

[illegible]

```

tion>Schema Name: NEI...der
documentation>
tion>Current Version
p://www.epa.gov/exchange...tion>Sch
cumenta
tion>Description: The NEI...tion>Cur
</xsd:documentation> //ww
tion>Application: Varies by
cumentation> >Des
tion>Developed By: Environme...tion>do
"UTF-8" ?> App
tat
: //www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unqu
ation="EN_NEI_Common_v3_0.xsd

```

```

tion>Current Version
p://www.epa.gov/exchange>Sch
cumenta
tion>Description: The NEI >Cur
</xsd:documentation> /ww
tion>Application: Varies by
cumentation> >Des
tion>Developed By: Environme
"UTF-8" ?> App
tat
3v
: //www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unqu
ation="EN_NEI_Common_v3_0.xsd

```

```

p://www.epa.gov/exchange/...>Sch
cumenta
tion>Description: The NEI ...>Cur
</xsd:documentation>
tion>Application: Varies by
cumentation>
tion>Developed By: Environme
"UTF-8" ?>
p://www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unqu
ation="EN_NEI_Common_v3_0.xsd

```

```
</xsd:documentation>
tion>Application: Varies by
documentation>
tion>Developed By: Environment: do
"UTF-8" ?>
App
tat
?w
: //www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unqu
ation="EN_NEI_Common_v3_0.xsd
```

```

tion>Application: Varies by
documentation>
tion>Developed By: Environment
"UTF-8" ?>
: //www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unq
ation="EN_NEI_Common_v3_0.xsd

```

```

on>Developed By: Environme1:do
"UTF-8" ?>
App
tat
2V
:/www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
ualified" attributeFormDefault="unq
ation="EN_NEI_Common_v3_0.xsd

```

```

"UTF-8" ?>
: //www.epa.gov/exchangenetw
w.w3.org/2001/XMLSchema"
w.epa.gov/exchangenetwork"
qualified" attributeFormDefault="unq
ation="EN_NEI_Common_v3_0.xsd

```

```

entation>
Current Version
www.epa.gov/exchangenetwork<
Description: The NEI XML 3.0 Point
documentation>
Application: Varies by
tation>

```

**Description:** The NEI XML 3.0 Point  
documentation>  
**Application:** Varies by  
tation>

documentation>  
Application: Varies by  
tation>





## DOCUMENT CHANGE RECORD

| Version Number | Date          | Description  |
|----------------|---------------|--|
| 0.1            | May, 01, 2007 | Draft  |
| 0.2            | May, 22, 2007 | Updated the document to include details of the processing.   |
| 0.9            | Jun 22, 2007  | Updated the document based on the working sessions with CDX. |
| 0.91           | Jul 20, 2007  | Incorporated EPA comments.                                   |
| 0.92           | Aug 20, 2007  | Incorporated comments from state users                       |
| 1.0            | Aug 29, 2007  | Final Delivery   |
| 2.0            | Sep 28, 2010  | Revised for Full Batch Phase 1                               |
| 2.1            | Oct 21, 2010  | Incorporated EPA Comments                                    |
| 3.0            | Sept 8, 2011  | Revised for Full Batch Phase 2                               |

## TABLE OF CONTENTS

|   |           |
|---|-----------|
| <b>1. INTRODUCTION.....</b>                               | <b>13</b> |
| 1.1 PURPOSE .....   | 13        |
| 1.2 ASSUMPTIONS AND CONSTRAINTS .....                     | 13        |
| 1.3 AUDIENCE.....   | 14        |
| 1.4 DOCUMENT OVERVIEW .....                               | 14        |
| <b>2. XML SUBMISSION DOCUMENT .....</b>                   | <b>15</b> |
| 2.1 OVERVIEW .....  | 15        |
| 2.2 XML SUBMISSION DOCUMENT STRUCTURE .....               | 15        |
| 2.2.1 Header .....  | 15        |
| 2.2.2 Payload.....  | 17        |
| <b>3. RESPONSE DOCUMENTS .....</b>                        | <b>19</b> |
| 3.1 FULL BATCH RESPONSE.....                              | 20        |
| 3.1.1 Batch Audit Report .....                            | 20        |
| 3.1.2 Batch XML Response Files .....                      | 24        |
| 3.1.3 Batch File-Level Error Report .....                 | 29        |
| 3.1.4 Batch File-Level Error XML Report .....             | 29        |
| 3.2 NETDMR RESPONSE .....                                 | 30        |
| 3.3 ICIS BATCH DMR HYBRID RESPONSE .....                  | 31        |
| <b>4. ICIS–NPDES BATCH FLOW DESCRIPTION.....</b>          | <b>33</b> |
| 4.1 FILE EXCHANGE IN NODE TO NODE CONFIGURATION.....      | 33        |
| 4.2 FILE EXCHANGE IN WEB FORM TO NODE CONFIGURATION ..... | 35        |
| <b>5. ICIS–NPDES BATCH NODE WEB METHODS.....</b>          | <b>39</b> |
| 5.1 SUBMIT WEB METHOD .....                               | 39        |
| <b>6. CDX NODE WEB METHODS.....</b>                       | <b>40</b> |
| 6.1 AUTHENTICATE WEB METHOD .....                         | 40        |
| 6.2 SUBMIT WEB METHOD .....                               | 40        |
| 6.3 NOTIFY WEB METHOD.....                                | 41        |
| <b>7. WEB METHODS USED BY STATE NODE.....</b>             | <b>42</b> |
| 7.1 AUTHENTICATE WEB METHOD .....                         | 42        |
| 7.2 SUBMIT WEB METHOD .....                               | 42        |
| 7.3 GET STATUS WEB METHOD .....                           | 43        |
| 7.4 DOWNLOAD WEB METHOD.....                              | 43        |
| <b>APPENDIX A: ICIS–NPDES BATCH SUBMISSION TYPES.....</b> | <b>45</b> |

## LIST OF FIGURES

|  |    |
|--|----|
| Figure 2-1 - XML Submission Document Sample .....                    | 15 |
| Figure 3-1 - Accepted Transactions PDF Report Screen Shot .....      | 21 |
| Figure 3-2 - Rejected Transactions PDF Report Screen Shot.....       | 22 |
| Figure 3-3 - Batch Transactions Summary PDF Report Screen Shot.....  | 22 |
| Figure 3-4 - Accepted Transactions XML Report Screen Shot.....       | 24 |
| Figure 3-5 - Rejected Transactions XML Report Screen Shot .....      | 25 |
| Figure 3-6 - Batch Transactions Summary XML Report Screen Shot ..... | 26 |
| Figure 3-7 - Batch File-Level Error Report Screen Shot .....         | 29 |
| Figure 3-8 - Batch File-Level Error XML File Screen Shot.....        | 29 |
| Figure 3-9 - NetDMR Report Screen Shot .....                         | 30 |
| Figure 3-10 - ICIS Batch DMR Hybrid Report Screen Shot .....         | 31 |
| Figure 4-1 - File Exchange in Node to Node Configuration .....       | 33 |
| Figure 4-2 - File Exchange in Client to Node Configuration .....     | 36 |

## LIST OF TABLES

|   |    |
|---|----|
| Table 2-1 - XML Submission Document Header Details.....   | 16 |
| Table 2-2 - XML Submission Document Payload Details ..... | 17 |
| Table 3-1 - Batch Response Files Details .....            | 20 |
| Table 3-2- Batch Audit Report Details .....               | 23 |
| Table 3-3 - Batch XML Response Files Details .....        | 27 |
| Table 3-4 - Batch File Level Error XML File Details ..... | 29 |
| Table 3-5 - NetDMR Response File Details .....            | 30 |



## 1. INTRODUCTION

---

The Integrated Compliance Information System – National Pollutant Discharge Elimination System (ICIS–NPDES) is being extended to support an integrated data flow to process electronic batch submissions using eXtensible Markup Language (XML) technology. Authorized states, tribes, and regional users submit National Pollutant Discharge Elimination System (NPDES) data via batch to ICIS–NPDES. They compose their transactions into predefined XML formatted files contained in a zip archive file to submit to ICIS–NPDES through the Environmental Protection Agency’s (EPA) Central Data Exchange (CDX).

Document submissions to CDX must be preceded by NAAS authentication. Upon successful authentication, the client node receives a security token that must be used in all subsequent service requests with CDX. Afterwards, the batch is submitted to CDX, passing in the security token. If the security token is valid, CDX generates a transaction ID for the request and returns it to the client node.

Upon receipt of the batch, CDX performs important functions on batch files such as validating the submitted data against approved XML schemas, scanning for viruses, archiving all XML files, and authenticating the submitters prior to making the XML files available for processing. CDX provides a Web services interface for distributing files to ICIS–NPDES for processing.

A Web service is deployed in the ICIS–NPDES environment to receive XML files from CDX. ICIS–NPDES is responsible for receiving and parsing the XML files, sequencing the transactions, and processing the transactions against the ICIS–NPDES service tier. For any transactions that are not successfully processed, detailed business-rule based errors will be generated. Finally, ICIS–NPDES records the batch processing results and sends the processing status back to CDX.

### 1.1 PURPOSE

The Flow Configuration Document (FCD) defines the required details to implement and configure a flow across CDX and ICIS–NPDES nodes to exchange XML data. These nodes will implement a subset of the web methods from the Network Node Functional Specification to facilitate the exchange of data between the state users and the ICIS–NPDES Batch system. This document should be referenced in the context of submitting ICIS–NPDES data to ICIS Batch. The NetDMR Flow Configuration Document addresses data exchanges between ICIS–NPDES and NetDMR.

### 1.2 ASSUMPTIONS AND CONSTRAINTS

The following assumptions apply to the ICIS–NPDES Batch flow configuration:

- All batch files will be zipped by the submitting party prior to submittal to CDX.
- Upon receiving files from states, CDX validates the received files against the target XML schema. Only valid batches are submitted to ICIS–NPDES. A batch is valid if at least one

of the files in that batch is a valid XML file. A file is valid if it is well formed and complies with the ICIS NPDES schema.

- CDX scans the received files for any viruses. Files with viruses will be rejected.
- CDX archives the received files from state users.
- CDX supports both manual and automatic batch submissions. For manual submissions, users can log in to the ICIS–NPDES CDX Web Form to submit batches. For automatic submissions, users will choose to configure a node in their environment to submit (“push”) the files to CDX.

### 1.3 AUDIENCE

The primary audience for this document is developers, project managers and architects throughout the ICIS–NPDES Batch implementation stakeholder organizations.

### 1.4 DOCUMENT OVERVIEW

The following sections comprise the rest of the document:

- **Section 2: XML Document Structure** – This section describes the overall structure of the ICIS–NPDES Batch XML Schemas.
- **Section 3: ICIS–NPDES Batch Flow Description** – This section describes the overall flow details for exchanging data between states, CDX and ICIS–NPDES.
- **Section 4: ICIS–NPDES Batch Node Web Methods** – This section describes the Web methods implemented by the ICIS–NPDES Batch node.
- **Section 5: CDX Node Web Methods** – This section describes the Web methods implemented by CDX.
- **Section 6: Web Methods Used by State Node** – This section describes the Web methods a state node can use to retrieve the required information from CDX.



## 2. XML SUBMISSION DOCUMENT

### 2.1 OVERVIEW

The ICIS–NPDES Batch Submission Document is a XML file that contains metadata to facilitate the data processing and sandwiching of ICIS–NPDES data for a specific submission type. Authorized states, tribes, or regions submitting Batch data to ICIS–NPDES must comply with the ICIS–NPDES Batch Document schema. Appendix A – ICIS–NPDES Batch Submission Type lists all the submission types supported by ICIS–NPDES Batch. Instructions to create XML instance submission documents for different submission types are contained in the ICIS–NPDES Batch User Guide.

A batch submission may consist of one or more submission documents. Batches must be submitted in compressed format (zip file).

### 2.2 XML SUBMISSION DOCUMENT STRUCTURE

The XML Submission Document is divided into two main sections, namely the header section and the payload section. The following sub-sections describe their structure, providing a brief description and example values for their major elements and attributes.

Figure 2-1 – XML Submission Document Sample, represents an example of instance document for the Permit Tracking Event submission type. In the picture, the header portion of the document is highlighted in yellow, while the payload portion is highlighted in green.

**Figure 2-1 - XML Submission Document Sample**

```
<?xml version="1.0" encoding="UTF-8"?>
<Document xmlns="http://www.exchangenetwork.net/schema/icis/3" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Header>
    <Id>UUStaffer1</Id>
    <Author>Jane Doe</Author>
    <Organization>UU Department of Environmental Protection</Organization>
    <Title>Permit Tracking Event Submission</Title>
    <CreationTime>2001-12-17T09:30:47.0Z</CreationTime>
    <Comment>2nd resubmittal of rejected transactions from May submission</Comment>
    <DataService>ICIS-NPDES</DataService>
    <ContactInfo>123 Main Street, Anytown UU, 00555, (888) 555-1212, jane.doe@uudeq.state.us</ContactInfo>
    <Property>
      <name>e-mail</name>
      <value>doe.john@state.us</value>
    </Property>
  </Header>
  <Payload Operation="PermitTrackingEventSubmission">
    <PermitTrackingEventData>
      <TransactionHeader>
        <TransactionType>N</TransactionType>
        <TransactionTimestamp>2001-12-17T09:30:47.0Z</TransactionTimestamp>
      </TransactionHeader>
      <PermitTrackingEvent>
        <PermitIdentifier>AL1005932</PermitIdentifier>
        <PermitTrackingEventCode>ANC</PermitTrackingEventCode>
        <PermitTrackingEventDate>2005-12-31</PermitTrackingEventDate>
        <PermitTrackingCommentsText>Application returned to applicant 3
times</PermitTrackingCommentsText>
      </PermitTrackingEvent>
    </PermitTrackingEventData>
  </Payload>
</Document>
```

```

    </PermitTrackingEvent>
  </PermitTrackingEventData>
</Payload>
</Document>

```

### 2.2.1 Header

The header section contains information about the document, such as the author, his/her employer and creation time. Table 2-1- XML Submission Document Header Details, describes the elements and attributes of the header and briefly explains how they are utilized during the processing of submitted data.

**Table 2-1 - XML Submission Document Header Details**

| Element      | Description  | Example Value   | Required | Notes   |
|--------------|--|---|----------|---|
| Id           | The ICIS ID of the user submitting the XML document  | Jdoe  | Yes      | The username is used to authenticate and determine if the user's ICIS ID has the rights to add, change or delete a record for the data family being submitted to ICIS |
| Author       | The first and last name of the person generating the XML document  | John Doe  | No       | Reference   |
| Organization | The name of the company generating the XML document  | State X Department of Environmental Quality                   | No       | Reference   |
| Title        | The type of submission   | Permit Tracking Event Submission                              | No       | Reference   |
| CreationTime | The date and time the document was created   | 2007-06-15T01:30:00.0Z  | No       | Reference   |
| Comment      | Free text information of the message contents  | 2nd resubmission of rejected transactions from May submission | No       | Reference   |
| DataService  | Name of backend application  | ICIS-NPDES  | No       | Always ICIS-NPDES   |
| ContactInfo  | Name, mailing address, city, state, zip, telephone number, and email address of the person who may be contacted with questions concerning the submission | John Doe<br>100 Somewhere St<br>McLean, VA 22102              | No       | Reference   |

| Element  | Description   | Example Value        | Required   | Notes  |
|----------|---|----------------------|--|--|
| Property | Name-value pairs used to provide additional information about the submitter | N/A (parent element) | No   | When the property name is "Source", its value is used to identify the source of the batch submission and therefore the appropriate response. Refer to Section 3 for details. |
| Name     | Name of the property  | e-mail               | Yes, if e-mail notification is desired, otherwise No | Used to identify the type of property. These are custom properties to facilitate the efficient communication of status.  |
| Value    | Value of the property   | jdoe@epa.gov         | Yes, if name element is present                      | Used to communicate the processing status to the submitter.  |

## 2.2.2 Payload

The payload section contains ICIS–NPDES data to submit to the ICIS–NPDES system. The data include the payload submission type, transaction time, transaction type, and NPDES data corresponding to the selected submission type. Table 2-2- XML Submission Document Payload Details, describes required elements and attributes of the payload section. Other elements and attributes must be present, depending on the specific submission type. An XML Submission Document may contain multiple payloads.

**Table 2-2 - XML Submission Document Payload Details**

| Element               | Description   | Example Value  | Required | Notes  |
|-----------------------|---|--|----------|--|
| Operation (attribute) | This attribute provides the payload submission type                     | Discharge Monitoring Report Submission,<br>BasicPermitSubmission,<br>ParameterLimitsSubmission | Yes      | Refer to the ICIS–NPDES Batch User Guide for a complete list of the possible submission types.         |
| Transaction Type      | The type of transaction that should be performed on the submitted data. | C, D, N, R, X  | Yes      | Refer to the ICIS–NPDES Batch User Guide for the transaction types available for each submission types |

|                          |  |                        |    |  |
|--------------------------|--|------------------------|----|--|
| Transaction<br>Timestamp | The date and time the<br>data was extracted. | 2001-12-17T09:30:47.0Z | No |  |
|--------------------------|--|------------------------|----|--|

### 3. RESPONSE DOCUMENTS

---

After batch data is submitted and processed, ICIS returns response files with information regarding the processing outcome of the submission to the user.

Response files are different for ICIS Full Batch Users, NetDMR Users, and ICIS Batch DMR Hybrid Users. ICIS Full Batch Users received a PDF with three tabs listing accepted transactions, rejected transactions and a summary of submission counts along with an XML file to match each tab. NetDMR Users only receive an XML file of rejected transactions. DMR Hybrid Users do not receive transaction result listings or counts at all but view their results within the ICIS application using the Business Objects reporting tool.

ICIS distinguishes between the three sources for batch submissions based on the information included in the header of the XML submission files. Specifically, the Name and Value tags under the Property tag are used to identify the source of the batch submission and therefore the appropriate response.

For ICIS Full Batch Users, the Property tag contains the following data:

```
<Property>
  <name>Source</name>
  <value>FullBatch</value>
</Property>
```

For NetDMR Users, the Property contains the following data:

```
<Property>
  <name>Source</name>
  <value>NetDMR</value>
</Property>
```

DMR Hybrid Users do not submit a Property tag with a name tag containing "Source". Note that only one Property tag can exist within an XML file with a name tag containing "Source".

The Property tag can also be used to instruct CDX to send an email notification to one or more people when a submission arrives at CDX and after it has been processed by CDX. This is done by having the name tag contain "e-mail" and the value tag containing an email address. These tags can be before or after a Property tag with a name tag containing "Source." The following is an example of Property tags for an ICIS Full Batch User wanting CDX submission email notifications to be sent to three people:

```
<Property>
  <name>Source</name>
  <value>FullBatch</value>
</Property>
```

```

<Property>
  <name>e-mail</name>
  <value>deer.john@state.gov</value>
</Property>
<Property>
  <name>e-mail</name>
  <value>doe.jane@state.gov</value>
</Property>
<Property>
  <name>e-mail</name>
  <value>buck.jack@state.gov</value>
</Property>

```

Table 3-1 – Batch Response Files Details, lists the response reports for different batch origins. The following sub-sections describe each report.

**Table 3-1 - Batch Response Files Details**

| Origin     | Archive File Name  | Report Name                           | Report File Name   |
|------------|--|---------------------------------------|--|
| FullBatch  | <Transaction ID>_<Submitting Party>_<TimeStamp>_Response.zip | Batch Audit Report                    | <Transaction ID>_<Submitting Party>_<TimeStamp>_ Response.pdf        |
|            |  | Accepted Transactions XML Report      | <Transaction_ID>_<SubmittingParty>_<TimeStamp>_Accepted_Response.xml |
|            |  | RejectedTransactions XML Report       | <Transaction_ID>_<SubmittingParty>_<TimeStamp>_Rejected_Response.xml |
|            |  | Batch Transactions Summary XML Report | <Transaction_ID>_<SubmittingParty>_<TimeStamp>_Summary_Response.xml  |
|            |  | Batch File-Level Error Report         | <Transaction_ID>_<SubmissionDate>_File_Error_Response.pdf            |
|            |  | Batch File-Level Error XML Report     | <Transaction_ID>_<SubmissionDate>_File_Error_Response.xml            |
| NetDMR     | <Transaction ID>_Response.zip                                | NetDMR Report                         | <Transaction ID>_Response.xml  |
| DMR Hybrid | N/A  | ICIS Batch DMR Hybrid Report          | <Transaction ID>_Response.pdf  |

### 3.1 FULL BATCH RESPONSE

Full Batch Users receive a Batch Audit Report in PDF format, with details about accepted transactions, rejected transactions, and totals for the accepted and rejected transactions on separate tabs. Users also receive report files in XML format, with data similar to the one in the PDF file.

#### 3.1.1 Batch Audit Report

The Batch Audit Report includes:

- the Accepted Transactions report,
- the Rejected Transactions report
- the Batch Transactions Summary report.

The Accepted Transactions Report displays information about transactions that were processed in ICIS without errors. Warning messages may also be displayed on the Report, if they were generated during transaction processing. Figure 3-1 – Accepted Transactions PDF Report Screen Shot, displays a sample Accepted Transactions report.

**Figure 3-1 - Accepted Transactions PDF Report Screen Shot**

| Environmental Protection Agency<br>Integrated Compliance Information System<br>Accepted Transactions Report |            |              |   | Created Date: 05/17/2007<br>Refresh Date: 06/24/2010<br>Created By: HQ |
|---|------------|--------------|---|--|
| Batch Submission Date From: * To: *   |            |              |   |  |
| Batch ID(s): f802a1c8-38f4-4fa5-95a9-ae0a168e8dee   |            |              |   |  |
| Submitting Party: HQ  |            |              |   |  |
| User ID(s): *   |            |              |   |  |
| Submitting Party: Region HQ   |            |              |   |  |
| Batch ID  | User ID    | Run Date     | Submission Date   |  |
| f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  | UNGERA     | 06/24/2010   | 06/24/2010  |  |
| Submission Type   | Trans Type | Key Value    | Error / Warning Message(s)  |  |
| Basic Permit  | New        | WYREP0001    | Warning: Telephone Extension Number is not a valid tag for Contact Telephone Number Type Code PGE. The Telephone Extension Number 423 was not saved.; |  |
| Basic Permit  | Change     | WYREP0002    |   |  |
| Basic Permit  | Replace    | WYREP0003    |   |  |
| Permitted Feature   | New        | WYREP0002002 |   |  |
| Batch ID  | User ID    | Run Date     | Submission Date   |  |
| f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  | WESTB      | 06/24/2010   | 06/24/2010  |  |
| Submission Type   | Trans Type | Key Value    | Error / Warning Message(s)  |  |
| Basic Permit  | New        | WYREP0004    | Warning: Telephone Extension Number is not a valid tag for Contact Telephone Number Type Code PGE. The Telephone Extension Number 423 was not saved.; |  |
| Basic Permit  | Change     | WYREP0005    |   |  |
| Basic Permit  | Replace    | WYREP0006    |   |  |
| Permitted Feature   | New        | WYREP0005002 |   |  |

The Rejected Transactions Report displays information about transactions that did not complete processing in ICIS because of errors caused by business rule violations and/or transaction processing errors. Figure 3-2 – Rejected Transactions PDF Report Screen Shot, displays a sample Rejected Transactions report.

**Figure 3-2 - Rejected Transactions PDF Report Screen Shot**

| Environmental Protection Agency<br>Integrated Compliance Information System<br>Rejected Transactions Report |            |              |   | Created Date: 05/17/2007<br>Refresh Date: 06/24/2010<br>Created By: HQ |
|---|------------|--------------|---|--|
| Batch Submission Date From: * To: *   |            |              |   |  |
| Batch ID(s): f802a1c8-38f4-4fa5-95a9-ae0a168e8dee   |            |              |   |  |
| Submitting Party: HQ  |            |              |   |  |
| User ID(s): *   |            |              |   |  |
| Submitting Party: Region HQ   |            |              |   |  |
| Batch ID  | User ID    | Run Date     | Submission Date   |  |
| f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  | UNGERA     | 06/24/2010   | 06/24/2010  |  |
| Submission Type   | Trans Type | Key Value    | Error / Warning Message(s)  |  |
| Basic Permit  | New        | WYREP0007    | The Permit Type Code NPJ is invalid for the Basic Permit Data Payload;  |  |
| Basic Permit  | Change     | WYREP0008    | A permit does not exist for the key data entered;   |  |
| Basic Permit  | Replace    | WYREP0009    | At least one Permittee Address must be entered for the permit.; Affiliation Type Text ZZB does not have an Affiliation_Category of Non-Government, or does not exist or is inactive in the ICIS reference table.;             |  |
| Permitted Feature   | New        | WYREP0010001 | A Permitted Feature cannot be added because the Permit Identifier does not exist in ICIS.;  |  |
| Batch ID  | User ID    | Run Date     | Submission Date   |  |
| f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  | WESTB      | 06/24/2010   | 06/24/2010  |  |
| Submission Type   | Trans Type | Key Value    | Error / Warning Message(s)  |  |
| Basic Permit  | New        | WYREP0011    | The Permit Type Code NPJ is invalid for the Basic Permit Data Payload.; Warning: Telephone Extension Number is not a valid tag for Contact Telephone Number Type Code PGE. The Telephone Extension Number 423 was not saved.; |  |
| Basic Permit  | Change     | WYREP0012    | A permit does not exist for the key data entered.;  |  |
| Basic Permit  | Replace    | WYREP0013    | At least one Permittee Address must be entered for the permit.; Affiliation Type Text ZZB does not have an Affiliation_Category of Non-Government, or does not exist or is inactive in the ICIS reference table.;             |  |
| Permitted Feature   | New        | WYREP0014001 | A Permitted Feature cannot be added because the Permit Identifier does not exist in ICIS.;  |  |

The Transactions Summary PDF Report displays the total number of transactions accepted, total number of transactions rejected, and the percentage of transactions accepted, at both the submission type level and at the batch level. Figure 3-3 –Batch Transactions Summary PDF Report Screen Shot, displays a sample Transactions Summary report.

**Figure 3-3 - Batch Transactions Summary PDF Report Screen Shot**



Environmental Protection Agency  
Integrated Compliance Information System  
Batch Transactions Summary Report

Created Date: 05/17/2007  
Refresh Date: 06/24/2010  
Created By: HQ

Submission Date From: \* To: \*  
Batch ID(s): f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  
Submitting Party: HQ  
User ID(s): \*

Submitting Party: Region HQ  
Batch ID: f802a1c8-38f4-4fa5-95a9-ae0a168e8dee  
Submission Date: 06/24/2010

| Transaction Type  | N        | N        | R        | R        | C        | C        | X        | X        | D        | D        |                    |                         |                   |
|-------------------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|--------------------|-------------------------|-------------------|
| Submission Type   | Accepted | Rejected | Accepted | Rejected | Accepted | Rejected | Accepted | Rejected | Accepted | Rejected | Total Transactions | % Transactions Accepted | Total Submissions |
| Basic Permit      | 2        | 2        | 2        | 2        | 2        | 2        | 0        | 0        | 0        | 0        | 12                 | 50.00 %                 | 12                |
| Permitted Feature | 2        | 2        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 0        | 4                  | 50.00 %                 | 4                 |
| Total             | 4        | 4        | 2        | 2        | 2        | 2        | 0        | 0        | 0        | 0        | 16                 | 50.00 %                 | 16                |

The following table, Table 3-2 - Batch Audit Report Details, describes the fields included in the Batch Audit Report.

**Table 3-2- Batch Audit Report Details**

| Element                         | Information   | Example   |
|---------------------------------|---|---|
| <b>All Reports</b>              |   |   |
| Submitting Party                | Two character postal code for a State or two digit number for a Region of the user that submitted the XML. EPA Headquarters is also able to submit XML batches to ICIS. | HQ, AL, or 01   |
| Batch ID                        | Identification string provided to the user by CDX after a batch is submitted to ICIS.   | f802a1c8-38f4-4fa5-95a9-ae0a168e8dee                                  |
| Submission Date                 | Date the batch was submitted by the user.   | 06/24/2010  |
| Submission Type                 | Name of the data family being updated.  | Basic Permit  |
| Transaction Type                | The type of the transaction.  | New   |
| <b>Accepted/Rejected Report</b> |   |   |
| User ID                         | ICIS ID of the person submitting the XML  | UNGERA  |
| Run Date                        | Date the batch was processed by ICIS.   | 06/24/2010  |
| Key Value                       | Key tags submitted for the XML transaction.   | WYREP0007   |
| Error/Warning Message           | Error/Warning Message generated when a business rule is violated during processing.   | The Permit Type Code NPJ is invalid for the Basic Permit Data Payload |
| <b>Summary Report</b>           |   |   |
| Accepted                        | Total number of accepted transactions per transaction type and submission type.   | 2   |
| Rejected                        | Total number of rejected transactions per transaction type and submission type.   | 2   |

| Element                 | Information  | Example |
|-------------------------|--|---------|
| <b>All Reports</b>      |  |         |
| Total Transactions      | Total number of transactions per submission type.      | 12      |
| % Transactions Accepted | Percentage of transactions that were accepted by ICIS. | 50.00%  |
| Total Submissions       | Total number of submissions for a submission type.     | 12      |
| Total                   | Total number of submissions for the entire batch.      | 16      |

### 3.1.2 Batch XML Response Files

In addition to the Batch Audit Report, ICIS returns three separate response files in XML format:

- the Accepted Transactions Report
- the Rejected Transactions Report
- the Batch Transactions Summary Report.

The Accepted Transactions Report provides the same information included in the corresponding PDF report. Figure 3-4 – Accepted Transactions XML Report Screen Shot, displays a sample Accepted Transactions report.

**Figure 3-4 - Accepted Transactions XML Report Screen Shot**

```

<?xml version="1.0"?>
<SubmissionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.exchangenetwork.net/schema/icis/1">
  <TransactionIdentifier>_457b7249-cb09-4764-98ac-ec502443c52a</TransactionIdentifier>
  <SubmissionDate>2010-07-19</SubmissionDate>
  <ProcessedDate>2010-07-23</ProcessedDate>
  <SubmittingParty>
    <UserID>hqs1</UserID>
    <SubmissionType>
      <SubmissionTypeName>Master General Permit</SubmissionTypeName>
      <SubmissionsAccepted>
        <SubmissionAccepted>
          <SubmissionAcceptedKey>
            <PermitRecordIdentifier>
              <PermitIdentifier>TXMGP1000</PermitIdentifier>
            </PermitRecordIdentifier>
            <SubmissionTransactionTypeCode>R</SubmissionTransactionTypeCode>
          </SubmissionAcceptedKey>
        </SubmissionAccepted>
      </SubmissionsAccepted>
    </SubmissionType>
    <SubmissionType>
      <SubmissionTypeName>Permitted Feature</SubmissionTypeName>
      <SubmissionsAccepted>
        <SubmissionAccepted>
          <SubmissionAcceptedKey>
            <PermittedFeatureRecordIdentifier>
              <PermitIdentifier>AK1100016</PermitIdentifier>
              <PermittedFeatureIdentifier>001</PermittedFeatureIdentifier>
            </PermittedFeatureRecordIdentifier>
            <SubmissionTransactionTypeCode>R</SubmissionTransactionTypeCode>
          </SubmissionAcceptedKey>
        </SubmissionAccepted>
      </SubmissionsAccepted>
    </SubmissionType>
  </SubmittingParty>
</SubmissionResponse>

```

The Rejected Transactions Report provides the same information included in the corresponding PDF report. Figure 3-5 – Rejected Transactions XML Report Screen Shot, displays a sample Rejected Transactions report.

**Figure 3-5 - Rejected Transactions XML Report Screen Shot**

```

<?xml version="1.0"?>
<SubmissionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.exchangenetwork.net/schema/icis/1">
  <TransactionIdentifier>_457b7249-cb09-4764-98ac-ec502443c52a</TransactionIdentifier>
  <SubmissionDate>2010-07-19</SubmissionDate>
  <ProcessedDate>2010-07-23</ProcessedDate>
  <SubmittingParty>
    <UserID>hqs1</UserID>
    <SubmissionType>
      <SubmissionTypeName>Limit Set</SubmissionTypeName>
      <SubmissionErrors>
        <SubmissionError>
          <SubmissionErrorKey>
            <LimitSetIdentifier>
              <PermitIdentifier>VALMITEST</PermitIdentifier>
              <PermittedFeatureIdentifier>001</PermittedFeatureIdentifier>
              <LimitSetDesignator>B</LimitSetDesignator>
            </LimitSetIdentifier>
            <SubmissionTransactionTypeCode>N</SubmissionTransactionTypeCode>
          </SubmissionErrorKey>
          <ErrorReport>
            <ErrorCode>LS110</ErrorCode>
            <ErrorTypeCode>Error</ErrorTypeCode>
            <ErrorDescription>Limit Set Modification Effective Date must exist.</ErrorDescription>
          </ErrorReport>
          <ErrorReport>
            <ErrorCode>LS120</ErrorCode>
            <ErrorTypeCode>Error</ErrorTypeCode>
            <ErrorDescription>Limit Set Modification Type Code is inactive.</ErrorDescription>
          </ErrorReport>
        </SubmissionError>
      </SubmissionErrors>
    </SubmissionType>
  </SubmittingParty>
</SubmissionResponse>

```

The Batch Transactions Summary XML Report provides the same information included in the corresponding PDF report, with the difference that the total accepted, total rejected, and percentage of accepted transactions are listed by ICIS User ID and then by Submission Type for each Batch, instead of by Submission Type only. Figure 3-6 – Batch Transactions Summary XMLReport Screen Shot, displays a sample Transactions Summary report.

**Figure 3-6 - Batch Transactions Summary XML Report Screen Shot**

```

<?xml version="1.0"?>
<SubmissionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.exchangenetwork.net/schema/icis/1">
  <TransactionIdentifier>_e3f8f43a-b67e-42d8-bf6d-192e69e26e10</TransactionIdentifier>
  <SubmissionDate>2010-07-02</SubmissionDate>
  <ProcessedDate>2010-07-15</ProcessedDate>
  <SubmittingParty>
    <UserID>HQS1</UserID>
    <SubmissionType>
      <SubmissionTypeName>Limit Set</SubmissionTypeName>
      <SubmissionSummary>
        <TransactionTypeTotals>
          <SubmissionTransactionTypeCode>N</SubmissionTransactionTypeCode>
          <TotalAcceptedTransactions>0</TotalAcceptedTransactions>
          <TotalRejectedTransactions>2</TotalRejectedTransactions>
        </TransactionTypeTotals>
        <TotalTransactions>2</TotalTransactions>
        <PercentTransactionsAccepted>0.0</PercentTransactionsAccepted>
      </SubmissionSummary>
    </SubmissionType>
  </SubmittingParty>
  <SubmissionType>
    <SubmissionTypeName>Permit Reissuance</SubmissionTypeName>
    <SubmissionSummary>
      <TransactionTypeTotals>
        <SubmissionTransactionTypeCode>X</SubmissionTransactionTypeCode>
        <TotalAcceptedTransactions>0</TotalAcceptedTransactions>
        <TotalRejectedTransactions>1</TotalRejectedTransactions>
      </TransactionTypeTotals>
      <TotalTransactions>1</TotalTransactions>
      <PercentTransactionsAccepted>0.0</PercentTransactionsAccepted>
    </SubmissionSummary>
  </SubmissionType>
  <TotalSubmissions>3</TotalSubmissions>
  <TotalPercentTransactionsAccepted>0.0</TotalPercentTransactionsAccepted>
</SubmissionResponse>

```

The following table, Table 3-3: Batch XML Response File Details, provides a brief description of the elements in the XML response files

**Table 3-3 - Batch XML Response Files Details**

| Element                | Information  | Example Value                        |
|------------------------|--|--------------------------------------|
| <b>All Files</b>       |  |                                      |
| TransactionIdentifier  | Identifier returned to the user from CDX after the batch submission.   | e3f8f43a-b67e-42d8-bf6d-192e69e26e10 |
| SubmissionDate         | Date of XML Submission   | 2010-06-01                           |
| ProcessedDate          | Date the batch was processed   | 2010-06-01                           |
| SubmittingParty        | Data related to a certain user submission                              | N/A (parent element)                 |
| UserID                 | ICIS User ID of the person submitting the XML                          | HQS1                                 |
| SubmissionType         | Data related to a certain data family                                  | N/A (parent element)                 |
| SubmissionTypeName     | Descriptive name for a submission type                                 | Basic Permit                         |
| <b>Accepted Report</b> |  |                                      |
| SubmissionsAccepted    | List of accepted submissions for a certain User ID and submission type | N/A (parent element)                 |
| SubmissionAccepted     | Data related to an accepted submission                                 | N/A (parent element)                 |
| SubmissionAcceptedKey  | Business key and transaction type for an accepted submission           | N/A (parent element)                 |

| Element                          | Information  | Example Value  |
|----------------------------------|--|--|
| < SubmissionType >Identifier     | Business key for a submission  | N/A (parent element).<br><br>Note: The actual name and content of the element depends upon the submission type   |
| SubmissionTransactionTypeCode    | Transaction type for a submission                                      | N  |
| AcceptedReport                   | Warning message for a submission                                       | N/A (parent element).  |
| InformationCode                  | Warning message code   | DMR300   |
| InformationTypeCode              | Warning message type   | Warning  |
| InformationDescription           | Warning message.   | Warning: The following Numeric Condition Quantity(ies) has a Percent Exceedence greater than 500%: Value will be listed for each column that has this error. |
| <b>Rejected Report</b>           |  |  |
| SubmissionsErrors                | List of rejected submissions for a certain User ID and submission type | N/A (parent element)   |
| SubmissionError                  | Data related to a rejected submission                                  | N/A (parent element)   |
| SubmissionErrorKey               | Business key and transaction type for a rejected submission            | N/A (parent element)   |
| < SubmissionType >Identifier     | Business key for a submission  | N/A (parent element).<br><br>Note: The actual name and content of the element depends upon the submission type   |
| SubmissionTransactionTypeCode    | Transaction type for a submission                                      | R  |
| ErrorReport                      | Error/Warning message for a submission                                 | N/A (parent element).  |
| ErrorCode                        | Error/Warning message code   | LS110  |
| ErrorTypeCode                    | Error/Warning message type   | Error  |
| ErrorDescription                 | Error/Warning message.   | Limit Set Modification Effective Date must exist.  |
| <b>Summary Report</b>            |  |  |
| SubmissionSummary                | Summary data for a certain User ID and submission type                 | N/A (parent element).  |
| TransactionTypeTotals            | Summary data per transaction type                                      | N/A (parent element).  |
| SubmissionTransactionTypeCode    | Transaction type   | C  |
| TotalAcceptedTransactions        | Total number of accepted transactions per transaction type             | 4  |
| TotalRejectedTransactions        | Total number of rejected transactions per transaction type             | 1  |
| TotalTransactions                | Total number of transactions per submission type.                      | 5  |
| PercentTransactionsAccepted      | Percentage of accepted transactions per submission type.               | 80.00%   |
| TotalSubmissions                 | Total number of submissions per User ID.                               | 12   |
| TotalPercentTransactionsAccepted | Total number of accepted transactions per User ID.                     | 97.25  |



### 3.1.3 Batch File-Level Error Report

The Batch File Level Error Report is a PDF file generated in case critical errors (e.g., an invalid ICIS user id populated in the XML files) prevented the processing of a received batch. The report includes the Transaction ID for the Batch, along with a list of the critical errors that occurred during batch processing. Figure 3-7 – Batch File-Level Error Report Screen Shot, displays a sample File-Level Error PDF report.

**Figure 3-7 - Batch File-Level Error Report Screen Shot**

#### **Batch \_3e134e25-5e7a-4d7d-abd9-6d2626ba279f has processed in ICIS.**

Some files could not be processed due to the following errors:

- Unable to parse file Unpermitted\_Facility.xml because 'xxx' is an invalid ICIS user
- Batch parsing failed

### 3.1.4 Batch File-Level Error XML Report

The Batch File Level Error XML Report provides the same information included in the corresponding PDF report. Figure 3-8 – Batch File-Level Error XML Report Screen Shot, displays a sample File-Level Error XML report.

**Figure 3-8 - Batch File-Level Error XML Report Screen Shot**

```
<?xml version="1.0"?>
<SubmissionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.exchangenetwork.net/schema/icis/1">
  <TransactionIdentifier>_3e134e25-5e7a-4d7d-abd9-6d2626ba279f</TransactionIdentifier>
  <SubmissionDate>2010-10-19</SubmissionDate>
  <ProcessedDate>2010-10-19</ProcessedDate>
  <FileSubmissionErrors>
    <FileSubmissionError>
      <FileErrorReport>
        <ErrorTypeCode>Error</ErrorTypeCode>
        <ErrorDescription>Unable to parse file Unpermitted_Facility.xml because 'xxx' is an invalid ICIS user</ErrorDescription>
      </FileErrorReport>
      <FileErrorReport>
        <ErrorTypeCode>Error</ErrorTypeCode>
        <ErrorDescription>Batch parsing failed</ErrorDescription>
      </FileErrorReport>
    </FileSubmissionError>
  </FileSubmissionErrors>
</SubmissionResponse>
```

**Table 3-4 - Batch File Level Error XML File Details**

| Element               | Information  | Example Value                        |
|-----------------------|--|--------------------------------------|
| TransactionIdentifier | Identifier returned to the user from CDX after the batch submission. | 457b7249-cb09-4764-98ac-ec502443c52a |
| SubmissionDate        | Date of XML submission   | 2010-08-01                           |
| ProcessedDate         | Date the batch was processed   | 2010-08-01                           |
| FileSubmissionErrors  | List of file-level errors occurred during batch processing           | N/A (parent element).                |
| FileSubmissionError   | List of file-level errors occurred during batch processing           | N/A (parent element).                |

|                  |                                    |                       |
|------------------|------------------------------------|-----------------------|
| FileErrorReport  | Data related to a file-level error | N/A (parent element). |
| ErrorTypeCode    | Error message type                 | Error                 |
| ErrorDescription | Error message.                     | Batch parsing failed  |

### 3.2 NETDMR RESPONSE

The NetDMR response report is an XML file including the errors that occurred for each DMR transaction in a batch submission. In case critical errors prevented the batch from being processed, the report lists the file-level errors that occurred. The response file is sent from ICIS to NetDMR, where it can be retrieved by NetDMR Users after login. Figure 3-9 – NetDMR Report Screen Shot, displays a sample NetDMR report.

**Figure 3-9 - NetDMR Report Screen Shot**

```
<?xml version="1.0"?>
<SubmissionResponse xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.exchangenetwork.net/schema/NetDMR/1">
  <TransactionIdentifier>_b631e454-195d-4230-95a6-d3e51cad00a8</TransactionIdentifier>
  <SubmissionDate>2010-07-20</SubmissionDate>
  <CreationDate>2010-07-20</CreationDate>
  <SubmissionErrors>
    <SubmissionError>
      <SubmissionErrorKey>
        <DMRParameterIdentifier>
          <PermitIdentifier>AK1234567</PermitIdentifier>
          <PermittedFeatureIdentifier>001</PermittedFeatureIdentifier>
          <LimitSetDesignator>XX</LimitSetDesignator>
          <MonitoringPeriodEndDate>2010-02-28</MonitoringPeriodEndDate>
          <ParameterCode>30383</ParameterCode>
          <MonitoringLocationTypeCode>J</MonitoringLocationTypeCode>
          <SeasonIdentifier>0</SeasonIdentifier>
        </DMRParameterIdentifier>
        <SubmissionTransactionTypeCode>C</SubmissionTransactionTypeCode>
      </SubmissionErrorKey>
      <ErrorReport>
        <ErrorCode>DMR080</ErrorCode>
        <ErrorTypeCode>Error</ErrorTypeCode>
        <ErrorDescription>A Parameter does not exist that matches the data entered.</ErrorDescription>
      </ErrorReport>
    </SubmissionError>
  </SubmissionErrors>
</SubmissionResponse>
```

Table 3-5 – NetDMR Response File Details summarizes the tags included in the XML response file, providing a brief description and an example for each tag.

**Table 3-5 - NetDMR Response File Details**

| Element               | Information  | Example Value                         |
|-----------------------|--|---------------------------------------|
| TransactionIdentifier | identifier returned to user from CDX after the DMR batch submission. | _b631e454-195d-4230-95a6-d3e51cad00a8 |
| SubmissionDate        | Date of batch submission   | 2010-08-01                            |
| CreationDate          | Date the batch was received  | 2010-08-01                            |
| SubmissionsErrors     | List of rejected DMR transactions                                    | N/A (parent element)                  |
| SubmissionError       | Data related to a rejected DMR transaction                           | N/A (parent element)                  |



| Element                       | Information  | Example Value   |
|-------------------------------|--|---|
| SubmissionErrorKey            | Business key and transaction type for a rejected DMR transaction | N/A (parent element)                                      |
| DMRParameterIdentifier        | Business key for a DMR transaction                               | N/A (parent element).                                     |
| PermitIdentifier              | Permit NPDES ID  | AK1234567   |
| PermittedFeatureIdentifier    | Permitted Feature ID   | 001   |
| LimitSetDesignator            | Limit Set Designator   | A   |
| MonitoringPeriodEndDate       | Monitoring Period End Date                                       | 2010-10-31  |
| ParameterCode                 | Parameter Code   | 00950   |
| MonitoringLocationTypeCode    | Monitoring Location Code   | J   |
| SeasonIdentifier              | Season Identifier  | 0   |
| SubmissionTransactionTypeCode | Transaction type for the DMR transaction                         | C   |
| ErrorReport                   | Error/Warning message data for the DMR submission                | N/A (parent element).                                     |
| ErrorCode                     | Error/Warning message code for the DMR submission                | DMR080  |
| ErrorTypeCode                 | Error/Warning message type for the DMR submission                | Error   |
| ErrorDescription              | Error/Warning message for the DMR submission                     | A Parameter does not exist that matches the entered data. |

### 3.3 ICIS BATCH DMR HYBRID RESPONSE

ICIS Batch DMR Hybrid Users receive a PDF file that simply informs whether the batch was processed in ICIS. If it was not, the file lists the files that were not processed along with errors. Figure 3-10 – ICIS Batch DMR Hybrid Report Screen Shot, displays a sample ICIS Batch Hybrid report.

**Figure 3-10 - ICIS Batch DMR Hybrid Report Screen Shot**

**Batch \_7a3da61e-674d-4472-8a0e-3db946474e5d has processed in ICIS. Please log on to ICIS to retrieve the audit reports.**

### 3.4 CDX RESPONSE

After a submitter has manually uploaded a file using the CDX Web form or electronically submitted the file through its Node, CDX will determine if it is a zip compressed file containing one or more XML instance documents, extract the zipped file, perform virus scans on the extracted file(s), then use the namespace in the Document tag to locate the correct version of the schema and check them against the ICIS-NPDES schema for well-formedness and validity before routing them to ICIS-NPDES Batch.

If one or more XML instance documents within the same zipped file fails the virus scan or schema validation, they will not be sent to ICIS-NPDES Batch but will be rejected by CDX with up to 100 errors documented in a downloadable file called “submission-metadata.xml”. All other files that pass the virus scan and schema validation checks will be sent to ICIS-NPDES Batch.

Figure 3-11 displays an example of the contents of “submission-metadata.xml” of a zipped file containing Three XML submission files, one that was able to pass a schema validation check, one that was unable to pass a schema validation check, and another that had an invalid Header tag.

```
<?xml version="1.0" encoding="UTF-8"?>
<submission>
  <file>
    <name>Construction.xml</name>
    <email>ecleckler@adem.state.al.us</email>
    <status>VALID</status>
    <result>Congratulations! The document,d:\SOAPServer\DataFolder\_0941bd02-
dfd7-4952-a922-175e322754400.unzipped\Construction.xml, is valid according to
the schema.</result>
  </file>
  <file>
    <name>Industrial.xml</name>
    <email>ecleckler@adem.state.al.us</email>
    <status>INVALID</status>
    <result> The document,d:\SOAPServer\DataFolder\_0941bd02-dfd7-4952-a922-
175e322754400.unzipped\Industrial.xml, contains the following error(s): Error
at line 4 column 40 : The 'c:\ICISSchemas\SchemasVersion3.0:Id' element is
invalid - The value 'UUSTaffer1123456789012345678901' is invalid according to
its datatype 'c:\ICISSchemas\SchemasVersion2.0:StringMin3Max30Type' - The
actual length is greater than the MaxLength value.</result>
  </file>
  <file>
    <name>General.xml</name>
    <status>INVALID</status>
    <statusDetails>E_ParseHeader:Error parsing EN Header
Document.</statusDetails>
    <result/>
  </file>
</submission>
```

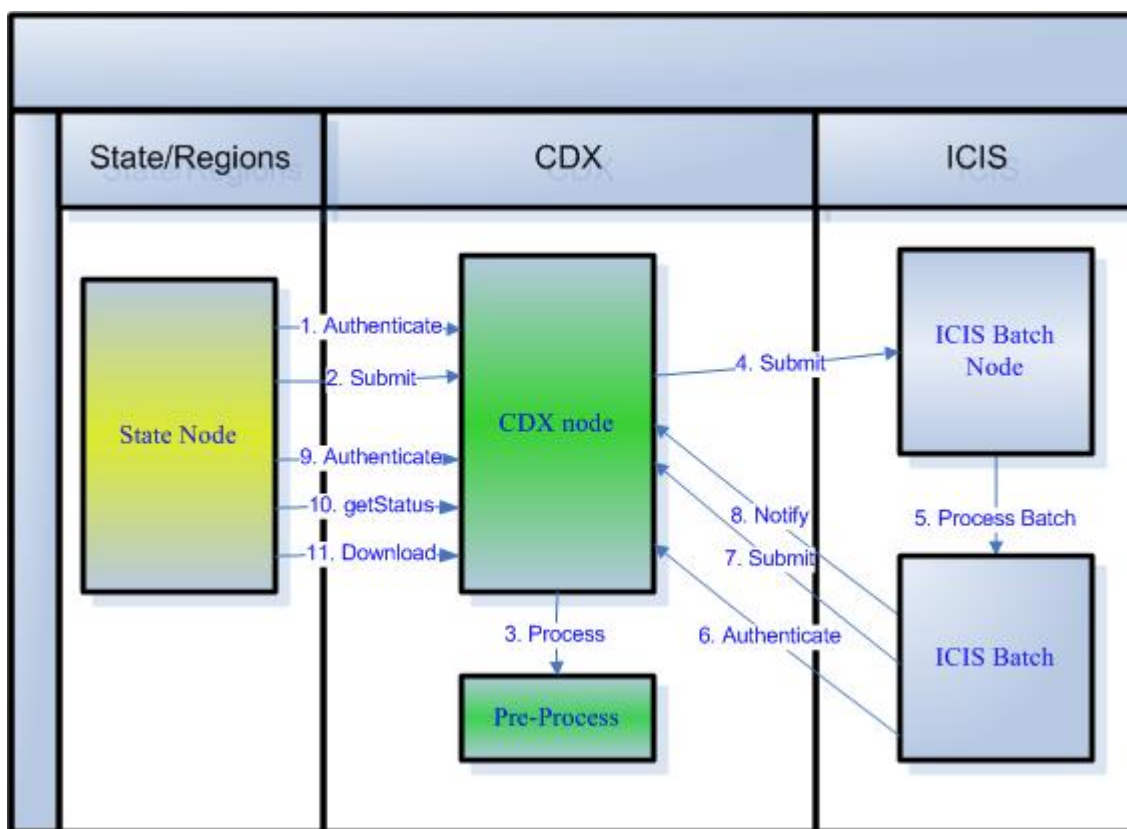
## 4. ICIS–NPDES BATCH FLOW DESCRIPTION

A node is a service provider on the Exchange Network conforming to the Network Node Functional Specification. The Network Node Functional Specification describes the behavior and interfaces of the service provider component. The specification is available on the Exchange Network website ([www.exchangenetwork.net](http://www.exchangenetwork.net)). CDX deploys a node to accept batch submissions from the state users. The ICIS–NPDES Batch system deploys a node in the ICIS–NPDES environment to allow the receipt of batch submissions from CDX.

### 4.1 FILE EXCHANGE IN NODE TO NODE CONFIGURATION

Figure 4-1 – File Exchange in Node to Node Configuration, describes the steps involved in exchanging the files between the states, CDX and ICIS–NPDES Batch for this configuration. In this deployment, an exchange node is configured in the state, CDX and ICIS–NPDES Batch environments. The following steps describe the details of the batch submissions in this configuration:

**Figure 4-1 - File Exchange in Node to Node Configuration**



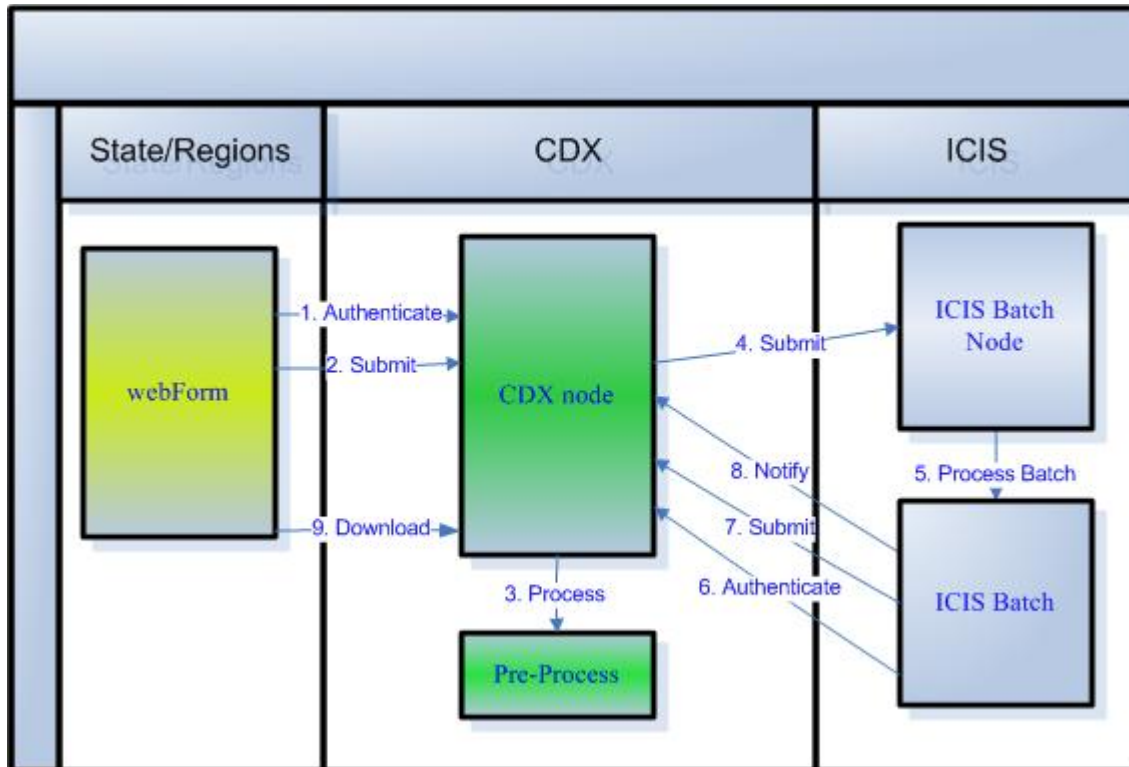
1. Before submitting a batch document to CDX, the state node uses the *Authenticate* method by supplying a CDX user id and password to authenticate. Upon successful authentication CDX returns a securityToken to the state node. If the supplied user id, password combination is invalid, a SOAP fault with “E\_UnknownUser” as the error code is returned to the invoker.

2. The state node uses the *Submit* method to submit the batch data and the issued security Token (from Step 1) to CDX.
  - a. CDX validates the security token. If the security token is invalid, the entire batch is rejected, a SOAP fault is returned to the client, and batch processing ends. Otherwise processing continues to step b.
  - b. CDX archives the received batch files. They will be maintained in an easily accessible archive for one month. After a month, the archived batches will be moved to a secondary storage device like a tape. Tapes will be maintained for a total of five years from the batch receipt date.
  - c. CDX saves the submitted XML files in the batch. The batch status is set to “Received” and a Transaction ID is generated and returned to the client. Non-XML files are silently ignored. If no XML files are submitted, CDX rejects the batch and no further processing takes place.
3. Upon receiving the files from the state node, CDX unzips the compressed batch submissions and performs the following process:
  - a. Invokes its virus scanning service to validate that the submitted files are free from viruses. If all of the files contain viruses, the entire batch submission is rejected and the batch status is set to “Failed”. Otherwise, virus-laden files are silently ignored, and processing continues with the valid files.
  - b. Valid files from step “a” are validated against the target XML schema. If all of the files are invalid, the entire batch submission is rejected and the batch status is set to “Failed”. If any files are valid, the batch status remains set to “Received” and the valid files continue processing at step c.
  - c. The QA server at CDX creates a report with the status of the validations. This report can be downloaded using the “*Download*” web method on CDX.
  - d. If steps “a” through “c” complete successfully, valid files from the batch are archived. Valid batches to be passed to ICIS–NPDES will be archived in an easily accessible storage area for one month. At the end of the month, the archived batches will be permanently deleted.
  - e. If steps a through d complete successfully, the batch status is set to “Pending” and the process continues at step 4.
4. CDX submits the valid files from the batch to the ICIS–NPDES Batch node by invoking the *Submit* method on the ICIS–NPDES Batch node. The ICIS–NPDES Batch node stores the received files and the corresponding Transaction ID supplied by CDX for parsing. The received files are stored until the batch process for the corresponding files is complete.
5. ICIS–NPDES Batch validates the ICIS–NPDES user id submitted with the batch submission.
  - a. If the user id is invalid, an error message is captured in the ICIS–NPDES batch logs. If all of the submitted files have invalid user ids, the process continues at step 6.

- b. If the user id is valid for at least one of the files in the submission, valid files are parsed, and payloads are extracted and saved in the ICIS Batch Operation Database.
6. In preparing to submit the processing details to CDX, the ICIS–NPDES Batch node authenticates itself by invoking the *Authenticate* method on CDX. On successful authentication, CDX returns a security Token.
7. Depending on the Batch origin, ICIS–NPDES Batch creates the processing report according to the following guidelines:
  - a. Critical errors (e.g., an invalid ICIS user id populated in the XML files) that prevented the processing of a received batch are reported in the response files, as described in Sections 3.1-3.3. If no critical errors occurred, processing continues at step b.
  - b. After the batch is processed, ICIS–NPDES Batch creates response files based on the batch origin, as described in Section 3.
  - c. The ICIS–NPDES Batch node uses the *Submit* method to send the documents created in step “a” or “b” to CDX. CDX archives the received documents.
8. After submitting the processing report to CDX, ICIS–NPDES Batch uses the *Notify* method to update the processing status of the batch at CDX. CDX updates the batch status to “Completed”.
9. In preparation for retrieving the batch processing status and the processing documents the state node obtains a securityToken by supplying the user id and password to the *Authenticate* method on CDX. As described in step 1, the *Authenticate* method returns a SOAP fault in case of failed authentication.
10. Using the Transaction ID provided by CDX in step 2, the state node retrieves the batch status at regular intervals using the *getStatus* method.
11. When the batch status is “Completed”, the state node downloads the batch processing documents for that specific Transaction ID from CDX using the *Download* method.

## 4.2 FILE EXCHANGE IN WEB FORM TO NODE CONFIGURATION

In this configuration, state users submit batches using the ICIS–NPDES CDX Web Form. Before submitting a batch, the user logs into CDX using a pre-established user id and password. Upon successful authentication, the user can submit the files to CDX. All batch files should be compressed prior to submitting to CDX. CDX returns a Transaction ID after receiving the batch submission. Figure 3-2 – File Exchange in Client to Node Configuration, describes the steps involved in exchanging the files between the states, CDX and ICIS–NPDES Batch for this configuration.

**Figure 4-2 - File Exchange in Client to Node Configuration**

1. Before submitting a document to CDX, the state user logs into CDX using the ICIS–NPDES CDX Web Form.
2. Upon successful login, the state user uploads a batch to CDX.
  - a. CDX validates the security token. If the security token is invalid, the entire batch is rejected, a SOAP fault is returned to the client, and batch processing ends. Otherwise processing continues to Step b.
  - b. CDX archives the received batch files. They will be maintained in an easily accessible archive for one month. After a month, the archived batches will be moved to a secondary storage device like a tape. Tapes will be maintained for a total of five years from the batch receipt date.
  - c. CDX saves the submitted XML files in the batch. The batch status is set to “Received” and a Transaction ID is generated and returned to the client. Non-XML files are silently ignored. If no XML files are submitted, CDX rejects the batch and no further processing takes place.
3. Upon receiving the files from the ICIS–NPDES CDX Web Form CDX unzips compressed batch submissions and performs the following process:
  - a. Invokes its virus scanning service to validate that the submitted files are free from viruses. If all of the files contain viruses, the entire batch submission is rejected

- and the batch status is set to “Failed”. Otherwise, virus-laden files are silently ignored, and processing continues with the valid files.
- b. Valid files from step “a” are validated against the target XML schema. If all of the files are invalid, the entire batch submission is rejected and the batch status is set to “Failed”. If any files are valid, the batch status remains set to “Received” and the valid files continue processing at step c.
  - c. The QA server at CDX creates a report with the status of the validations. This report can be downloaded to view the validation errors of the submitted files.
  - d. If steps “a” through “c” complete successfully, valid files from the batch are archived. Valid batches to be passed to ICIS–NPDES will be archived in an easily accessible storage area for one month. At the end of the month, the archived batches will be permanently deleted.
  - e. If steps a through d complete successfully, the batch status is set to “Pending” and the process continues at step 4.
4. CDX submits the valid files from the batch to the ICIS–NPDES Batch node by invoking the *Submit* method on the ICIS–NPDES Batch node. The ICIS–NPDES Batch node stores the received files and the corresponding Transaction ID supplied by CDX for parsing. The received files are stored until the batch process for the corresponding files is complete.
  5. ICIS–NPDES Batch validates the user id submitted with the batch submission.
    - a. If the user id is invalid, an error message is captured in the ICIS–NPDES batch logs. If all of the submitted files have invalid user ids, the process continues at step 6.
    - b. If the user id is valid for at least one file in the submission, ICIS–NPDES Batch parses and processes the valid files according to the predefined processing logic.
  6. In preparing to submit the processing details to CDX, the ICIS–NPDES Batch node authenticates itself by invoking the *Authenticate* method on CDX. On successful authentication, CDX returns a securityToken.
  7. Depending on the Batch origin, ICIS–NPDES Batch creates the processing report according to the following guidelines:
    - a. Critical errors (e.g., an invalid ICIS user id populated in the XML files) that prevented the processing of a received batch are reported in the response files, as described in Sections 3.1-3.3. If no critical errors occurred, processing continues at step b.
    - b. After the batch is processed, ICIS–NPDES Batch creates response files based on the batch origin, as described in Section 3.
    - c. The ICIS–NPDES Batch node uses the *Submit* method to send the documents created in step “a” or “b” to CDX. CDX archives the received documents.
  8. After submitting the processing report to CDX, ICIS–NPDES Batch node uses the *Notify* method to update the processing status of the batch at CDX. CDX updates the batch status to “Complete”.



9. The state user logs in to the ICIS–NPDES CDX Web Form, which shows a status of “Complete”, and selects the transaction history to view all the processing details and documents related to the batch transaction including the ICIS–NPDES Batch Processing Status PDF.



## 5. ICIS–NPDES BATCH NODE WEB METHODS

---

ICIS–NPDES Batch implements the following Web services from the methods defined in the Network Node Functional Specification to accept batch submissions from CDX:

- Submit - CDX invokes this Web method to submit files to ICIS–NPDES Batch.

### 5.1 SUBMIT WEB METHOD

#### **Description:**

Submit is a utility method used to send one or more files to a service provider. A positive response from the node indicates that the files were successfully received. An exception message means that the submitted documents were not delivered to the recipient. The ICIS–NPDES Batch node returns the Transaction ID it received in the submission to indicate the files were received successfully.

#### **Arguments:**

The Submit method has four arguments

- securityToken: A security ticket issued by the service provider.
- transactionId: CDX sends a unique transactionId along with the submission.
- dataflow: The name of the target dataflow. In this case, “ICIS–NPDES”.
- documents: An array of documents of type nodeDocument. This argument contains the XML documents for the specific Transaction ID.

#### **Return:**

- When successful, the Submit method returns the transactionId it received from CDX. In case of an error, the ICIS–NPDES Batch node returns a SOAP Fault with the details of the errors that occurred while receiving the batch.

## 6. CDX NODE WEB METHODS

---

CDX implements the following Web methods from the methods defined in the Network Node Functional Specification to facilitate authentication and accept batch processing status from the ICIS–NPDES Batch node:

- **Authenticate** - ICIS–NPDES Batch obtains a securityToken before initiating Submit and Notify.
- **Submit** - ICIS–NPDES Batch Submits an ICIS–NPDES Batch Processing Status PDF document with Batch/File level errors or a notice of successful processing to CDX.
- **Notify** - ICIS–NPDES Batch notifies CDX of the final Batch Processing Status.

### 6.1 AUTHENTICATE WEB METHOD

**Description:**

The Authenticate method authenticates a user using the supplied credentials. Upon successful authentication, it returns a securityToken. The securityToken is used while invoking the Notify and Submit methods on CDX.

**Arguments:**

The Authenticate method has three arguments:

- **userId**: The User id to be authenticated. The ICIS–NPDES Batch node sends the user id assigned to the ICIS–NPDES Batch node.
- **credential**: User password for accessing CDX.
- **authenticationMethod**: Specifies the authentication method to be used. In this case, it will be "password".

**Return:**

- Upon successful authentication, the service provider returns a securityToken wrapped in a SOAP message. If authentication fails, a SOAP fault message with error details is returned.

### 6.2 SUBMIT WEB METHOD

**Description:**

ICIS–NPDES Batch invokes the Submit method on CDX to send the processing status document to CDX. In case of critical errors which cause a file or batch to be rejected (e.g., invalid user id), the ICIS–NPDES Batch Processing Status Document will report these errors. If no critical errors are received while processing a batch, the ICIS–NPDES Batch Processing Status Document will contain a message indicating the successful completion of processing for that batch.

**Arguments:**

The Submit method has eight arguments:

- URL: URL for CDX.
- securityToken: A security token issued by the service provider (NAAS).
- transactionId: The transactionId associated with the processed batch. CDX assigns a unique transactionId for each batch submitted to ICIS–NPDES Batch.
- dataflow: The target dataflow name. In this case, “ICIS–NPDES”.
- flowOperation: The target dataflow operation. In this case no value is passed for this argument.
- recipient: The target recipient. In this case no value is passed for this argument.
- notification URI: The target notification URI. In this case no value is passed for this argument.
- documents: An array of documents of type nodeDocument. ICIS–NPDES Batch will submit the ICIS–NPDES Batch Processing Status Document.

**Return:**

- The Submit method returns the transactionId submitted while invoking the Submit method. If the submission fails, a SOAP fault with details of the error will be returned.

### 6.3 NOTIFY WEB METHOD

**Description:**

The Notify method can be used for three purposes: document notification, event notification, and status notification. ICIS–NPDES Batch uses the notify method to inform CDX about the status of a processed batch for a given Transaction ID.

**Arguments:**

The Notify method has four arguments:

- securityToken: A security ticket issued by the service provider (NAAS).
- nodeAddress: Address of node to notify. No value is passed for status notification.
- dataflow: This parameter contains a URL to indicate the type of notification. There are three kinds of notifications, an event, a status or a document. ICIS–NPDES Batch node uses “<http://www.exchangenetwork.net/node/status>” to indicate status notification.
- documents: An array of documents of type nodeDocument. ICIS–NPDES Batch will provide a nodeDocument with the transaction ID and batch processing status of “Processed”.

**Return:**

- For status notification, CDX returns a string other than transactionId signaling the acceptance of the status.

## 7. WEB METHODS USED BY STATE NODE

---

State nodes will invoke the following Web methods on CDX to submit batches to CDX, get status of a batch submission and download any processing status documents including error reports from CDX:

- **Authenticate** – The state node obtains a securityToken before initiating Submit, GetStatus, or Download.
- **Submit** – The state node invokes this method on CDX to submit a batch.
- **GetStatus** – The state node invokes this method on CDX to obtain the batch status.
- **Download** – The state node invokes this method on CDX to download the processing status documents related to a batch submission.

### 7.1 AUTHENTICATE WEB METHOD

#### **Description:**

The State node invokes the Authenticate method on CDX to authenticate a user using the supplied credentials like user id and password. Upon successful authentication, CDX returns a securityToken. The securityToken is used while invoking the Submit, GetStatus and Download methods on CDX.

#### **Arguments:**

The Authenticate method has three arguments:

- **userId**: The User id to be authenticated. The state node sends CDX userid assigned to the state node.
- **credential**: User password for accessing CDX.
- **authenticationMethod**: Specifies the authentication method to be used. In this case, it will be "password".

#### **Return:**

- Upon successful authentication, the service provider returns a securityToken wrapped in a SOAP message. If authentication fails, a SOAP fault message with error details is returned.

### 7.2 SUBMIT WEB METHOD

#### **Description:**

The state node invokes the Submit method on CDX to submit a batch.

#### **Arguments:**

The Submit method has eight arguments:

- **URL**: URL for CDX.
- **securityToken**: A security token issued by the service provider (NAAS).

- transactionId: The transactionId associated with the batch. While submitting batches this is always empty.
- dataflow: The target dataflow name. In this case, “ICIS–NPDES”.
- flowOperation: The target dataflow operation. In this case no value is passed for this argument.
- recipient: The target recipient. In this case no value is passed for this argument.
- notification URI: The target notification URI. In this case no value is passed for this argument.
- documents: An array of documents of type nodeDocument. ICIS–NPDES The state node will submit XML document(s).

**Return:**

- The Submit method returns the transactionId associated with the batch submission. If the submission fails, a SOAP fault with details of the error will be returned.

### 7.3 GET STATUS WEB METHOD

**Description:**

The GetStatus web method is used for retrieving the current status of the batch. As described in section 3.1, a batch could be in Receiving, Pending, Processing, Completed, or Failed status.

**Arguments:**

The GetStatus method has two arguments:

- securityToken: A security ticket issued by the service provider (NAAS).
- transactionId: The transactionId associated with the batch. When a batch is submitted to CDX, a transactionId is returned for successful submissions.

**Return:**

- This method returns a description of the current status of the batch if the operation is successful. Possible responses correspond to the batch states described in Step 1 through 7 of Section 3.1: “Receiving”, “Pending”, “Processing”, “Completed”, or “Failed” This method returns a SOAP Fault with an error code of E\_TransactionId if the transaction ID is invalid; it returns a SOAP Fault with an error code of E\_InvalidToken or E\_TokenExpired if the securityToken is invalid or expired.

### 7.4 DOWNLOAD WEB METHOD

**Description:**

The state node invokes the Download method on CDX to download all of the processing status documents associated with a batch submission. These documents include the QA report (contains XML validation errors if any), Batch processing status and processing error

reports from ICIS-NPDES. Prior to invoking this method it is recommended to call the `GetStatus` method to verify the batch status.

**Arguments:**

The Download method has four arguments:

- `securityToken`: A security ticket issued by the service provider (NAAS).
- `transactionId`: The `transactionId` associated with the batch. When a batch is submitted to CDX, a `transactionId` is returned for successful submissions.
- `dataflow`: The target dataflow name. In this case, "ICIS-NPDES".
- `documents`: An array of documents of type `nodeDocument`. This should be set to empty.

**Return:**

- The response contains a dataflow identifier and a set of documents. Documents transmitted can be either embedded payloads or separate attachments. If the Download fails, a SOAP fault with details of the error will be returned.

## **APPENDIX A: ICIS–NPDES BATCH SUBMISSION TYPES**

ICIS–NPDES Batch supports the submission types listed below. Instructions to create XML instance submission documents for different submission types are contained in the ICIS-NPDES Batch User Guide.

- Discharge Monitoring Report
- Basic Permit
- Biosolids Permit
- CAFO Permit
- CSO Permit
- General Permit
- Limit Set
- Limits
- Master General Permit
- Narrative Condition Schedule
- Parameter Limits
- Permit Reissuance
- Permit Tracking Event
- Permitted Feature
- POTW Permit
- Pretreatment Permit
- SW Construction Permit
- SW Industrial Permit
- SW MS4 Large Permit
- SW MS4 Small Permit
- Unpermitted Facility
- Compliance Monitoring
- Biosolids Program Report
- CAFO Annual Report
- Compliance Monitoring Linkage
- Compliance Schedule
- CSO Event Report
- DMR Program Report Linkage
- DMR Violation
- Effluent Trade Partner
- Enforcement Action Milestone
- Enforcement Action Violation Key
- Formal Enforcement Action
- Informal Enforcement Action
- Local Limits Program Report
- Pretreatment Performance Summary
- Schedule Event Violation

- Single Event Violation
- SSO Annual Report
- SSO Event Report
- SSO Monthly Event Report
- SW Event Report
- SW MS4 Program Report