/update or delete records from a table, as identified by the primary keys for the table. The transactional model of submission processing is supported for the Handler, CME, Permitting, Corrective Action, Financial Assurance, and GIS modules of RCRAInfo.

The issues outlined in the supporting document *RCRAInfo Data Submission Overview and Challenges,* addressing authorized users and edits to primary keys, have an impact on transactional processing. A RCRAInfo authorized user is a RCRAInfo registered user with proper permission. To translate data, a user must have at least *Delete* permission for the module they want to translate. Authorized users are advised to consider this mode only if they have a clear understanding of the processing and have the mechanisms necessary to maintain the synchronization of the data between the State and EPA systems.

#### RCRAInfo Processing

The transaction type implemented in RCRAInfo is affected based upon the transaction code submitted for the record within the XML instance document.

Valid transaction codes for this payload operation are:

* A (Add/Update)
* D (Delete)

When a transaction code of A is sent for a record in RCRAInfo, the process will attempt to match the record based on the primary keys for the table. If a match is made and the partner has the proper permission for the record, it is updated with the contents of the submission. Element level changes are not supported; as a result, the record is, in effect, replaced.

If the match is made and the translator does not have proper permission for the record, a critical error will be raised, and the entire submission will be unsuccessful.

If a match on the table’s primary keys is not made and the translator has proper permission, then the translation routines recognize the record as new, and inserts it into the database.

When a transaction code of D is sent for a record, the RCRAInfo routines will attempt to match the record based on the primary keys for the table. If a match is made and and the translator has proper permission, then the record will be deleted. There are several notable circumstances associated with deletions that affect the success of some delete transactions:

* Cascade deletes: RCRAInfo supports cascade deletes and implements a specific deletion hierarchy by module. The reader is encouraged to review the deletion hierarchy outlined in the translator specifications. The section titled *Schema User’s Guide* in this FCDalso contains a diagram of the deletion hierarchy for the modules.
  + As an example of the deletion hierarchy, a delete transaction for an Enforcement Action record belonging to the implementer would result in the deletion of not only the action record but also any linking records (Enforcement/Violation) for any associated Violations.
* Child Table Delete Transactions: Since RCRAInfo supports cascade deletion of child records, the submission of delete transactions for child data is not needed.
  + Submission of a delete transaction for a record that has previously been deleted through a cascade delete will raise a critical error.
* Translator and Cascade Deletes: To successfully delete a record and any associated child data, the submitting implementer must have proper permission on that module.
  + Example A: if a delete transaction is submitted for a violation record, which has associated EPA owned Enforcement Actions, the request will fail.
  + Example B: if EPA owns the Enforcement action records and the linking records in the Compliance Schedule table, any attempt to cascade delete the linking records owned by EPA would fail, as would the requested deletion of the Violation.

Please see section *Schema User’s Guide* in this FCDfor additional guidance on this processing approach

# Data Processing

RCRAInfo supports Submit, Query, and Solicit services. This section describes how to interact with these services. The Submit service is used to send data to EPA for the batch loading of data into RCRAInfo. Query services provide specific functions or small amounts of data. Solicit services are for syncing partner databases with RCRAInfo.

## Submit Processing to EPA CDX

The RCRAInfo Network Exchange provides a mechanism for submitting data to EPA. The specific steps to perform this operation are described in this section, including a description of back-end processing at EPA and methods for the submitter to retrieve processing results and diagnose error conditions.

### Submitting Data to EPA using “Submit”

**Type:** Submit

**Data Service-level Business Rules:** Please see Section 2: Payload Operations for a discussion of how each submission type is specified and the specific rules for each.

**XML Header Usage:** Please see Section 1 of this document for information on how the header must be implemented.

#### Request

The Submit request must be formulated as follows:

**dataflow Parameter:** Must be set to **RCRA**

**flowOperation Parameter** (Node 2.0 only)**:** Can be left empty or set to **default**

**recipients Parameter** (Node 2.0 only)**:** Not applicable. Recipient information, if provided, will be ignored by EPA CDX.

**notificationURI Parameter** (Node 2.0 only)**:** Used to optionally specify an email address that should receive a notification when processing is complete at CDX. Node v1.1 submission should use the Property Name/Value pairs specified in the header in order to receive notifications since this feature is not supported in the Node v1.1 specification. See the Header/Payload discussion in this document for more information on email notifications for Node v1.1 submission.

Per the Node v2.0 specification, the notificationURI parameter allows the requestor to optionally include a notification type. The notification type is intended to only send notifications for specific types of events such as errors or warnings. Please note that CDX will ignore any value specified in this attribute. Notifications will be sent to the email address specified, regardless of success or failure.

**Documents Parameter:** Must contain a single ExchangeNetworkDocument whose content consists of an Exchange Network Header v0.9 with a single payload. The payload must be an XML document conforming to the RCRA v5.11 XML Schema. The root element of the instance file must match the root element from one of the RCRA\_HazardousWasteCMESubmission\_v5.11.xsd, RCRA\_HazardousWasteHandlerSubmission\_v5.11.xsd, RCRA\_HazardousWasteCorrectiveAction\_v5.11.xsd, RCRA\_HazardousWastePermitSubmission\_v5.11.xsd, RCRA\_FinancialAssuranceSubmission\_v5.11.xsd, or RCRA\_GeographicInformationSubmission\_v5.11.xsd schema.

The document should be compressed in ZIP format.

#### Response

The response returned by the CDX Node will contain the following elements:

**transactionId**: Per the Node v1.1 and v2.0 specifications, a Transaction ID will be returned. The element name for Node v1.1 exchanges will be “return” whereas the element name for Node v2.0 implementations will be “transactionId”. The content will be the same in either case.

**status** **and statusDetail** (Node 2.0 only): For Node v2.0 submissions, the response will also contain a **status** and **statusDetail** element. The status information will not immediately indicate success or failure as submission processing is an asynchronous process that will likely take a period of time to complete.

**Error Conditions and Return:** No flow-specific errors are defined for the submit process of the RCRAInfo Exchange. SOAP exceptions, timeouts, and other low-level errors may occur, but are out of scope for this document.

### Determining Submission Processing Results using “GetStatus”

**Type:** GetStatus

**Data Service-level Business Rules:** None

#### Request

The GetStatus request must be formulated as follows:

**transactionID Parameter**: For both Node v1.1 and Node v2.0 exchanges, the GetStatus operation requires a Transaction ID to be provided. The Transaction ID should be that which was returned by EPA CDX in the Submit Response in the step above.

#### Response

The response returned by the CDX Node will contain the following elements:

**transactionId** (node 2.0 only): Node v2.0 implementations will echo back the same Transaction ID provided in the request. There is no equivalent in Node v1.1 implementation.

**status**: A textual description of the processing status will be returned. Please see the Submission Processing and Feedback section of this document for a description of available responses and the meaning of each.

Node v1.1 implementation will simply return a string (named “return”) containing the status text (e.g. “Completed”, “Failed”). Node v2.0 implementations will return a Transaction Status Code and Status Detail. The Status Detail in this case is the status text (e.g., “Completed”, “Failed”).

**Error Conditions and Return:** No RCRAInfo Exchange specific errors have been defined.

### Retrieving Submission Processing Reports using “Download”

**Type:** Download

**Data Service-level Business Rules:** None

#### Request

The Download request must be formulated as follows:

**dataflow Parameter**: Must be set to **RCRA**

**transactionID Parameter**: For both Node v1.1 and Node v2.0 exchanges, the Download operation requires a Transaction ID to be provided. The Transaction ID should be that which was returned by EPA CDX in the Submit Response in the step above.

**documents Parameter**: Should be left null. All associated documents will be returned.

## Submission Processing and Feedback



Figure 2 – Submission Processing Diagram

### Submission Processing

Submission of RCRAInfo Exchange Network Documents to the CDX node follows the processing steps outlined above. (See Figure 2):

* State Node will initiate a Submit action

It is critical that the proper Node/Database be addressed to ensure that data is loaded into the correct database environment.

* + If the State addresses the CDX test Node for the submission, the data will be loaded into the RCRAInfo Pre-Production database.
  + If the State addresses the CDX production Node for the submission, the data will be loaded into the RCRAInfo Production database.
* CDX node receives the documents.
  + The RCRAInfo User ID is critical to the processing of the submission. The RCRAInfo security rights and record auditing are derived from the ID. Therefore, the User ID specified in the Header file must have rights to transact within the modules being submitted. The CDX processes will derive the User ID from the header property name/value pair: *RCRAInfoUserID.*
* Upon completion of the staging table load process, the RCRAInfo data validation process is triggered.
  + RCRAInfo data processing requires that all errors raised by the validation routines for a module (e.g., Handler) must be resolved prior to promoting the staged data to the RCRAInfo data tables.
  + If there are errors, they must be corrected, and the data for the offending module must be resubmitted.
* Upon successful validation of the submission, the staged data is automatically promoted into the RCRAInfo data tables.

The submitting Partner may check the status of the submission to CDX with the GetStatus method. As Figure 2 indicates there are several potential points of process termination (Failed, Completed), after which submitting Partner may issue a Download request with the submission transaction ID to access the available processing documents for the submission.

Alternatively, it is possible to receive automatic notifications. For Node v1.1 implementation, the Partner may provide an email address in the name/value pairs in the Property element of the header. See the Header/Payload discussion in this document for more information. For 2.0 Nodes the submitter may optionally provide an email addresse(s) in the NotificationURI fields of the Node 2.0 web service request.

The available documents are:

* Original Submission: zipped XML submission made by Partner.
* Schema Validation Report: Validation Results document, text document from QA Server
* RCRAStagingStatusReport\_: Report generated as a result of the RCRAInfo data validation processing; based on the rules outlined in the DED.

# Data Access

## Data Access Services using the Exchange Network

The RCRAInfo data flow provides the ability for Exchange Network (EN) partner nodes (i.e., node clients and full nodes) to request and receive RCRA data in XML payloads. This flow is implemented according to EN practices and recommendations via Query and Solicit web services. In both service exchanges, RCRAInfo will provide either public or confidential data. If the requesting NAAS user has a valid RCRAInfo account association, confidential data will be provided; in all other cases the RCRAInfo will provide only public data to the requestor. Both Query and Solicit support both node functional specifications 1.1 and 2.0. All data returned via these services follows the RCRAInfo schema version 5.11.

### Registration / Authorization

Users of the Exchange Network RCRAInfo services must be registered with NAAS. The users must have NAAS policies to perform RCRA Query and Solicit web services. Privileges will be granted by the CDX Node HelpDesk/Administrator. Some users may have a RCRAInfo account. When supplied, the CDX node administrator will map the NAAS user identifier to the RCRAInfo user account.

All user requests will be authenticated via NAAS by the CDX node. Authorization will be also completed via NAAS. The CDX node will perform NAAS user account to RCRAInfo user account mapping and the RCRAInfo userid will be passed to the RCRAInfo node via the NAAS security token to be used for authorization purposes. If the RCRAInfo userid is valid, then RCRAInfo will return all available data. Otherwise, RCRAInfo will return only publicly available data.

#### Request

**Result Documents:** Must contain a single ExchangeNetworkDocument whose content consists of an Exchange Network Header v0.9 with a single payload. The payload must be an XML document conforming to the RCRA v5.11 XML Schema. The root element of the instance file must match the root element from the RCRA\_HazardousWasteReportUniv\_v5.11.xsd or RCRA\_HazardousWasteEmanifests\_v5.11.xsd schema. The document should be compressed in ZIP format.

### Query

For small data sets, a synchronous data retrieval model will be used. Node requestors (i.e., node clients and full nodes) will execute Query web services against the CDX Node. These Query requests will be synchronously processed by Central Data Exchange (CDX) and RCRAInfo. An overview of this data exchange is shown in the figure below.

The general structure of a Query request:

**Security Token:** valid NAAS security token

**Dataflow:** RCRA

**Request:** See Appendix B for service details

**Row Id:** 0

**Max Rows:** -1

**Parameters:** See Appendix B for parameter details

If error occurs CDX Node will return following errors:

• User is not authenticated

• User is not authorized to perform RCRA Query operation

• Invalid service request

• Invalid service parameters

• User is not authorized to obtain specific confidential data

• CDX Node system error

Notes:

• Mandatory parameters must always be passed

• 2 options in passing Node 1.1 empty optional parameters

o No parameter value

o Provide an empty string value

• 2 options in passing Node 2.0 empty optional parameters

o no key and no value

o key and empty string value



**Figure 3 Synchronous RCRA Publishing Data Exchange (Query)**

### Solicit

For large data sets an asynchronous data retrieval model will be used. Node requestors (*i.e.,* node clients and full nodes) will execute Solicit web services against the CDX node to request RCRA data. The Solicit requests will be asynchronously processed by CDX and RCRAInfo. An overview of this data exchange is show in the figure below.

The general structure of a Solicit request:

**Security Token:** valid NAAS security token

**Dataflow:** RCRA

**Request:** See appendix B for service definitions

**Recipient:** If requesting node would like result submitted to their node, endpoint URL is provided, otherwise empty string should be provided.

**NotificationURI:** If requesting node would like email notification, email addresses should be provided, otherwise empty string should be provided.

**Parameters:** See appendix B for service definitions



**Figure 4 RCRA Data Exchange (Solicit Request)**

#### Determining Solicit Status using GetStatus

If GetStatus is called and the status response is COMPLETED, no status detail is available.

If GetStatus is called and the status response is FAILED, the following are possible status details.

|  |  |
| --- | --- |
| **statusDetails Value** | **Description** |
| E\_ServiceUnavailable: Invalid service called | Requesting a service that is not defined on the RCRA Node. |
| E\_InvalidRCRAInfoAccount: NAAS account is mapped to an invalid RCRAInfo User account | If an invalid RCRAInfo user account was provided and the request is for private data. |
| E\_AccessDenied: RCRAInfo User does not have access to obtain <parameter> data | If a state user is trying to obtain data outside of their state, or if a regional user is trying to obtain data outside of their region. |
| E\_InvalidParameter: Invalid parameter value for handlerId | If the required handler id parameter is invalid. |
| E\_InvalidParameter: Invalid parameter value for state | If the required state parameter is invalid. |
| E\_XMLError: error generating xml file | If there is an error in generating the xml file. |
| E\_DBMSError: error connecting to RCRAInfo Database | If there is an error connecting to the database. |
| E\_ZippingError: error zipping xml response | If there is an error in zipping the xml. |
| E\_ArchiveError: error saving result | If there is an error archiving zip file. |

Notes:

* Mandatory parameters must always be passed
* 2 options in passing Node 1.1 empty optional parameters
  + No parameter value
  + Provide an empty string value
* 2 options in passing Node 2.0 empty optional parameters
  + no key and no value
  + key and empty string value

## Data Publishing Services using REST

The RCRAInfo Node has exposed Representational State Transfer (REST) publishing capabilities for its user community and the public. This capability allows retrieval of public data from the RCRAInfo system in an XML format. The XML schema is that of the RCRA dataflow on the EN.

### Access / Authorization

A username and password are not required to obtain data using the RCRAInfo REST interface as only public data is made available.

All RCRAInfo REST services have the following base URLs.

|  |  |
| --- | --- |
| **Environment** | **Base URL** |
| Pre-production | <http://rcrainfopreprod.epa.gov/webservices/rcrainfo> |
| Production | <http://rcrainfo.epa.gov/webservices/rcrainfo> |

### Services

The URL format to access any RCRAInfo REST service is:

Base\_URL/public/query/rcra/{service\_name}/{parameter\_name}/{parameter\_value}

where /{parameter\_name}/{parameter\_value} repeats for each parameter provided.

Sample URL:

https://rcrainfopreprod.epa.gov/webservices/rcrainfo/public/query/rcra/GetHDDataByHandler/handlerId/KSD007482029/changedate/2010-03-01

See Appendix B for a list of services and parameters.

### Error Handling

In case of an error, REST services will respond with an HTTP Error Code and XML containing application error code and message.

**Sample HTTP error with xml response:**

HTTP/1.1 403 Forbidden

Date: Tue, 15 Jun 2010 16:03:03 GMT

Content-Type: application/xml

Content-Length: 160

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<Error>

<Code>E\_RestServiceUnavailable</Code>

<Message>REST Service currently unavailable</Message>

</Error>

**Common Error Codes:**

|  |  |  |
| --- | --- | --- |
| **HTTP Error Code** | **Error Code** | **Message** |
| 400 | E\_RestNotSupported | REST currently not supported |
| 403 | E\_RestUnavailable | REST currently unavailable |
| 403 | E\_RestServiceUnavailable | REST Service currently unavailable |
| 403 | E\_RestModuleUnavailable | REST Module currently unavailable |
| 500 | E\_TechnicalDifficulties | we are experiencing technical difficulties |

# Data Validation

Partners are also strongly encouraged to consult with the RCRAInfo Documentation/DED (<https://rcrainfo.epa.gov/rcrainfo-help/application/ded/index.htm>) to familiarize themselves with the RCRAInfo specific data validation rules. There are numerous data quality rules that are dependent upon data in the RCRAInfo database, and therefore cannot be validated until loaded into the system. It is the responsibility of the submitting party to ensure that their data meets **all** the RCRAInfo validation rules and to confirm that the data was successfully loaded into RCRAInfo.

## Understanding the RCRAInfo Schemas

For each of the supported modules of RCRAInfo, three diagrams are presented as reference to implementers of the RCRA flow: These diagrams are the:

* RCRAInfo Entity Relationship Diagram
* XML Schema Hierarchy
* RCRAInfo Deletion Hierarchy

These diagrams are intended to help implementers understand and visualize the correlation between the RCRAInfo relational data model and the RCRA Network Exchange XML Schema. In addition, the RCRAInfo deletion hierarchy is presented to give implementers an understanding and impacts of cascade deletes and their implications for Schema implementation.

















## Schema Implementation

### Null/Optional Values

Null element references should not be included in the RCRAInfo XML submission document. It should be noted that the RCRAInfo translation routines have extensive validation checking, addressing items such as, required fields, referential integrity, and permitted values. This validation is addressed after the XML documents are loaded to RCRAInfo staging tables for RCRAInfo processing.

### Transaction Codes

As described in 2.2.1.2, valid transaction codes for the “transactional” payload operation are:

* A (Add/Update)
* D (Delete)

When using the Transactional payload operation, transaction codes must be supplied for entities that are supported by transaction codes.

Due to the hierarchical structure of the schema and the fact that system keys necessary for identifying child data (in RCRAInfo) are located in the parent node of the schema, complete parent data must be supplied along with a parent transaction code when acting on a child record. In this instance the parent record will also be transacted upon, necessitating that all relevant data be provided in the XML document.

Appendix A - RCRAInfo Flow Implementation and Testing Checklist

The following Implementation Testing and Checklist is intended to be used as a guide for State’s when setting up their RCRAInfo Network Exchange. The purpose of the pre-exchange testing process is to build confidence amongst the partners that data is being exchanged with the expected results.

The guidance may be considered for inclusion in the Trading Partner Agreements as a confirmation between partners that the necessary set-up and testing has been performed prior to implementation of the production data exchange. Since the scope of the RCRAInfo Network Exchange is different for each State, the TPA process should also be used to define the scope and detail of testing that will be required for each module.

These checklists are not intended to guide the State through the set-up and implementation of a Node. These guidelines assume that the Node has been set-up and configured prior to implementing the RCRAInfo Network Exchange (e.g., Obtaining RCRAInfo authorization). Rather these are intended to be used as a guide to implement the specifics of the RCRAInfo Exchange

**Implementation Checklist**

| **Task Description** | **Notes** | **Completed** |
| --- | --- | --- |
| Evaluate readiness and compatibility of state system data sources for exchange with RCRAInfo | Several issues outlined in the reference document *RCRAInfo Data Submission Overview* and Challenges may impact State’s priorities and approach for exchanging data with RCRAInfo. Example issues include:   * Mapping of look-up values from State to RCRAInfo * Use of sequence numbers / replication of RCRAInfo keys * Validation rules being applied to older data * Impacts of Implementer of Record issues. |  |
| Obtain or review the document online RCRAInfo Documentation/DED | This document may be obtained at http://www.exchangenetwork.net/data-exchange/rcrainfo/. It is critical that the tester have an understanding of the scope of validation rules that will be used to evaluate the submitted data. |  |
| Evaluate the need to modify State data system and download source data from RCRAInfo. | As previously mentioned and covered in depth in the reference document *RCRAInfo Data Submission Overview and Challenges,* it is necessary to replicate RCRAInfo’s key structure when submitting data. In addition, if the State is attempting to send data which is referenced by other data, the exact keys must be referenced when submitting the data.  For example: A State violation record is referenced by an EPA enforcement record. When the State begins to submit this data to RCRAInfo, the violation record’s keys, as submitted by the State, must exactly match the RCRAInfo sourced record’s keys, for the EPA owned enforcement record to continue to correctly reference the violation record.  To meet this requirement States may need to modify their system to include the additional key data as reference and perform an extract from RCRAInfo to match-up the data. |  |
| Enter Payload operation. | The payload operation for XML submission is **Transactional.** |  |
| Obtain RCRAInfo User IDs with authorization to transact within the RCRAInfo Pre-Production and Production databases. | The RCRAInfo user ID is required in the Header portion of the submission and is necessary for rights to the RCRAInfo system.    A new user may register to RCRAInfo at its first page. Once registered, a translator has to request ‘Delete’ or up permission on each module he/she wants to flow data.  Pre-Production is separated from Production. A Production id and its associated permission are not applicable to Pre-Production, and vice-versa. That is, a user has to register in both platforms as suggested.  It is advisable to also obtain a CDX User ID. This will allow testers to access the CDX Web interface to view the results of the RCRAInfo data validation process when testing. This data can also be obtained through a node GetStatus and Download call. |  |
| Contact CDX to pair the States’ NAAS account with the RCRAInfo User ID that will be provided in the header of the submission. | This step is necessary to migrate data from CDX to RCRAInfo staging. The User must have security rights to RCRAInfo before this pairing can occur. Contact RCRAInfo staff for these security rights. |  |
| Enter into agreement with EPA / Region on the approach to testing, submission and data stewardship. | The Regions express varying preferences for involvement in the submission process. Some are active in the testing and validation, while others prefer that the State take ownership of the process.  Based on the experiences of some States, this step can be a very important task. Data once housed in RCRAInfo that is now being sourced from the State’s system may have differing degrees of coverage on optional elements, and overall quality. The parties may need to come into agreement on expectations between the parties.    These issues and agreements may be addressed through a Trading Partner Agreement |  |
| Notify EPA RCRAInfo system support of intention to submit data for testing | A relationship with the RCRAInfo support team is beneficial in testing and troubleshooting of submissions. |  |
| Perform testing of selected processing method and XML document creation. | See submission test cases.  To be successful and have the data move form staging to production all errors in the RCRAInfo validation must be resolved. |  |
| Notify EPA RCRAInfo system support of intention to submit data to production system.  Notify CDX of intent to submit data to RCRAInfo Production to authorize NAAS account for production CDX node. | Authorization and verification from RCRAInfo support is necessary to send data to the production system. |  |

Submission Test Cases

The breadth of testing should take place only to the extent that the State will use the schema and the associated payload operations.

Upon successful completion of testing on the pre-production database, the State should notify the designated RCRAInfo staff for validation. After sign-off from RCRAInfo staff, the State should complete some basic tests in the Production database. A full test suite may not be necessary against the Production RCRAInfo database, as the Pre-production database mirrors the Production in both content and form.

General Suggestions

The following testing checklists are organized by payload operation. These suggested confirmations are applicable for all modules being exchanged through the payload operation.

Transactional Payload Operation Testing Checklist

| **Test Description** | **Expected Results** | **Notes** | **Completed** |
| --- | --- | --- | --- |
| Perform an Add/Update transaction for each entity (e.g., Enforcement Actions, Penalties) | The record should be added or updated to the database. |  |  |
| Make modifications to the data in the source system. Regenerate the record(s) and re-submit the records | The selected record should be updated with the data modifications. | This test is necessary to confirm that the State system is correctly generating/referencing RCRAInfo keys. If this is not being done correctly, the record will be identified as a new record and added to RCRAInfo, rather than updating the existing record |  |
| Perform a deletion of a parent record in the State system | The deleted record should be deleted in RCRAInfo. | This test is necessary to confirm that the State source system is not sending delete transactions for cascade deleted children records. |  |
| Modify data element(s) which RCRAInfo considers to be a primary key (e.g., Evaluation Date). | The source State system should identify this as a delete transaction for RCRAInfo, as well as an add transaction for the “new” record. | This test case is only necessary if the State source system allows editing of data elements which RCRAInfo considers to be primary keys. |  |
| Perform an add/update transaction on State data that is associated to EPA owned data | The State owned data should be added or updated in RCRAInfo. Errors should not be raised by RCRAInfo referencing IOR issues | This test may be necessary only for those States that model cross ownership of data in their source system. |  |
| Delete a join record in a many to many relationship (e.g. Compliance Schedule) | The association between items in the many to many relationship, should be removed but the individual entities should remain in RCRAInfo (e.g., Violations and Enforcement Action) | This test is necessary to confirm that the XML documents are being structured properly and placing the delete transaction in the correct level of the data hierarchy in the schema |  |
| Delete a primary element involved in a many to many relationship (e.g., Enforcement) | The primary element should be deleted with the associated join record being cascade deleted by RCRAInfo | This test is necessary to confirm that the XML documents are being structured properly and placing the delete transaction in the correct level of the data hierarchy in the schema |  |
| Perform a full load of data | Verify key data points are correctly being loaded into the system | It is recommended that prior to initiating a full load of data a snapshot of data is obtained from RCRAInfo, to facilitate data verification. Whether the snapshot is a download of data from RCRAInfo, or screen shots of select sites’ data. |  |
| Perform an overall record count assessment | The correct number of records should be expected in each table in RCRAInfo |  |  |

# Appendix B – RCRAInfo Data Access Services

Below is the list of RCRAInfo data access service requests that are available over the Exchange Network and/or REST.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Service** | **Parameters** | **EN Query** | **EN Solicit** | **REST** |
| GetCADataByHandler | * handlerId * changeDate | Yes | No | Yes |
| GetCADataByState | * state * changeDate | No | Yes | No |
| GetCEDataByHandler | * handlerId * state * agency * changeDate | Yes | No | Yes |
| GetCEDataByState | * state * changeDate | No | Yes | No |
| GetCEDeletedDataByState | * state * changeDate | No | Yes | No |
| GetFADataByHandler | * handlerId * changeDate | Yes | No | Yes |
| GetFADataByState | * state * changeDate | No | Yes | No |
| GetGSDataByHandler | * handlerId * changeDate * owner * sequenceNumber | Yes | No | Yes |
| GetGSDataByState | * state * changeDate | No | Yes | No |
| GetHDDataByHandler | * handlerId * changeDate * state * sourceType * sequenceNumber | Yes | No | Yes |
| GetHDDataByState | * state * changeDate * endDate * sourceType * reportCycle | No | Yes | No |
| GetHDMaxSequence | * handlerId * sourceType * stateId | Yes | No | Yes |
| GetPMDataByHandler | * handlerId * changeDate | Yes | No | Yes |
| GetPMDataByState | * state * changeDate | No | Yes | No |
| GetCurrentHandlerByID | * handlerId * changeDate | Yes | No | Yes |
| GetCurrentHandlerByState | * state * changeDate | No | Yes | No |
| GetEMDataByState | * state * changeDate * endDate | No | Yes | No |

**GetCADataByHandler:**

This service will retrieve corrective action data for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| changeDate | No |

**GetCADataByState:**

This service will retrieve corrective action data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetCEDataByHandler:**

This service will retrieve compliance monitoring and enforcement data for the specified handler id. Other parameters can be used to filter the amount of data returned. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| state | No |
| agency | No |
| changeDate | No |

**GetCEDataByState:**

This service will retrieve compliance monitoring and enforcement data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this state will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetCEDeletedDataByState:**

This service will provide CM&E deletion audit information. It will retrieve compliance monitoring and enforcement deleted records key information for the specified state. If a change date is provided, the result will only contain information that has been modified/updated since the change date. If a change date is not provided all the information for this state will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetFADataByHandler:**

This service will retrieve financial assurance data for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| changeDate | No |

**GetFADataByState:**

This service will retrieve financial assurance data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetGSDataByHandler**

This service will retrieve geographic information system data for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| changeDate | No |
| owner | No |
| sequenceNumber | No |

**GetGSDataByState:**

This service will retrieve geographic information system data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetHDDataByHandler:**

This service will retrieve handler data for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| changeDate | No |
| sourceType | No |
| sequenceNumber | No |
| state | No |

**GetHDDataByState**

This service will retrieve handler data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned. If an end date is provided, the result will only contain data that has been modified/updated before the end date.

When sourceType = ‘B’, parameter reportCycle will apply, and parameters, changeDate & endDate, will be ignored at the same time. Otherwise reportCycle will be ignored.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |
| endDate | No |
| sourceType | No |
| reportCycle | No |

**GetPMDataByHandler:**

This service will retrieve permitting data for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| changeDate | No |

**GetPMDataByState:**

This service will retrieve permitting data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetHDMaxSequence**

This service will retrieve the maximum handler source record sequence number used for the specified handler id. The source type and state are optional parameters. If the source type is provided, only the maximum sequence number for the specified source type will be provided. The state id is used for the rare occurrence where a handler is associated to more than one activity location.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| handlerId | Yes |
| sourceType | No |
| stateId | No |

**GetCurrentHandlerById:**

This service will retrieve current handler data from Report\_Univ for the specified handler id. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| Id | Yes |
| changeDate | No |

**GetCurrentHandlerByState**

This service will retrieve handler data from Report\_Univ for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data for this handler will be returned.

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No |

**GetEMDataByState**

This service will retrieve eManifest data for the specified state. If a change date is provided, the result will only contain data that has been modified/updated since the change date. If a change date is not provided all data from very beginning for this ‘state’ will be returned. If an end date is provided, the result will only contain data that has been modified/updated before the end date. (The change date is an inclusive from date while the exclusive end date is a to date. Since emanifest date columns have time part, changeDate and endDate will have a static 00:00:000 time part.)

|  |  |
| --- | --- |
| **Parameter Name** | **Required** |
| state | Yes |
| changeDate | No (may change to Yes if any performance issue pops up) |
| endDate | No (may change to Yes if any performance issue pops up) |

Compared with other modules, Emanifest has its own structures. The ‘state’ parameter will apply to Generator / tsdf / Transporters (to check if a generator is in the input ‘state’, for example).

Similar to Emanifest API, following optional parameters may apply (in second phase):

|  |  |
| --- | --- |
| **Parameter Name** | **Notes** |
| dateType | CertifiedDate  ReceivedDate  ShippedDate |
| siteType | Generator  TSDF |
| status | InTransit  readyForSignature  Received  Scheduled  Signed |

If ‘dateType’ parameter is not specified, the ‘changeDate’ (and ‘endDate’) parameter will apply to UPDATED\_DATE column. Otherwise, the date range parameter will apply to the specified date column.

If ‘siteType’ parameter is not specified, result data includes all site type manifests. Otherwise, result data includes only the specified site type manifests.

If ‘status’ parameter is not specified, result data includes all data except pending manifest. Otherwise, result data includes only specified status manifests.