

```
Default="uniquetargetNamespace="http://v
xmlns:xsd="http://v
xmlns:nei="http://w
elementFormDefault=
version="3.0">
<xsd:include schema
- <!--
XML 3.0
Start of Schema Header
-->
<xsd:annotation>
<xsd:documentation>
Point</xsd:docu
Point data</xsd:documentation>
Available:http://
cumentation>
<xsd:documentation
input format</x
encoding="UTF-8" ?
user</xsd:docu
<xsd:documentation
Agency
http://www.w3.org/.1.0" encoding="UT
http://www.epa.gov/ea
Default="qualified" attribute="http://
">
http://www.
schemaLocation="EN_NEI_http://www.e
Default="qual
Header
'>
chemaLocatio
on>
mentation>Schema Name: NEI.der
xsd:documentation>
mentation>Current Version
e:http://www.epa.gov/excha
tion>Sch
ion>
mentation>Description: The NEI n>Cur
rmat</xsd:documentation>
/ww
mentation>Application: Varies by
sd:documentation>
>Des
mentation>Developed By: Environme
App
ding="UTF-8" ?>
http://www.epa.gov/exchangenetw
/www.w3.org/2001/XMLSchema"
/www.epa.gov/exchangenetwork"
t="qualified" attributeFormDefault="unq
aLocation="EN_NEI_Common_v3_0.xsd
```

```
ion>Schema Name: NEI XML 3.0
cumentation>
on>Current Version
//www.epa.gov/exchangenetwork<
>Description: The NEI XML 3.0 Poin
f:documentation>
Application: Varies by
```

ATTAINS

Flow Configuration Document

ATTAINS Version 1.0

ATTAINS DET Version 1.0

October 16, 2018

Developed for:

United States Environmental Protection Agency
Office of Environmental Information
1200 Pennsylvania Avenue, NW
Washington, DC 20460

Developed by:

Gold Systems, Inc.
2121 McClelland Street
Salt Lake City, Utah 84106

Environmental Information

exchange
Network

Table of Contents

1.0	FLOW COMPONENT AND FCD VERSION HISTORY.....	1
2.0	INTRODUCTION	2
2.1	BACKGROUND	2
2.2	DOCUMENT PURPOSE	2
2.3	HOW TO USE THIS FCD.....	2
3.0	ATTAINS WEB SERVICE METHODS AT CDX	4
3.1	REGISTRATION / AUTHORIZATION	4
3.2	AUTHENTICATE	4
3.3	SUBMIT.....	4
3.4	GETSTATUS.....	5
3.5	DOWNLOAD.....	5
3.6	QUERY.....	6
3.7	SOLICIT	6
3.8	ATTAINS END POINTS	7
4.0	ATTAINS SUBMISSION FLOW AND FEEDBACK.....	7
5.0	ATTAINS SUBMISSION DATA PROCESSING	8
5.1	UPDATE-INSERT	9
5.1.1	<i>Specific Functionality</i>	9
5.2	DELETE.....	13
6.0	ATTAINS SUBMISSION EXCHANGE NETWORK DOCUMENT.....	14
6.1	OVERVIEW.....	14
6.2	HEADER (SECTION)	14
6.3	PAYLOAD (SECTION)	17
6.4	FILE STRUCTURE FOR A SUBMISSION PAYLOAD	17
7.0	ATTAINS QUERY & SOLICIT FLOW AND FEEDBACK.....	18
7.1	QUERY PROCESS.....	19
7.2	SOLICIT PROCESS	19
7.3	DATA PUBLISHING SERVICES USING REST	20
8.0	ATTAINS QUERY/SOLICIT REQUESTS	22
9.0	APPENDIX A – ATTAINS BUSINESS AND VALIDATION RULES	28
10.0	APPENDIX B – EXAMPLE PROCESSING REPORTS.....	34
11.0	APPENDIX C - ATTAINS SCHEMA DIAGRAM.....	36

1.0 Flow Component and FCD Version History

Component	ATTAINS Version	Date	Changed	Description of Change
Entire Document	1.0	5/5/2017	RTI/Gold Systems	Initial Version
5.0 ATTAINS Submission	1.0	10/16/2018	Gold Systems	Description of new cycle submission process

2.0 Introduction

2.1 Background

The Office of Water has embarked on a multi-year effort called the Water Quality Framework (WQF) to better integrate existing data systems to support water quality managers and to streamline water quality assessment and reporting to provide a more complete picture of the nation's water quality. Under the WQF, the Assessment TMDL Tracking and Implementation System (ATTAINS) is the first system to undergo improvements, which supports state reporting under Clean Water Act Sections 303(d) and 305(b).

The purpose of this effort is to re-design and develop ATTAINS to improve the tracking of state reported information under the Clean Water Act (CWA) Sections 303(d) and 305(b) Integrated Reporting process, which includes:

- Section 303(d) lists of impaired and threatened waters that need a TMDL;
- Section 305(b) assessment and attainment status of waters; and
- State-wide statistical survey assessments of the extent of waters supporting the goals of the Act as reflected in state water quality standards.
- Total Maximum Daily Loads (TMDLs)

The ATTAINS data system supports state reporting under CWA Sections 303(d) and 305(b). Beginning with the 2002 Integrated Reporting cycle to date, EPA recommended that states not only submit to EPA hard copy Integrated Report, but also the associated tabular and geospatial data. Due in part to the increased emphasis being placed on electronic reporting, the CWA Sections 303(d) and 305(b) programs have embarked on the ATTAINS redesign to fully transition states to electronic reporting by the 2018 Integrated Reporting cycle.

The Exchange Network provides a means for states to submit data directly from their databases to EPA via a pre-defined and agreed upon format. The Exchange Network also provides methods for states to retrieve data from EPA data systems to keep their databases in sync with EPA systems. This is all accomplished through agreed-upon web services and XML schema (more information on the Exchange Network is available at: <http://www.exchangenetwork.net>). The ATTAINS system provides the ability for states to both submit data to EPA and retrieve data from EPA via the Exchange Network. These methods would be available for states that have their own data systems for tracking 303(d)/305(b)/TMDL data, and who are not using the ATTAINS Cloud system. The data flow supports the following Exchange Network services: Submit, Query, and Solicit. The data flow is designed in such a way so that the data flow can be automated, with automated data loading, validation, and notification.

2.2 Document Purpose

This Flow Configuration Document (FCD) is intended to define the supported data services, the approaches and processes that are used to exchange information over the Exchange Network via web services technology. In addition, the FCD serves as a guide for trading partners in understanding the details and challenges associated with a data flow.

The purpose of this FCD is to describe the operation of the ATTAINS Network Exchange using XML-based data submissions through Node-to-Node or Client-to-Node transfers. The focus of the document is the core ATTAINS data flow between the state, local, and tribal agencies and EPA.

2.3 How to use this FCD

This FCD provides guidance to implement an XML/web-service based model for data submission. For the ATTAINS project, there are two types of submissions, Update-Insert and Delete. This FCD expands upon the ATTAINS Schema and introduces the implementation of the Document Header.

This document includes the following main sections:

ATTAINS Service Methods at CDX

This section describes the Web Service Methods at CDX used by the ATTAINS Data Flow.

ATTAINS Submission Flow and Feedback

This section describes the flow of the submission steps, the status changes that occur as a submission flows through CDX and the ATTAINS System, and the responses provided by the major components of the system.

ATTAINS Submission Data Processing

This section describes the details of the processing of an ATTAINS submission once it has been passed to the ATTAINS system.

ATTAINS Submission File

This section describes how the ATTAINS Network Exchange makes use of the Exchange Network Header Document to describe the payload content of a Network message. This submission structure is used for both Node-to-Node and Client-to-Node submissions.

ATTAINS Query & Solicit Flow and Feedback

This section describes the various Query and Solicit services that are supported, their input parameters, and their returned data elements.

ATTAINS Query and Solicit Data Processing

This section describes the details of the processing of an ATTAINS submission once it has been passed to the ATTAINS system.

Appendix A: ATTAINS Business and Validation Rules

This section simply refers the reader to the ATTAINS Rules Registry for all ATTAINS business and validation rules.

Appendix B: Example Processing Reports

This section provides examples of the transaction processing reports that ATTAINS generates.

Appendix C: ATTAINS Schema Diagram

This section provides a diagram of the ATTAINS schema.

3.0 ATTAINS Web Service Methods at CDX

The following sections discuss the web service methods available at CDX for the ATTAINS data flow. This reference can be used by the developers of a State/Regional/Tribal Node in implementing the interface between their node and CDX. For more information, see the Network Node Functional Specification document available on the Exchange Network web site (www.exchangenetwork.net).

3.1 Registration / Authorization

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

Request user/system accounts from the ATTAINS administrator(s) and they will coordinate with CDX to provide a NAAS account and grant it the appropriate method access for the appropriate organizational data in ATTAINS.

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 ATTAINS user account mapping and the ATTAINS user account identifier will be passed to the ATTAINS node via the NAAS security token to be used for authorization purposes. If the ATTAINS user account identifier matches with an ATTAINS user ID, then ATTAINS will return all available data. Otherwise, ATTAINS will return only publicly available data.

3.2 Authenticate

The authenticate method is used to obtain a security token from the Network Authentication Authorization Service (NAAS). This token will be passed in all subsequent method calls in the exchange between your node and CDX. Please note that security tokens will expire 20 minutes after the token is last used.

Parameters:

- `userId`: the User ID identifying your node.
- `credential`: the password you were issued along with the User ID.
- `authenticationMethod`: the method used to authenticate. Currently only "password" is allowed

Returns: a `securityToken` used to identify your session with CDX.

3.3 Submit

The Submit method is used to send your ATTAINS submission file to CDX.

Parameters:

- `securityToken`: A security token issued by the NAAS and returned from the Authenticate method.
- `transactionId`: unused.
- `dataflow`: The name of target dataflow: "ATTAINS"
- `documents`: An array of type `nodeDocument`. Each `nodeDocument` structure contains a single submission document/file. For ATTAINS, only a single instance of `nodeDocument` is submitted. The `nodeDocument`'s "documentName" property must have a file extension of ".zip" (e.g. "TestSubmission.zip"). Any documents with other extensions in the `documentName` property will not be processed.

- **notificationURI** (Version 2.0 nodes only): Used to optionally specify an email address that should receive a notification when processing is complete at CDX. Node v1.1 submissions 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 submissions.

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 of the submission. Submissions that fail at the CDX validation level will not trigger the email notification.

Returns: a **transactionId** which identifies your submission. This can be used in the **GetStatus** and **Download** methods (described below).

3.4 GetStatus

The **GetStatus** method is used to check on the current status of a submission file being processed or a solicitation for ATTAINS data. Generally, this is used to follow-up on a previous call to the **Submit** or **Solicit** method.

Parameters:

- **securityToken**: A security token issued by the NAAS and returned from the **Authenticate** method.
- **transactionId**: A Transaction ID returned by the **Submit** or **Solicit** method.

Returns: the status of the specified transaction: **Received**, **Pending**, **Completed**, or **Failed**.

3.5 Download

The **Download** method is used to download documents from the CDX node relating to a specific **transactionId**. Generally, this method is used after the **GetStatus** returns either **Completed** or **Failed**.

There are two purposes for the **Download** method:

1. Retrieve the Processing Report after a submission file has been processed.
 - a. This report describes the processing that occurred on your submission and any errors or warnings that arose.
2. Retrieve the results of a request for ATTAINS data using the **Solicit** method.

Parameters:

- **securityToken**: A security token issued by the NAAS and returned from the **Authenticate** method.
- **transactionId**: A transaction ID returned by the **Submit** or **Solicit** method.
- **dataflow**: The name of the dataflow: **"ATTAINS"**.
- **documents**: (optional).

If the **documents** parameter is left blank, then all documents relating to this transaction will be returned. This parameter is useful to avoid downloading things you don't need. For example: pass in the Processing Report name to download just the processing report, and avoid downloading your original submission file along with it.

The documents parameter is made up of the following three fields:

- name: the name of the document you wish to download.
- type: the file type (XML, ZIP, OTHER).
- content: not used.
- Returns: a set of documents.

The following documents are available for download:

For a *Submit* transaction:

1. The original submission document.
- 2a. The Schema Validation Report (regardless of whether CDX found a problem with the format of your XML submission file).
(name = "Validation Results", type = "XML")
and
- 2b. The Processing Report from the ATTAINS System.
(name = "ProcessingReport", type = "XML"; compressed as a .zip file named "ProcessingReport.ZIP")

For a *Solicit* transaction:

1. The results file.
(name = "Results.zip", type = "ZIP").

3.6 Query

The Query method is used to synchronously query the ATTAINS system and retrieve data from it. Queries have restrictions to keep the size of the output reasonably small. For requests that may return a larger set of data, please use the Solicit method. Section 8.0 describes the Query and Solicit Requests that are currently support by the ATTAINS system.

Parameters:

- securityToken: A security token issued by the NAAS and returned from the Authenticate method.
- request: The name of the query to be performed (see section 8.0 for more details).
- rowId: not used.
- maxRow: not used.
- parameters: An array of parameter values for the query to be performed. Parameters are specific to the request that is made (see section 8.0 for details).

Returns: a set of data in an XML instance document. The returned data set is dependent on the request made.

3.7 Solicit

The Solicit method is used to query the ATTAINS system and retrieve data from it. Unlike the Query method, which returns the results immediately, the Solicit method processes the request asynchronously and creates a zipped XML document that can be downloaded later. The system will email a notification when the processing is completed and the GetStatus method can also be used to ascertain whether the processing has completed. Due to its asynchronous nature, the solicit methods are also able to handle requests for a larger volume of data than the Query method. Section 8.0 describes the requests that are currently support by the ATTAINS system.

Parameters:

- **securityToken:** A security token issued by the NAAS and returned from the *Authenticate* method.
- **returnURL:** Not supported at CDX (leave blank).
- **request:** The name of the query to be performed (see section 8.0 for more details).
- **parameters:** An array of parameter values for the query to be performed. Parameters are specific to the request that is made (see section 9 for details).

Returns: a *transactionId* which identifies your request. This can be used with the *GetStatus* and *Download* methods to retrieve your results.

3.8 ATTAINS End Points

Development:

<https://devngn.epacdxnode.net/ngn-enws20/services/NetworkNode2Service>

Staging:

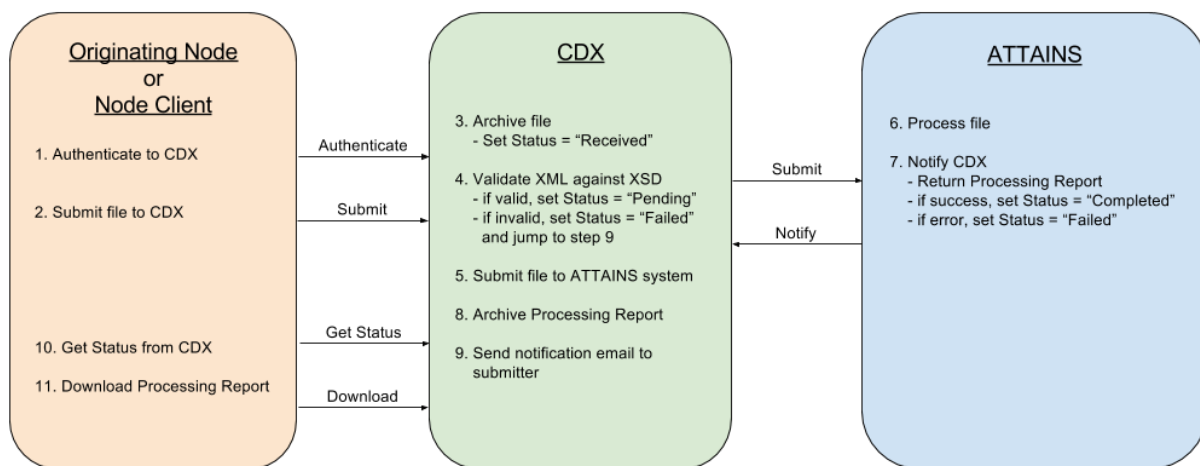
<https://testngn.epacdxnode.net/ngn-enws20/services/NetworkNode2Service>

Production:

<https://cdxnodengn.epa.gov/ngn-enws20/services/NetworkNode2Service>

4.0 ATTAINS Submission Flow and Feedback

Submission of Exchange Network Documents to the ATTAINS System via the CDX Node follows these processing steps:



1. The Originating Node/Node Client calls the *Authenticate* method to initiate a session with CDX.
 - a. If authentication is successful, a Security Token will be returned.
2. The *Submit* method is called to submit an ATTAINS file for processing.
 - a. A Transaction ID is returned, indicating that the file transfer was successful.

3. The submission file (.zip or .xml) is archived at CDX.
 - a. The status of the submission is set to “Received”.
4. The XML submission file is validated at CDX:
 - a. The file is compared with the Header Schema and the relevant ATTAINS Schema to confirm that it complies with the required structure.
 - b. If the XML file passes validation, the submission status is set to “Pending”.
 - c. Otherwise, a Processing Report is generated and the submission status is set to “Failed”. No further processing occurs (skip to step 9).
5. CDX submits the file to the ATTAINS System (by calling its *Submit* method).
6. The ATTAINS System processes the submission file (i.e. performs database inserts, updates, and deletes as indicated in the file) and loads any attached binary objects. All ATTAINS business and data validation rules, their descriptions, and their respective change histories are listed in the ATTAINS Business Rule Registry (see Appendix A, below).
7. The ATTAINS System notifies CDX to update the submission status and return a Processing Report.
 - a. If the file was processed without errors, then the status is set to “Completed”.
 - b. If there were errors, then the status is set to “Failed”.
8. The Processing Report is archived at CDX.
9. An email is sent to the Submitter User ID (an email address) notifying him/her of the final status of the submission (“Failed” or “Completed”). If the notificationURI parameter has been supplied, a duplicate notification will be emailed to that address as well.
10. The Originating Node/Node Client calls the *GetStatus* method to determine the status of the submission. If the Originating Node is manually controlled and the submitter is already aware of the status (from the email in step 9), then this step could be skipped.
11. Once the status is either “Completed” or “Failed”, the *Download* method is called to retrieve the Processing Report from CDX.
 - a. The Processing Report is an XML-based summary of the processing that occurred on the ATTAINS System (including processing time, insert/update/delete counts and any error or warning messages). See Appendix A for an example Processing Report.

5.0 ATTAINS Submission Data Processing

This section provides more detail regarding the data processing of ATTAINS submissions, once received from the CDX node by the ATTAINS node.

The first element value in each of the XML schemas for ATTAINS is the *OrganizationId*. This is a unique identifier for an organization and must be assigned by the EPA before that organization can submit data to the ATTAINS system. An organization cannot be created or deleted via a data submission.

Before processing any submission file, the ATTAINS system will first validate that the *OrganizationId* in the file is valid and that the User ID used to authenticate and submit the file to CDX has been authorized to modify data for that organization.

Once the user and organization have been validated, the ATTAINS system will proceed to process the submission file. Each submission file can contain only one type of payload, either “Update-Insert” or “Delete”. The following sections explain the processing that occurs for each type of payload

5.1 Update-Insert

This section describes the ATTAINS Schema for the “Update-Insert” operation. Please refer to the ATTAINS Update-Insert Schema Model (see Appendix C) and to the associated example XML files.

The Update-Insert schema defines one root component for the entire hierarchy: Organization. Each submission file must contain one (and only one) Organization, uniquely identified with an OrganizationIdentifier.

In addition to the Organization, three other primary components exist which must also be uniquely identified (within the organization): AssessmentUnits, ReportingCycle, Actions. A valid ATTAINS submission file only needs one of these primary components to be valid. In other words, you can submit a file of ReportingCycle only, AssessmentUnits only or Actions only. Likewise, it is also valid to submit any combination of these components in a single file (e.g. AssessmentUnits, ReportingCycle, Actions in one file).

Updates can be made to ReportingCycle, AssessmentUnits and Actions only at the top-level. Child elements of these primary elements can only be updated via an update to their respective parent. For example, to update an AssessmentUnit’s Locations, you must resubmit the parent AssessmentUnit with its corrected list of associated Locations because Locations are not uniquely identified.

The ATTAINS System will automatically determine if the primary component you have submitted is to be Inserted or Updated in the database. The following procedure is used to make this determination: the unique identifier for each AssessmentUnit or Action in the payload is compared with the identifiers for these respective items in the ATTAINS database. If the identifier is found, an “Update” is performed, replacing the record in the database with the copy from the payload. If it is not found, an “Insert” is performed, creating a new record in the database. Accurately updating the ATTAINS System is only possible if you have uniquely identified each of the primary components in your submission file.

The ATTAINS System performs a commit (i.e. save) each time it successfully loads all the data belonging to one of these primary components. If there are any errors encountered while loading a primary component (or its children), all changes for that primary component will be rolled back (i.e. not saved). For example: if one Location has an error in it, then its Assessment Unit (which is the primary component) and all other Locations on that Assessment Unit will fail and not be saved. This error, however, only affects this one Assessment Unit. All other Assessment Units in the file that are loaded error-free will still be saved. This rule relates to each of the primary components. Each AssessmentUnit, ReportingCycle or Action that is free of errors will be loaded into the ATTAINS System. Those containing errors will not be loaded.

If any errors occur while loading the submission file into the ATTAINS System, the submission status will be set to “Failed” at CDX. Likewise, the Processing Report will include an itemized list of any primary components that failed to load.

If a submission fails to load successfully, there are two approaches to correcting it.

- 1) Resubmit the entire dataset again (after correcting the data that originally failed). In this case, any data that originally loaded successfully will now be treated as an “Update”. Any data that originally failed will be treated as an “Insert”.
- 2) Submit a new, corrected, dataset that only includes the components that originally failed. This new file will be more efficient to process because it is likely much smaller than the complete data set.

5.1.1 Specific Functionality

Most the fields in the DET are straight forward and need little explanation. Some fields have business rules that are important to note during submission. Those business rules can be found at the end of this document as well as the DET. There is another subset of fields that have substantial backend logic. These fields are described in detail in this section.

5.1.1.1 Reporting Cycle

ReportingCycles encapsulate Assessments of AssessmentUnits, among other things. ReportingCycles do not have a unique identifier; nor do the encapsulated Assessments of AssessmentUnits. ReportingCycles are identified by the cycle year (ReportingCycleText), and their Assessments are identified by the AssessmentUnitID within a specific cycle year. There can be only one Assessment for each AssessmentUnit per 2-year reporting cycle. This section describes how submitting reporting cycles works in the EN:

Process

If a cycle is not in the database and is submitted for the first time via the EN

- A new cycle will be created.
- All assessments in the previous cycle will be carried over to the new cycle.
- Any assessments in the submission file will be added/updated.
- For any assessment that was carried over that does not exist in the submission file, we will produce a warning in the processing report. The warning will let the user know that there are some other assessments in the cycle that were not in their submission file.

If a cycle is already in the database and a new file is submitted for the same cycle:

- We will not create a new cycle.
- All assessments for that status are deleted, including any changes made in the UI.
- Assessments from the previous snapshot are brought forward.
- Any assessments in the submission file will be added/updated.
- We will also show the same warning as described in the previous section.

Validation

Cycle promotion, which is performed by changing the ReportStatusCode (i.e. from Organization Draft to Organization Public Comment), will not be supported via EN. Users will need to promote cycles via the user interface. However, users can create new cycles or update in progress versions of the same status for existing cycles. New cycles can only be created if these rules are met:

- Cycle year in file is greater than current cycle year
- Current cycle year is in one of these statuses: EPA Document Decisions, EPA Interim Action, EPA Final Action, or Historical

Updates to existing cycles only reset the “in progress” version of the current snapshot. These submissions will not affect any locked snapshots created from previous status promotions. Existing cycle in progress snapshots can only be updated if these rules are met:

- Cycle year in file is the same as the current cycle year
- Current cycle year is one of these statuses: Organization Draft, Organization Public Comment, Organization Final – Internal Review, Organization Final – Submittal (Only if there is an in progress copy that is not locked).

Besides assessments there are a few other Reporting Cycle level fields that are supported in the EN. Users can pass Combined Cycles through the EN for cycle inserts only. This means that the Combined Cycles block needs to be present when a new cycle is created for it to save. This is similar to the UI functionality. Users can also pass Cycle Documents via the EN. We recommend that these Cycle Documents are versioned with the year and the cycle status. If a document already exists for an organization, it is ignored in the submission. Cycle Documents are brought forward from the previous snapshot and documents from the submission are then added to those existing documents.

To remove an assessment from a cycle or prevent an assessment from being carried over the associated assessment unit should be retired. We will still support deleting a ReportingCycle via EN which will delete

all assessments associated with the cycle from the previous submission (but not the assessments that were carried over).

It is important to note that the ReportingCycle will save even if one or more assessments fail. This is a result of order of operations necessary to support submissions that encapsulate more than one primary component. If a new ReportingCycle is submitted, but one or more assessments fail the new cycle year will be created and the Assessments from the previous cycle will be brought forward. The failed Assessments will not save and therefore not be a part of the new cycle until fixed and resubmitted. This is also true if resubmitting a cycle update.

5.1.1.2 Snapshotting

A reporting cycle is promoted when the ReportStatusCode is changed. This is currently only possible in the Cloud Interface. When a reporting cycle is promoted a snapshot of the data is taken and a working version is created. Any changes to assessment units, or assessments done after the promotion will be a part of the working copy but will not be part of the previous snapshot. Actions are not being snapshotted as of May 2017.

Example: A user promotes 2016 Reporting Cycle from draft to public review. After the promotion, the user changes information on an assessment unit and assessment that is part of that cycle. The user then decides the information is correct and promotes to state final. The draft snapshot will show the older version of the changed assessment unit and assessment. The public comment snapshot will show the updated data. A working copy is then automatically started to handle any changes made before the next cycle is created.

5.1.1.3 Delisting

Delisting occurs when a water moves from category 5 in a previous cycle to some lower category in the new cycle. The system is smart enough to compare cycles and understand when a water has been delisted. However, we have included a delistings section in the schema so the user can specify a reason and a comment. The system can also track Water Quality Restorations, although it can't identify the specific reason for the restoration. The Delisting block is used to track both actual 'Delistings', and Water Quality Restorations. For Restorations, we recommend that DelistingReasonCode be provided, otherwise a default code will be used. This is how the system will handle delistings/restorations:

- When a new assessment is loaded via the exchange network the system will compare against the previous cycle and determine if it has been delisted.
- If it has been delisted the system will then attempt to assign a DelistingReasonCode. The system can calculate the following DelistingReasonCodes:
 - TMDL Approved or established by EPA (4a)
 - Other pollution control requirements (4b)
 - Not caused by a pollutant (4c)
 - If the EPAIRCategory has gone from 5 or 4 to another category not listed above then the system will assign the value: "Applicable WQS attained; reason for recovery unspecified" value for the DelistingReasonCode
- Users can then override the assigned DelistingReasonCode by passing the DelistedWaters block in the same file or subsequent files. The user submitted DelistingReasonCode will always overwrite any system calculated reason. If the user doesn't pass a DelistingReasonCode than the system generated reason will remain.
- If the system determines an assessment to be involved in a delisting, but no DelistedWaters block is passed a warning will be returned in the processing report. It will still be saved as a delisting, warnings are for information purposes only and won't stop the submission from saving.

- If a DelistedWaters block has been passed for a water that has not been delisted the system will generate a warning that lets the user know that water has not been delisted. The cycle will continue to save, but that delistings block won't be part of the cycle as it references a water that isn't a true delisting.

5.1.1.4 Assessment Unit Modification

5.1.1.4.1 Split

Users can split an Assessment Unit into two or more Assessment Units that have historic ties to the parent AU. Splitting is a specific ModificationTypeCode in the Modifications block, which is a child of AssessmentUnit. Here is an example of a successful split:

AU1 is a current Assessment Unit that needs to be split into AU1_a and AU1_b. The user will include an Assessment Unit for AU1_a and AU1_b in the submission file. Each of those will have a modification block with ModificationTypeCode = 'split' and Previous AssessmentUnitIdentifier = 'AU1'. AU1 should also be in the submission file or already exist in the database.

Often states will want to retire AU1 (the referenced Assessment Unit), but it is not required. The decedents from a split can be either new or already stored Assessment Units.

Split Business Rules

- If PreviousAssessmentUnit isn't in the system, AU will not save. An error will be reported in the processing report.
- For splits, only one PreviousAssessmentUnit is allowed. If more are passed the AU will not save and an error will be reported in the Processing Report.
- The PreviousAssessmentUnitIdentifier referenced in modifications must be in the same submission or exist currently in the database. If not AU will not save. An error will be reported in the processing report.
- A split can only be performed through the EN if the current cycle is in draft status. We will look to enhance this functionality in future releases.

5.1.1.4.2 Rename

Users can change an Assessment Unit Identifier by performing a rename. The Assessment Unit Identifier is the field that defines a unique Assessment Unit. Under the hood an Assessment Unit Rename is just creating a brand-new Assessment Unit and associating it to the previous one. Here is an example of a successful rename:

AU1 is a current Assessment Unit that needs to be renamed to AU2. The user will include an Assessment Unit for AU2 in the submission file. AU2 will have a modification block with ModificationTypeCode = 'rename' and Previous AssessmentUnitIdentifier = 'AU1'. AU1 should also be in the submission file, but not necessary.

Rename Business Rules

- Only one PreviousAssessmentUnit can be passed, anything else will throw an error.
- Rename will automatically retire the PreviousAssessmentUnit
- The PreviousAssessmentUnit needs to be an active AU already in the database or an error will be thrown.
- Rename can only be performed for a new Assessment Unit (Insert not Update). Users can't decide to rename an Assessment Unit into an Assessment Unit that already exists. All rename modification blocks will be ignored, and a warning thrown, for Assessment Units that are already in the database and are just "updating".

5.1.1.5 Assessment Unit Retire

Assessment Unit Retire occurs when the Status Indicator (child of Assessment Unit) is changed from “A” to “R”. This essentially locks the Assessment Unit from being edited. It also prevents future Assessments from referencing back to that AU. When a new cycle is created, an Assessment will not be brought forward for a retired Assessment Unit. Currently the system will not allow the status to be changed back to active after an Assessment Unit has been retired. The following use cases show how the system will handle various situations:

- All active Assessment Units can be retired except AU’s that have an Assessment with EPAIRCategory 5 in the latest EPA Final cycle.
- If an AU is changed and retired in the same file the system will save the changes and then retire. The new changes will be part of that retired AU.
- If an Assessment Unit with either status (“A” or “R”) is passed that references an Assessment Unit already in the database with status “R”, the system will not save it. A warning will be provided in the processing report for the user’s information.
- If an Assessment Unit that is being retired is in the same file as a new reporting cycle the AU will change status to retire first and then the cycle will be created. There won’t be an assessment in the new cycle for that retired AU.
- If a cycle references a retired AU it will fail and not be saved. An error will be shown in the processing report for the user’s information.

5.2 Delete

This section describes the ATTAINS Schema for the “Delete” operation. Please refer to the ATTAINS Delete example submission file. Note that ATTAINS implements certain rules of data retention based on the status of the records in question. See Appendix A for more information on data retention rules.

The XML Schema for the “Delete” operation is very simple. It includes the identifiers for the primary components in the system. Because Assessments are passed as a group, they cannot be deleted individually. The Delete operation allows the user to delete all Assessments for a particular cycle by specifying the cycle.

OrganizationIdentifier is required (and must be provided only once). The OrganizationIdentifier is required for context information only (organizations cannot be deleted).

In addition to the OrganizationIdentifier, at least one of the other identifiers (e.g. AssessmentUnitIdentifier, ActionIdentifier, ReportingCycle) must be provided and can be repeated as many times as necessary to identify all of the records to be deleted from the ATTAINS Database.

The following procedure defines the delete process:

- The unique identifier for each AssessmentUnit, Reporting Cycle and/or Action in the payload is compared with the identifiers for these respective items in the ATTAINS Database. If the identifier is not found, an error is added to the Processing Report, and the identifier is ignored. If the identifier is found, then the matching record in the database is deleted. In the case of Assessments, all Assessments for the given reporting cycle will be deleted.

When one of these primary components is deleted from the system, all child data is deleted as well.

6.0 ATTAINS Submission Exchange Network Document

6.1 Overview

The ATTAINS Exchange Network Flow supports the Exchange Network Document Header Specification Version: 2.0 (http://www.exchangenetwork.net/dev_schema/headerspecification_v2.0.doc); which provides the necessary definitions for the document sections and elements. ATTAINS supports XML version 1.0.

This section will provide additional ATTAINS-specific clarifications and examples regarding the Submission Exchange Network Document.

The ATTAINS Submission Exchange Network Document must contain the following sections:

- **Header** - The Exchange Network Header provides information describing the contents of the payload and facilitates automation of the data exchange process by making data more readily identified during transport and managed at its processing destination. The Header Section of the document 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 inserts and updates or deletions.
- **Payload** - The Payload section of the submission file contains the ATTAINS data, which must conform to the standard ATTAINS schema for data submissions. Although some data flows allow for multiple payloads in one document, the ATTAINS data flow allows only a single payload per submission.

The following table describes the document-level attributes and how they are utilized for an ATTAINS submission.

Document

Attribute	Description	Example Value / Notes	Required
Xmlns	Namespace	"http://www.exchangenetwork.net/schema/header/2"	Yes
Id	A unique identifier for the document. This is an attribute of ExchangeNetworkDocument and provides a unique Id for each document in the payload.	"MyDocumentId0001"	Yes

6.2 Header (Section)

Any network exchange for ATTAINS must follow the Header 2.0 structure to meet EPA CDX and ATTAINS processing requirements and prior Exchange Network agreements. Refer to the Header Specification 2.0 document alluded to in the previous section. Pay particular attention to the required elements. The specification also refers to several elements that are not required by the specification, nor by ATTAINS. However, it may be helpful to provide them for various reasons explained in the specification.

Beyond the Header 2.0 specification, the following additional ATTAINS-specific clarifications and examples are applicable. Some of these elements are not required per the Header 2.0 specification but are considered required when using ATTAINS.

Element Name	Required	Additional ATTAINS-specific Clarifications and Examples
OrganizationName: The organization to which the author belongs. It may be a state name, an organization name or a company name. For submissions to the CDX node, this should be the name of the organization.	Yes	“State X Department of Environmental Quality” This identifies the <u>author's organization</u> , which may be different from the organization who owns the data. The data owner is identified in the ATTAINS payload.
DocumentTitle: Title of the document.	Yes	Follow this example for ATTAINS: "ATTAINS 2015/2016 Reporting Cycle Submission"
DataFlowName: The name of the data flow associated with the payload. It could be the name of the data source for Query results.	Yes	“ATTAINS” The Header 2.0 specification does not require this element but ATTAINS does; and it must be spelled exactly as shown here.
SenderContact: The sender’s additional contact information. It could contain sender’s electronic address and/or telephone numbers where the author can be reached.	Yes	“Joe Smith 123 Main St. Portland, OR 97226 503 123 4567 Joe@deq.statex.gov” The Header 2.0 specification does not require this element but ATTAINS does. For ATTAINS use Name, mailing address, city, state, zip, telephone number, and email address of person who may be contacted with questions concerning the submission.
SenderAddress: A well-formed URI where result/report can be sent. Currently the Network will make use of the Notification mechanism at the Document Level as described in the Protocol and Specification. Note that this could contain multiple addresses, including that of the submitter and/or other technical people related to contents of the payload.	Yes	“mailto:jsmith@example.com” The Header 2.0 specification does not require this element but ATTAINS does.

The following provides an ATTAINS-specific example use of the header showing only required elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Example ATTAINS Header-->
<Document xmlns="http://www.exchangenetwork.net/schema/header/2" id="AS2016-01">
```

```
<Header>
  <AuthorName>Joe Smith</AuthorName>
  <OrganizationName>Utah Department of Environmental
Quality</OrganizationName>
  <DocumentTitle>ATTAINS 2015/2016 Reporting Cycle
Submission</DocumentTitle>
  <CreationDateTime>2016-03-05T09:30:47-05:00</CreationDateTime>
  <DataFlowName>ATTAINS</DataFlowName>
  <SenderContact>Joe Smith, 123 Main St., Portland, OR 97226, 503 123 4567,
Joe@deq.statex.gov</SenderContact>
  <SenderAddress>mailto:jsmith@example.com</SenderAddress>
</Header>
<Payload operation="Update-Insert">
...
</Payload>
</ExchangeNetworkDocument>
```

6.3 Payload (Section)

The Payload Section of the Exchange Network Document contains an attribute named “Operation”. This is used to denote the type of processing for a submission. There are two acceptable values: “Update-Insert” or “Delete”. Use of these operators triggers the processing outlined in the *Configuration of the Network Exchange* section of this document. For example, a payload operation of “Update-Insert” informs the back-end application that the payload contains new or existing data. “Delete” denotes that the payload contains identifiers for data to be deleted.

If the “operation” attribute is “Update-Insert” then the Payload must conform to the Update-Insert Schema (“ATTAINS_v1.0.xsd”). If the “operation” attribute is “Delete” then the Payload must conform to the Delete Schema (“ATTAINS_Delete_v1.0.xsd”). See the associated example ATTAINS submission files to gain a better understanding of how to implement Update-Insert and Delete submit requests.

Payload

Attribute	Description	Example Value	Required
operation	This describes the operation to be performed on the payload. Multiple values are not allowed	Must be either “Update-Insert” or “Delete”	Yes
ATTAINS (Payload Sub-Section)			
XML Namespace (“xmlns”)	ATTAINS approved namespace for submission files	During testing: “http://www.exchangenetwork.net/schema/IR/0” After release of version 1: “http://www.exchangenetwork.net/schema/IR/1”	Yes

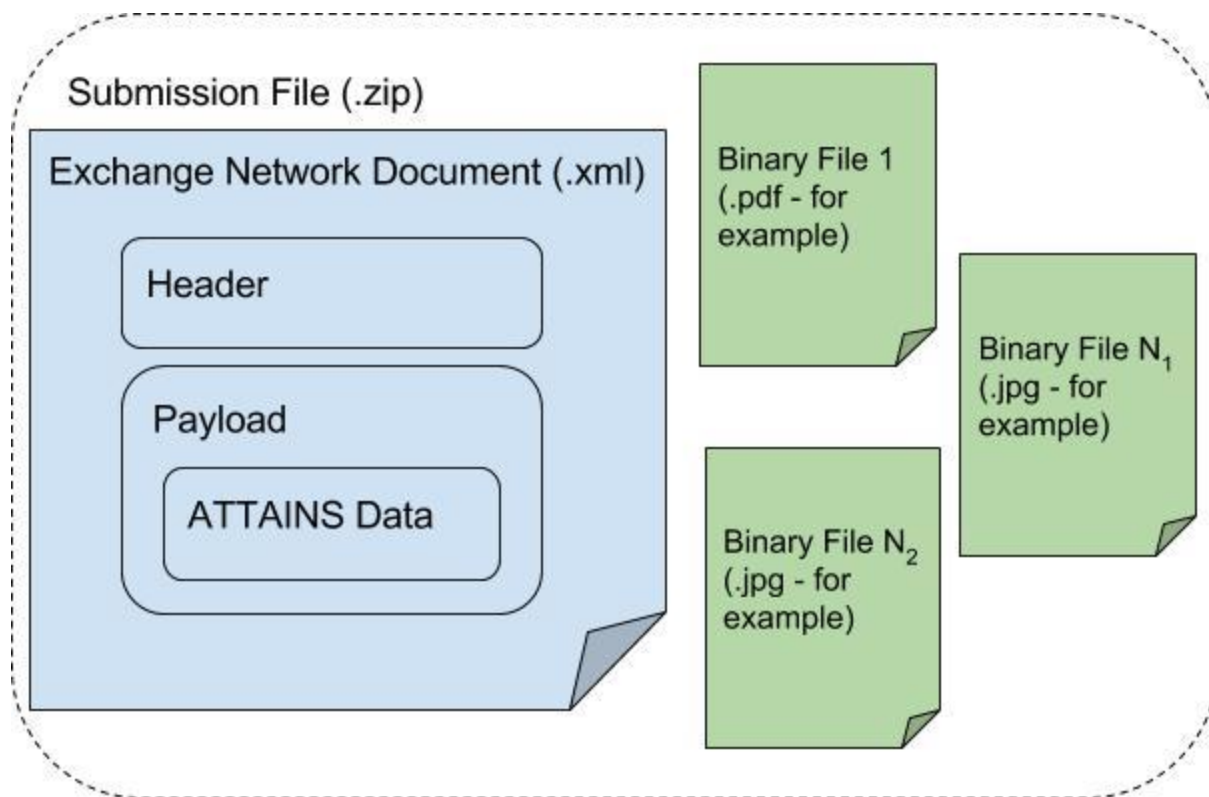
Update-Insert operation-type payloads can reference, by unique name, binary objects, such as images or documents, to be included as part of the submission. The binary objects, themselves, are provided as individual, external documents, and should be included in the .zip file with the submission. See section 6.4 for more information on including external, binary objects.

6.4 File Structure for a Submission Payload

In order to group all related documents, (the ATTAINS Exchange Network Document and any referenced binary objects), each ATTAINS submission should be provided to CDX in a single compressed file in ‘.zip’ format. This also improves transmission efficiency due to the significantly smaller size of compressed files. Even if a submission does not contain any external binary objects, it must be provided in ‘.zip’ format.

Additionally, the Exchange Network Document within the submission must be the only file with an ‘.xml’ extension. This allows CDX to differentiate between the Exchange Network Document and any other binary object files included in the submission.

The following diagram describes the structure of a Submission Zip File for ATTAINS:



7.0 ATTAINS Query & Solicit Flow and Feedback

The 8.0 ATTAINS Query & Solicit Exchange Network services allow a State/Regional/Tribal node to request data from the ATTAINS system in XML payloads. Except for the GetTransactionHistory method, all returned XML payloads are in accordance with the ATTAINS schema.

There are two methods for retrieving this data: the Query method, which returns the results immediately, and the Solicit method which creates a data file (offline) which can be downloaded later. The Query method is ideal for smaller requests and is limited in terms of the number of results returned. The Solicit method is intended to accommodate requests for larger result sets asynchronously.

There is a numeric configuration setting called “MaxQueryResults” in the ATTAINS configuration file. If the query will retrieve more than the value of MaxQueryResults setting, the system will return an error message indicating that the query will retrieve too many results and, therefore, the corresponding Solicit method should be used. The current value of the setting can be determined by either testing the query methods or by contacting the EPA ATTAINS support personnel.

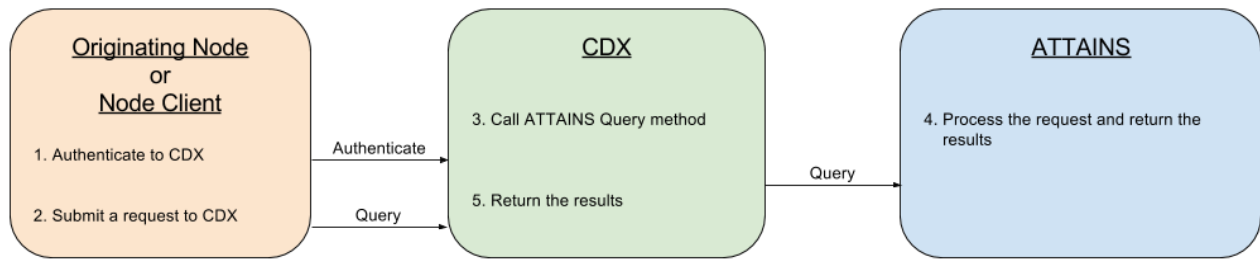
If the requesting NAAS user has a valid ATTAINS account association, confidential data will be provided by the Query and Solicit services; in all other cases the ATTAINS will provide only public data to the requestor.¹

The ATTAINS Query & Solicit Exchange Network services will also be duplicated as non-authenticated, public-data-only, REST-based web services as part of the ATTAINS Public Web Site. NOTE: the REST-based Query services are being developed and are not yet available. See section 7.3 for more detail.

¹ Geospatial data will be available via Web Map Service (WMS) and Web Feature Service as part of the ATTAINS Cloud Interface. See the Cloud Interface documentation for details.

7.1 Query Process

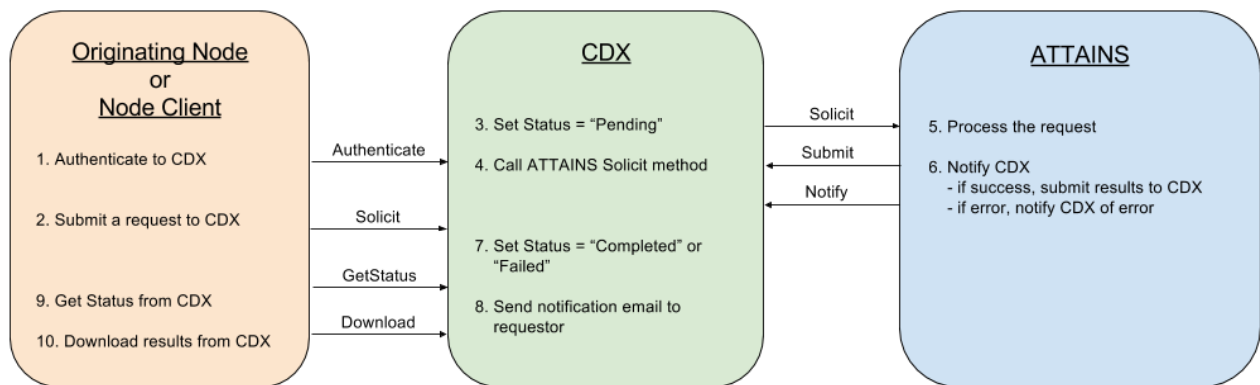
The Query process follows these processing steps:



1. The Originating Node/Node Client calls the *Authenticate* method to initiate a session with CDX.
 - a. If authentication is successful, a Security Token will be returned.
 2. The *Query* method is called to request data from CDX.
 3. CDX forwards the request on to the ATTAINS System (via the *Query* method).*
 4. The ATTAINS System processes the request and returns the results or an error to CDX.*
 5. CDX returns the results as XML (or a SOAP exception in the case of an error).*
- * Steps 3, 4, and 5 are all transparent to the Originating Node. The *Query* method call (in step 2) returns the results (mentioned in step 5).

7.2 Solicit Process

The Solicit process follows these processing steps:



1. The Originating Node/Node Client calls the *Authenticate* method to initiate a session with CDX.
 - a. If authentication is successful, a Security Token will be returned.
2. The *Solicit* method is called to request data from CDX.
 - a. A Transaction ID is returned.
3. The Transaction Status is set to "Pending".
4. CDX calls the *Solicit* method on the ATTAINS System.
5. The ATTAINS System processes the request and creates a results file corresponding to the criteria it was passed.

6. If the request is successful, then the ATTAINS System calls the *Submit* method to return the results file to CDX.
If the request fails, then the ATTAINS System calls the *Notify* method to return the error message.
7. CDX sets the Transaction Status to “Completed” if the request was successful or “Failed” if not successful.
8. An email is sent to the requestor notifying him/her of the final status of the solicit request (“Failed” or “Completed”).
9. The Originating Node/Node Client calls the *GetStatus* method to determine the status of the transaction. If the Originating Node is manually controlled and the requestor is already aware of the status (from the email in step 8), then this step could be skipped.
10. Once the status is either “Completed” or “Failed”, the *Download* method is called to retrieve the results file (or error message).
- 11.

7.3 Data Publishing Services using REST

The ATTAINS Node has exposed Representational State Transfer (REST) publishing capabilities for its user community and the public. This capability allows retrieval of public-only data from the ATTAINS system in JSON format. The schema closely follows the ATTAINS DET for the Exchange Network.

7.3.1 Access / Authorization

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

All ATTAINS REST services have the following base URLs.

Environment	Base REST URL
Pre-production	http://attainsstage.rtpnc.epa.gov/attains-public/api
Production	http://attains.epa.gov/attains-public/api

7.3.2 URL Format for REST Services

The URL format to access any ATTAINS REST service is:

BASE_REST_URL/{service_name}?{parameter_name}={parameter_value}&{parameter_name}={parameter_value}

where &{parameter_name}={parameter_value} repeats for each parameter provided.

7.3.3 Standard Services and Parameters

Each of these services corresponds to a service in section 8. Please refer to section 8 for a complete list of possible parameters with associated descriptions. Please note that the REST parameters are camel case.

Services	Description
domains	Used to look up ATTAINS domain values
assessmentUnits	Returns Assessment Units and corresponding children
assessments	Returns Assessments and corresponding children for a designated Reporting Cycle
actions	Returns Actions and corresponding children

7.3.4 Additional Services and Parameters

These services are unique to our public services and don't have corresponding EN services. They generally serve a very specific purpose and have been built primarily to interact with the new How's my Waterway application. However, they are public and can be utilized by any application.

Service Name - plans

Sample URL - <http://attains.epa.gov/attains-public/api/plans?huc=01234567>

Returns - complete actions block in JSON format; similar to the actions service.

Description – This service accepts HUC values and returns associated Actions. Specifically, the system looks for any Assessment Unit related to that HUC via a catchment. The service then passes those Assessment Units to our actions service and returns the unique matching Actions.

Parameter	Description
huc	Accepts any huc value of 8 characters or greater. Multiple comma separated values accepted.

7.3.5 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>
```

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

8.0 ATTAINS Query/Solicit Requests

The ATTAINS Exchange Network Flow will support the following requests available via both the *Query* and *Solicit* web service methods described in section 8. For a given request, at least one of the query parameters with an asterisk ("*") must be provided in the request. Multiple search criteria may be combined. Records must match all specified (or default) search criteria to be considered a match. Note that multi-valued search parameters should be delimited using commas. Some domain values may contain commas (i.e. 2,3,7,8-TETRACHLORODIBENZOFURAN) these will need to be escaped using quotations (CSV rules) so the domain value doesn't get split on each comma (i.e. "2,3,7,8-TETRACHLORODIBENZOFURAN").

GetAssessmentUnits	
For retrieving basic information about the Assessment Units	
Query Parameter (any combination)	Description/Notes
assessmentUnitIdentifier*	Search criterion – Filters the list of assessment units to one or more specific assessment units. Multiple values can be provided.
stateCode*	Search criterion – Filters the list of assessment units to only those having a state code matches one in the provided list of states. Multiple values can be provided.
epaRegion*	Searching by EPARegion will not be available in ATTAINS version 1. Search criterion – Filters the list of assessment units to only those having an EPA region that matches one in the provided list of regions. Multiple values can be provided.
huc*	Search criterion – Filters the list of assessment units to only those which have a location type of HUC and the location value matches one in the provided list of HUCs. Multiple values can be provided.
county*	Search criterion – Filters the list of assessment units to only those which have a location type of county and the location value matches one in the provided list of counties. Multiple values can be provided.
assessmentUnitName	Search criterion – Filters the list of assessment units to only those having an assessment unit name matching the provided value.
reachCode* ²	Searching by Reach Code will not be available in ATTAINS version 1. Search criterion – Filters the list of assessment units to only those associated with one or more of the provided reach codes. Multiple codes may be provided.
organizationId*	Search criterion – Filters the list of assessment units to only those having an organization ID that matches one in the provided list of IDs. Multiple values can be provided.
lastChangeLaterThanDate	Search criterion – Filters the list of assessment units to only those last changed after the provided date. Can be used in conjunction with LastChangeEarlierThanDate to retrieve assessment units changed within a date range. If LastChangeEarlierThanDate is also provided, then LastChangeLaterThanDate must be earlier than LastChangeEarlierThanDate. Date format must be like “2005-10-16T14:00:00-06:00” or “2005-10-16”.
lastChangeEarlierThanDate	Search criterion – Filters the list of assessment units to only those last changed before the provided date. Can be used in conjunction with LastChangeLaterThanDate to retrieve assessment units changed within a date range. If LastChangeLaterThanDate is also provided, then LastChangeEarlierThanDate must be later than LastChangeLaterThanDate. Date format must like “2005-10-16T14:00:00-06:00” or “2005-10-16”.
statusIndicator	(A=“Active”, R=“Retired”) Search Criterion – Filters the list of assessment units to only those currently in the specified status.

² Reach codes will not be in the ATTAINS database. It will be part of the indexing (RAD).

	*Not relevant for REST services
includeDocuments	(Y/N) – Option - Causes the method to return associated documents in the zip file. For Solicit only. If you leave this blank, the default is “N”. *Not relevant for REST services
returnCountOnly	(Y/N) – Option - Causes the method to return only the count of assessment units that match the query search criteria. If you leave this blank, the default is “N”.

GetAssessments	
Retrieves a single reporting cycle’s assessment information. (Returns current status only)	
Query Parameter	Description / Notes
assessmentUnitIdentifier*	Search criterion – Specify which specific assessment unit assessment within the given reporting cycle to retrieve. Multiple values can be provided.
state*	Search criterion - Filter the list of reporting cycle assessments to those only from the specified state.
reportingCycle	Search criterion - Filter the list of assessments to only those of a specific reporting cycle in terms of the four-digit year the reporting cycle ended (e.g. for the 6/1/2016 – 5/31/2018 reporting cycle the four-digit reporting cycle is “2018”). Will also return if the value passed matches a combined cycle year. If left blank, the default is the current cycle relative to the current date.
listedAs303d	Searching by listedAs303d will not be available in ATTAINS version 1. (Y/N) – Search criterion - Filter the list of reporting cycle assessments to only those who’s assessment units are on the 303d list.
use	Search criteria - Filter the list of reporting cycle assessments to only those associated with the specified uses. Multiple uses may be listed.
useSupport	Search criterion - Filter the list of reporting cycle assessments to only those fully supporting the specified uses or that are threatened. Multiple UseSupports may be listed. The allowed values for this parameter are X = Not Assessed, I = Insufficient Information, F = Fully Supporting, N = Not Supporting, and T = Threatened. Values X, I, F, and N filter based solely on the UseAttainmentCode field of an Assessment. However, the last value (T) filters on a combination of the UseAttainmentCode and the ThreatenedIndicator where the UseAttainmentCode = 'F' and ThreatenedIndicator = 'Y'.
parameter	Search criterion - Filter the list of reporting cycle assessments to only those associated with one or more of the specified parameters. Multiple parameters may be listed.
parameterStatusName	Search criterion - Filter the list of assessments to only those assessments where the assessment has one or more associated parameters that are in the provided status. Valid values for this criterion are: “Monitored”, “Cause”, “Observed Effect.” Multiple ParameterStatusNames may be listed.

organizationId*	Search criterion - Filter the list of reporting cycle assessments to only those “belonging to” the specified organization.
probableSource	Search criterion - Filter the list of reporting cycle assessments to only those “having” the specified ProbableSource(s). Multiple values can be provided.
agencyCode	(E="EPA", S="State", T="Tribal") - Search criterion - Filter the list of reporting cycle assessments by the type of agency responsible for the assessment.
delistingStatus	Searching by delistingStatus will not be available in ATTAINS version 1. (Y/N) – Search criterion – Filter the list of reporting cycle assessments by whether the assessment’s assessment unit ID is or is not in the delisted waters list for the same reporting cycle.
reportStatusCode	Searching by reportStatusCode will not be available in ATTAINS version 1. Search criterion – Filter the list of assessments by the status of the specified reporting cycle (DRAFT, PUBLIC COMMENT, FINAL ACTION STATE, FINAL ACTION EPA). In other words, get the latest snapshot of the assessments for the given status code. If you don’t provide this then assessments for the available highest level status of the reporting cycle are retrieved.
epalrCategory	Searching by IRCategory will not be available in ATTAINS version 1. Search criterion – Filter the list of assessments to only those having one of the specified IR Categories. Multiple values can be provided.
stateIrcategoryCode	Search criterion – Filter the list of reporting cycle assessments to include only those having one of the provided codes (ON USES OR ON CAUSES). *Not relevant for REST service
multiCategorySearch	(Y/N) – Specifies whether to search at multiple levels. If this parameter is set to “Y” then the query applies the StateIrcategoryCode at the Assessment, UseAttainment, and Parameter levels; if the parameter is set to “N” it looks only at the Assessment level.
reachCode ³	Searching by Reach Code will not be available in ATTAINS version 1. Search criterion – Filters the list of assessments to only those associated with one or more assessment units associated one or more of the provided reach codes. Multiple codes may be provided.
lastChangeLaterThanDate	Search criterion – Filter the list of reporting cycle assessments to only those last changed after the provided date. Can be used in conjunction with LastChangeEarlierThanDate to retrieve assessments changed within a date range. If LastChangeEarlierThanDate is also provided, then LastChangeLaterThanDate must be earlier than LastChangeEarlierThanDate. Date format must be like “2005-10-16T14:00:00-06:00” or “2005-10-16”.
lastChangeEarlierThanDate	Search criterion – Filter the list of reporting cycle assessments to only those last changed before the provided date. Can be used in

³ Reach codes will not be in the ATTAINS database. It will be part of the indexing (RAD).

	conjunction with LastChangeLaterThanDate to retrieve assessments changed within a date range. If LastChangeLaterThanDate is also provided, then LastChangeEarlierThanDate must be later than LastChangeLaterThanDate. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
includeDocuments	(Y/N) – Option - Causes the method to return associated documents in the zip file. For Solicit only. If you leave this blank, the default is "N". *Not relevant for REST services
returnCountOnly	(Y/N) – Option - Causes the method to return only the count of assessments that match the query search criteria. If you leave this blank, the default is "N".

GetActions	
For retrieving actions (TMDLs or other actions)	
Query Parameter	Description/Notes
actionIdentifier*	Search criterion – Specifies exactly which action to retrieve. Multiple values may be provided.
assessmentUnitIdentifier*	Search criterion – Filters the list of actions to only those associated with any (i.e. at least one) of the provided assessment unit IDs, plus any statewide actions (Statewide Actions will not be available in ATTAINS version 1). Multiple values may be provided.
stateCode*	Search criterion - Filters the list of actions to those only "belonging to" the specified state.
organizationIdentifier*	Search criterion - Filters the list of actions to only those "belonging to" one of the specified organizations. Multiple values may be specified.
parameterName	Search criterion – Filters the list of actions to only those associated with any (i.e. at least one) of the provided parameters. Multiple values may be provided.
pollutantName	Search criterion – Filters the list of actions to only those associated with any (i.e. at least one) of the provided pollutants. Multiple values may be provided.
actionTypeCode	Search criterion – Filters the list of actions to only those of the specified ActionTypeCodes. Multiple values may be specified.
agencyCode	(S=State, E=EPA, T=Tribal) – Search criterion – Filters the list of actions to only those of the specified agency code. Multiple values may be specified.
npdesIdentifier	Search criterion - Filters the list of actions to only those whose NPDESIdentifier matches the value(s) passed. Multiple values may be provided.
pollutantSourceTypeCode	Search criterion - Filters the list of actions to only those whose PollutantSourceTypeCode matches the value(s) passed. Multiple values may be provided.
actionStatusCode	Search criterion – Filters the list of actions to only those currently having one of the provided ActionStatusCode. Multiple values may be specified.

	*Not relevant for REST, can only get EPA Final Action
reachCode ⁴	Searching by Reach Code will not be available in ATTAINS version 1. Search criterion – Filters the list of assessments to only those associated with one or more assessment units associated with one or more of the provided reach codes. Multiple codes may be provided.
completionDateLaterThan	Search criterion – Filters the list of actions to only those having a CompletionDate later than the value passed. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
completionDateEarlierThan	Search criterion – Filters the list of actions to only those having a CompletionDate earlier than the value passed. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
tmdlDateLaterThan	Search criterion – Filters the list of actions to only those having a TMDLDate later than the value passed. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
tmdlDateEarlierThan	Search criterion – Filters the list of actions to only those having a TMDLDate earlier than the value passed. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
lastChangeLaterThanDate	Search criterion – Filters the list of actions to only those last changed after the provided date. Can be used in conjunction with LastChangeEarlierThanDate to retrieve actions changed within a date range. If LastChangeEarlierThanDate is also provided, then LastChangeLaterThanDate must be earlier than LastChangeEarlierThanDate. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
lastChangeEarlierThanDate	Search criterion – Filters the list of actions to only those last changed before the provided date. Can be used in conjunction with LastChangeLaterThanDate to retrieve actions changed within a date range. If LastChangeLaterThanDate is also provided, then LastChangeEarlierThanDate must be later than LastChangeLaterThanDate. Date format must be like "2005-10-16T14:00:00-06:00" or "2005-10-16".
includeDocuments	(Y/N) – Option - Causes the method to return associated documents in the zip file. For Solicit only. If you leave this blank, the default is "N". *Not relevant for REST services
returnCountOnly	(Y/N) – Option - Causes the method to return only the count of actions that match the query search criteria. If you leave this blank, the default is "N".

GetDomainValues	
For retrieving valid domain values.	
Query Parameter	Description/Notes

⁴ Reach codes will not be in the ATTAINS database. It will be part of the indexing (RAD).

domainName	Search criterion – Specifies the domain for which values should be retrieved. If it is left blank a list of domains is returned.
context	<p>Search criterion – Filters the list of domain values by some sub-category. Examples:</p> <p>“Give me all Use Class Type domain values where the Context is ‘North Carolina DEQ’”</p> <p>“Give me all Source Name domain values where the Context is ‘USEPA’”</p> <p>If left blank, all values matching the supplied DomainName from all contexts are returned. Accepts multiple values.</p>

GetTransactionHistory	
Returns a summary of the transactions processed by the ATTAINS System	
Query Parameter	Description/Notes
TransactionId*	Search criterion – Specifies one or more specific transactions for which details should be retrieved. Multiple values may be provided.
OrganizationId*	Search criterion – Filters the list of transactions to only those initiated for the specified Organization’s data.
TransactionLaterThanDate	Search criterion – Filters the list of transactions to only those initiated after the provided date. Can be used in conjunction with TransactionEarlierThanDate to retrieve transactions initiated within a date range. If TransactionEarlierThanDate is also provided, then TransactionLaterThanDate must be earlier than TransactionEarlierThanDate. Date format must be like “2005-10-16T14:00:00-06:00” or “2005-10-16”.
TransactionEarlierThanDate	Search criterion – Filters the list of transactions to only those last changed before the provided date. Can be used in conjunction with TransactionLaterThanDate to retrieve transactions initiated within a date range. If TransactionLaterThanDate is also provided, then TransactionEarlierThanDate must be later than TransactionLaterThanDate. Date format must be like “2005-10-16T14:00:00-06:00” or “2005-10-16”.
UserId*	Search criterion – Filters the list of transactions to only those initiated by the specified user. This is the same as the NAASID.
ReturnCountOnly	(Y/N) – Option - Causes the method to return only the count of transactions that match the query search criteria. If you leave this blank, the default is “N”.

9.0 Appendix A – ATTAINS Business and Validation Rules

General

Many of these rules have not been implemented yet. Please see DET and special functionality section for current implementation.

Organizations

There can only be one contact for each ContactType within an Organization. If a contact is passed, any previously saved contact of the same ContactType will be replaced.

Assessment Units

Assessment Units must have at least one water type.

Assessment Unit ID's must be unique within a state. ID's from retired Assessment Units will not be available for reuse.

Assessment Units can be deleted if they have never been linked to an Assessment or Action. If they are not eligible for deletion, they can be retired.

Assessment Units can be renamed. If they are renamed, the change will be stored in the Assessment Unit's history.

The extent (size) of an Assessment Unit can change. This change will not be stored in the Assessment Unit's history, but can be derived from the Assessment Unit snapshots.

An Assessment Units Waterbody Type can be changed. This change will not be stored in the Assessment Unit's history, but can be derived from the Assessment Unit snapshots.

Assessment Units can be corrected to update the size based on GIS information. This change will not be stored in the Assessment Unit's history, but can be derived from the Assessment Unit snapshots.

Assessment Units will be archived each time a linked Integrated Report, Action or Priority changes status. The Assessment Unit information for the archive will be pulled from the current state of the Assessment Unit, except when a linked Integrated Report is in EPA Review. In that case, the snapshot when the Integrated Report moves to Final Action, will use the Assessment Units that are archived with the EPA Review version of the report.

EPA users will not be able to create or modify Assessment Units.

When viewing an Assessment Unit, the system will indicate if the GIS information provided for the AU is older than the date of the last update to the Assessment Unit.

Assessment Units will be available to the public when they are linked to an Integrated Report, Action or Priority that has a status that is viewable by the public.

At least one Assessment unit block is required if AssessmentUnits is reported.

Location information is required if Locations block is reported.

Reporting Cycle

Reporting Cycle submissions should include Assessments for all Assessment Units with each submission. There is no requirement to validate that all defined Assessment Units in the system are included in the submission.

Organizations should not submit/create a new cycle in the system until the previous cycle has reached “Organization Final Action – Submittal” or the organization has decided not to submit for the previous cycle and are submitting a combined report. Once a new cycle is established in the system, organizations will be unable to promote/submit draft or public review cycles from a previous cycle. Progress from Organization Final Action through EPA Final Action may proceed even after a newer cycle has been created.

The Reporting Cycle Text can be any four digit year (yyyy).

When a cycle status is changed, the date of the change will be stored along with the user that changed it.

When a cycle status is changed, a read-only snapshot of the data as it looks at that moment will be stored in the database. This includes a snapshot of the current state of the Assessment Units, except when the cycle status is changing to Final Action, in which case the Assessment Unit snapshot is taken from the EPA review archive.

Changes to Reporting Cycle data when they are “Draft” do not need to be tracked. Changes at all other stages should be tracked.

When Reporting Cycle data has a status of “EPA Review”, organizations cannot make changes to the assessments that are part of a 303(d) list (category 4a, 4b, 4c or 5) or that contain delisting information. Any changes by the organization to these Assessments will be stored in pending tables, and must be approved by EPA before they will be incorporated into the data set. Organizations may make changes to other 305(b) Assessments freely.

Reporting Cycle data with a “Draft” status are only viewable by the owner organization unless they have explicitly shared the data with other ATTAINS organizations. Reporting Cycle data that are submitted as draft through the Exchange Network will be shared with the appropriate EPA Region by default.

Once a cycle has a status of “Final Action”, the all assessment data (including 305(b)) will become read-only for all users of the system.

The data displayed in the public website and services will always be pulled from the most recent snapshot of the data - not the current working copy of the data.

Assessments

The cycle last assessed for an Assessment should be an ~~even-numbered~~ year less than or equal to the current assessment cycle.

At least one designated use with a use attainment value is required for an Assessment.

Uses with a use attainment value of "Not Assessed" should not be linked to parameters.

If a designated use is assigned a use attainment of "Not Supporting", at least one cause must be entered. This cause can be "cause unknown" (from the 3/17 design team meeting). The cause use attainment must be "not meeting criteria".

Uses that have a use support value of "Not Assessed" should not be linked to parameters.

Causes cannot be assigned to uses that are fully supporting unless they are threatened.

Listing Information is required for causes assigned to waters that are on the 303(d) list. If the water is in Category 5 and the Pollutant Indicator = 'Y', then listing information is also required. These listing elements include:

Cycle First Listed

Cycle Scheduled for TMDL or TMDL Priority Name

In order for an Assessment to be a valid category 4A submission, a TMDL report identifier is required.

In order for an Assessment to be a valid category 4B submission, it must have an expected to attain date, as well as a linked "category 4B" type Action.

Causes should have at least one related designated use. This use should match a corresponding designated use from the Assessment.

Parameter Attainment Code is conditionally required if the ParameterStatus = "Cause".

Observed Effects should have at least one related designated use. This use should match a corresponding designated use from the Assessment.

Probable sources should have at least one related cause. This cause must match a corresponding cause from the Assessment.

ATTAINS users will be able to communicate about Assessments via comments. These comments will not be viewable by the public.

Assessments cannot be created on retired/inactive Assessment Units.

Trophic Status should only be entered for a defined subset of Assessment Unit water types, such as lakes.

Assessments linked to a retired Assessment Unit will not be considered valid when the Reporting Cycle status is draft. This linkage is permitted at all other statuses.

Assessments that are linked to TMDL's with a status of "withdrawn" will not be able to move from draft to public review.

If a parameter type = "monitored" the parameter attainment value cannot be "Not Meeting Criteria"

A parameter can only have one status (monitored/observed effect/cause) per AU/use/parameter/season in the same cycle. Same parameter can have a different parameter attainment for each use.

Actions

Action status cannot go backwards. For example, if a state has submitted a "Public Review" Action, they cannot subsequently submit the same Action as "Draft". They may submit as many versions with the same status as they want, and each submission with the same status will be treated as a complete replace.

When an Action status is changed, the date of the change will be stored along with the user that changed it.

When an Action status is changed, or each time it is submitted, a "snapshot" of the Action, along with associated Assessment Units and Statewide Assessments will be taken.

When an Action has "Draft" status, changes do not need to be tracked.

When an Action has a "Public Review" or "Final" status, changes can be made, but need to be tracked.

When an Action has a status of "EPA Review", changes by EPA will be tracked. Changes by the owning organization (if not EPA) will be stored in pending tables and must be approved by EPA in order to become part of the data set.

Actions with a status of "Final Action" cannot be changed.

Either Planned Completion Date or Actual Completion date is required.

At least one Statewide Assessment or Assessment Unit/Parameter combination must be assigned to an Action.

Assessment Units linked to actions should be active Assessment Units, or inactive/retired Assessment Units that were listed as category 5 from the last cycle that had a minimum status of Organization Final Action.

When an Assessment unit is linked to an Action (Specific Waters), at least one parameter must be selected if the Action has pollutants.

If the Action Type is “TMDL Complete” then at least one pollutant must be associated with a "Specific Water". This pollutant should be one of the pollutant values linked to the Action.

Actions that are not TMDLs “Approved” – there is no final action. Some Actions will never be reviewed by EPA, and will stay in the “Final” status. Actions that will be approved by EPA are actions that count towards one of the measures.

In order for an Action with a TMDL Action Type to be considered final, at least one pollutant is required.

If the Pollutant Source Type Code is “Point Source” and the action type is "TMDL Complete" then NPDES details are required.

In order for an Action with a “TMDL” action type to be considered final, Waste Load Allocation information must be provided if the Pollutant Source Type is “Point source” and NPDES information is provided.

ATTAINS users will be able to communicate about Actions via comments. These comments will not be viewable by the public.

The associated pollutants for a specific water (AU/parameter combination) should be limited to the list of pollutants that are identified in the Action.

The system should validate that the cycle identified for linked statewide assessments is correct.

Actions linked to a retired Assessment Unit will be considered invalid when their status is "draft", unless the Assessment Unit was listed as category 5 in the previous reporting cycle with a minimum cycle status of Organization Final Action.

The system will prevent the creation of a TMDL that addresses the same pollutant on the same water as an existing TMDL (unless the existing TMDL has a status of 'withdrawn'),

All Actions must have a linked document representative of their type in order to move from draft to public review. So for example, TMDL's are required to have a TMDL document. (need to work with EPA to identify appropriate document types to require for each type of Action).

Users can not add a pollutant twice unless season information is provided.

A document is required if the Action type is TMDL Complete

10.0 Appendix B – Example Processing Reports

Example Processing Report for a Successful Submission:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="http://www.epa.gov/storet/download/validation.xsl"?>
<ProcessingReport>
  <TransactionIdentifier>6ede652a-6fbc-4af4-9972-37e1d8316ba4</TransactionIdentifier>
  <Status>Succeeded</Status>
  <ProcessingSoftware Version="0.1">ATTAINS Node</ProcessingSoftware>
  <ProcessingSoftware Version="0.1">ATTAINS Database</ProcessingSoftware>
  <Counts>
    <Error>0</Error>
    <Warning>0</Warning>
    <Action Action="Delete">0</Action>
    <Action Action="Update">0</Action>
    <Action Action="Insert">0</Action>
    <Priority Action="Delete">0</Priority>
    <Priority Action="Update">0</Priority>
    <Priority Action="Insert">0</Priority>
    <AssessmentUnit Action="Delete">0</AssessmentUnit>
    <AssessmentUnit Action="Update">1</AssessmentUnit>
    <AssessmentUnit Action="Insert">0</AssessmentUnit>
    <ReportingCycle Action="Update">0</ReportingCycle>
    <ReportingCycle Action="Delete">0</ReportingCycle>
    <ReportingCycle Action="Insert">0</ReportingCycle>
  </Counts>
  <Log>
    <LogDetail>
      <Type>Message</Type>
      <Text>Parse and Load started at Thu Mar 17 12:47:48 MDT 2016</Text>
      <Context></Context>
    </LogDetail>
    <LogDetail>
      <Type>Message</Type>
      <Text>Parse and Load completed at Thu Mar 17 12:57:28 MDT 2016</Text>
      <Context></Context>
    </LogDetail>
  </Log>
  <ProcessingFailures/>
</ProcessingReport>
```

Example Processing Report for a Failed Submission:

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<?xml-stylesheet type="text/xsl" href="http://www.epa.gov/storet/download/validation.xsl"?>
<ProcessingReport>
  <TransactionIdentifier>6ede652a-6fbc-4af4-9972-37e1d8316ba4</TransactionIdentifier>
  <Status>Failed</Status>
  <ProcessingSoftware Version="0.1">ATTAINS Node</ProcessingSoftware>
  <ProcessingSoftware Version="0.1">ATTAINS Database</ProcessingSoftware>
  <Counts>
    <Error>0</Error>
    <Warning>0</Warning>
    <Action Action="Delete">0</Action>
    <Action Action="Update">0</Action>
    <Action Action="Insert">0</Action>
    <Priority Action="Delete">0</Priority>
    <Priority Action="Update">0</Priority>
    <Priority Action="Insert">0</Priority>
    <AssessmentUnit Action="Delete">0</AssessmentUnit>
    <AssessmentUnit Action="Update">1</AssessmentUnit>
    <AssessmentUnit Action="Insert">0</AssessmentUnit>
    <ReportingCycle Action="Update">0</ReportingCycle>
    <ReportingCycle Action="Delete">0</ReportingCycle>
    <ReportingCycle Action="Insert">0</ReportingCycle>
  </Counts>
  <Log>
    <LogDetail>
      <Type>Message</Type>
      <Text>Parse and Load started at Thu Mar 17 12:47:48 MDT 2016</Text>
      <Context></Context>
    </LogDetail>
    <LogDetail>
      <Type>Error</Type>
      <Text>Failed to Update Assessment Unit with Id: AU-12345-A: missing required field:
AssessmentUnitName</Text>
      <Context></Context>
    </LogDetail>
    <LogDetail>
      <Type>Message</Type>
      <Text>Parse and Load completed at Thu Mar 17 12:57:28 MDT 2016</Text>
      <Context></Context>
    </LogDetail>
  </Log>
  <ProcessingFailures>
    <AssessmentUnitIdentifier>AU-12345-A</AssessmentUnitIdentifier>
  </ProcessingFailures>
</ProcessingReport>

```

11.0 Appendix C - ATTAINS Schema Diagram

Note: In the first diagram you can Ctrl+Click on the plus signs (“+”) to hyperlink to the corresponding breakout of the given node.

