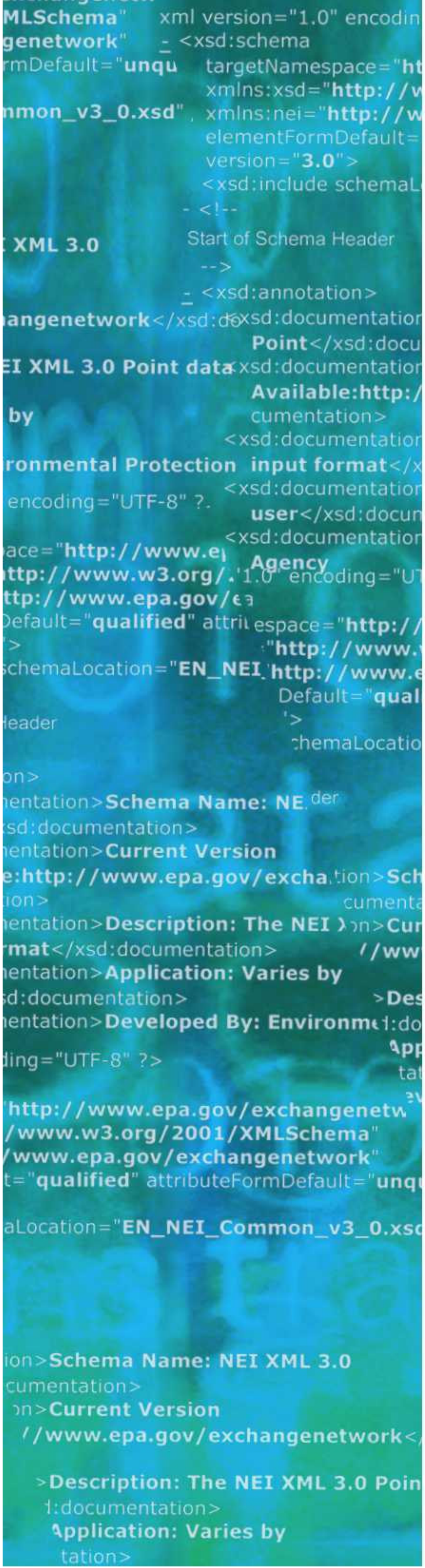


Substance Registry Services (SRS)

Flow Configuration Document

Version: 3.0d
Revision Date: 12/4/2013

Prepared for: EPA
Prepared by: CGI Federal

Environmental Information
eXchange
Network

THIS PAGE INTENTIONALLY LEFT BLANK

Table of Contents

1 Component Alignment

The major and minor version number of the FCD should be identical to the major and minor version number of the current schema supported by this exchange. If the FCD is changed without changes to the schema, then a revision number should be used on the FCD's version number.

1.1 Flow Component Versions Currently Supported

Component	Version(s) Supported	Explanation (optional)
FCD	**3.0**	**Replacement of previous versions.**
Schema	**3.0**	**Replacement of previous versions.**
DET	**3.0**	**Replacement of previous versions.**
GetSubstanceByALTIDListAcronym	**3.0**	**Query data service.**
GetSubstanceByCAS	**3.0**	**Query data service.**
GetSubstanceByEPAID	**3.0**	**Query data service.**
GetSubstanceByInternalTrackingNumber	**3.0**	**Query data service.**
GetSubstanceByName	**3.0**	**Query data service.**
GetSubstanceByTSN	**3.0**	**Query data service.**
GetSubstanceMatchByName	**3.0**	**Query data service.**
GetSubstancesByCAS	**3.0**	**Solicit data service.**
GetSubstancesByCategoryClassifications	**3.0**	**Solicit data service.**
GetSubstancesByCharacteristics	**3.0**	**Solicit data service.**
GetSubstancesByEPAIDs	**3.0**	**Solicit data service.**
GetSubstancesByInternalTrackingNumber	**3.0**	**Solicit data service.**
GetSubstancesByListAcronym	**3.0**	**Solicit data service.**
GetSubstancesByNamesExact	**3.0**	**Solicit data service.**
GetSubstanceCASByListAcronym	**3.0**	**Solicit data service.**
GetSubstanceListAcronymsByAcronym	**3.0**	**Solicit data service.**
GetPMNACCInformation	**3.0**	**Solicit data service.**
ListSubstanceAddition	**3.0**	**Submit data service.**

2 Introduction

2.1 Flow Identification

Flow Name:

Substance Registry Services (SRS)

Flow Owner:

John Harman
USEPA / Office of Environmental Information (OEI)

Flow Owner Contact Information:

John Harman
USEPA Headquarters
Ariel Rios Building
1200 Pennsylvania Avenue, N. W.
Mail Code**:** 2823T
Washington, DC 20460

202.566.0748
Harman.John@epa.gov

2.2 Background

A primary focus for the EPA is tracking, monitoring, or regulating substances in the environment. This might include discharges of volatile organic chemicals into the air, monitoring the presence of biological contaminants in the water, regulating pesticides usage, or working to minimize potential hazards from pollutants at a Superfund site.

In addition, the United States has enacted a variety of laws to address substances in the environment. These include the Clean Air Act, the Clean Drinking Water Act, the Toxics Substances Control Act, plus many others. EPA has created programs to implement these laws.

Substances, particularly chemicals, often have multiple valid names. For example, the scientific name for one chemical is Benzene, methyl. Other names for this chemical include Toluene, Toluol, Phenylmethane, and Methyl benzene. Biological organisms likewise can have a variety of names.

The Substance Registry Services (SRS – www.epa.gov/srs) is the Environmental Protection Agency's central registry for information about substances that are tracked or regulated by EPA or other sources. It is the authoritative resource for basic information about chemicals, biological organisms, and other substances of interest to EPA and its state and tribal partners.

The SRS makes it possible to identify which EPA data systems, environmental statutes, or other sources have information about a substance and which synonym is used by that system or statute. As a result, the SRS makes it possible to map substance data across EPA programs regardless of synonym. Prior to implementing SRS, discovering which EPA program tracked or regulated a particular substance was time-consuming and often required a degree of understanding of chemistry.

The SRS system provides a common basis for the identification of and information about:

- Chemicals
- Biological organisms
- Physical properties
- Miscellaneous objects

The SRS system is a part of EPA's centralized System of Registries (SoR – www.epa.gov/sor). The System of Registries provides access to the EPA's core registry systems.

Recognizing the diversity of names for substances, EPA created two data standards: the Chemical Identification Data Standard for chemicals and the Biological Taxonomy Data Standard for biological organisms. The SRS system supports and conforms to both standards. Each of these standards requires EPA to work with its state partners to identify a core set of identifiers. For chemicals, one of those identifiers is the scientific name (SRS Systematic Name), and one is a common name (SRS Registry Name) if it exists. In the example above, the SRS Systematic Name is Benzene, methyl-, and the SRS Registry Name is Toluene. Toluol, Phenylmethane and Methyl benzene are maintained in SRS as synonyms.

SRS identifies:

- Which EPA program or environmental statute tracks or regulates a given substance

- The particular name (synonym) used for the substance

The SRS system provides the ability to create and maintain lists of substances, allowing a given, standardized substance to be a member of any number of lists. List membership includes information about the substance that is specific to the purpose of the list. For example, the name (synonym) that is commonly used for that substance in relation to the list's purpose, the reason the substance is being regulated, and what alternate identifiers might commonly be used for a substance are maintained as part of list membership.

EPA programs have developed substance lists within SRS to track substances related to their particular program area. In addition, lists have been developed within SRS to track substances in relation to the laws enacted by the United States to monitor and control substances in the environment. The following are examples of lists of substances within the SRS:

- AQS – Air Quality System
- ATSDR – Agency for Toxic Substances and Disease Registry
- CAA 109 – National Ambient Air Quality Standards
- TRI – Toxic Chemical Release Inventory
- TTO – Toxic Organics

New EPA programs have begun to adopt standardized substance names for their substance lists. Some older EPA programs have replaced their names for substances with the standardized names.

The SRS system provides a web-based user interface (www.epa.gov/srs) for users to interact with SRS to retrieve, research, and (for authorized users) maintain substance information and lists within the SRS system. The SRS data flow and its data services defined by this document enable interaction with the SRS system through machine-to-machine interfaces to automate retrieval, collaboration and synchronization of substance information between systems.

2.3

2.3 Getting Started with the SRS Data Flow

Before using the SRS data flow, developers must obtain an NAAS account from the CDX Node Helpdesk. To obtain an NAAS account, contact nodehelpdesk@epacdx.net and provide your name, email address, organization, telephone number, and a brief description of your planned use of the account (i.e., accessing the SRS data flow). The Helpdesk will contact you with details of your account and provide access instructions.

Documentation on the SRS Web Services are available from the Exchange Network site (http://www.exchangenetwork.net/data-exchange/srs/). The XML Schema shows the format of returned data and this document, the Flow Configuration Document, provides details on the requests that can be handled by the web service.

The WSDL provides standard services that all Exchange Network Nodes implement. Each data flow works within these standard services implementing Submit, Solicit or Query services. The most commonly used services are the Query and Solicit services. Query services are intended for immediate responses to the requests. Solicit services are intended for responses that may take some time to process. The Solicit request goes through a workflow that provides notification when the data is ready.

The basic workflow of each type of service is as follows:

Query:
Authenticate
Query

Solicit:
Authenticate
Solicit
GetStatus
Download

The first step in calling each of the services is a call to Authenticate to get a securityToken. That securityToken is then included with all other requests.

The WSDL can be found here: https://cdxnodengn.epa.gov/ngn-enws20/services/NetworkNode2Service?wsdl

The endpoint can be found here: https://cdxnodengn.epa.gov/ngn-enws20/services/NetworkNode2Service

Below are a few sample SOAP requests. EPA recommends using SOAP UI for testing.

Authenticate:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.exchangenetwork.net/schema/node/2">
  <soap:Header/>
  <soap:Body>
    <ns:Authenticate>
```

```
      <ns:userId>andrew.w.hampton@cgifederal.com</ns:userId>
      <ns:credential>password </ns:credential>
      <ns:domain></ns:domain>
      <ns:authenticationMethod>Password</ns:authenticationMethod>
    </ns:Authenticate>
  </soap:Body>
</soap:Envelope>
```

Query:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.exchangenetwork.net/schema/node/2"
xmlns:srs="http://www.exchangenetwork.net/schema/SRS/3/">
  <soap:Header/>
  <soap:Body>
    <ns:Query>
      <ns:securityToken>token</ns:securityToken>
      <ns:dataflow>SRS</ns:dataflow>
      <ns:request>GetSubstanceByCAS</ns:request>
      <ns:maxRows>0</ns:maxRows>
      <ns:parameters parameterName="CASRegistryNumber">543-90-8</ns:parameters>
    </ns:Query>
  </soap:Body>
</soap:Envelope>
```

Solicit:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.exchangenetwork.net/schema/node/2">
  <soap:Header/>
  <soap:Body>
    <ns:Solicit>
      <ns:securityToken>token</ns:securityToken>
      <ns:dataflow>SRS</ns:dataflow>
      <ns:request>GetSubstancesByListAcronym</ns:request>
      <ns:parameters parameterName="ListAcronym" parameterType="String"
parameterEncoding="None">WDNR</ns:parameters>
    </ns:Solicit>
  </soap:Body>
</soap:Envelope>
```

GetStatus:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.exchangenetwork.net/schema/node/2">
  <soap:Header/>
  <soap:Body>
    <ns:GetStatus>
      <ns:securityToken>token</ns:securityToken>
      <ns:transactionId>_c1b655fe-8b8f-4bc0-ad1e-79e0bd4b9704</ns:transactionId>
    </ns:GetStatus>
  </soap:Body>
</soap:Envelope>
```

Download:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.exchangenetwork.net/schema/node/2"
xmlns:xm="http://www.w3.org/2005/05/xmlmime">
  <soap:Header/>
  <soap:Body>
    <ns:Download>
      <ns:securityToken>token</ns:securityToken>
       <ns:dataflow>SRS</ns:dataflow>
       <ns:transactionId>transactionID</ns:transactionId>
    </ns:Download>
  </soap:Body>
</soap:Envelope>
```

2.4 Data Flow Overview

The Substance Registry Services (SRS) data flow provides data services that can be used by trading partners to retrieve substance information from the SRS system based upon a variety of identifiers, substance characteristics, and list membership. The data services also provide the ability for authorized users to maintain pre-defined lists of substances within the SRS system. Implementation of these data services under a trading partner agreement makes it possible to use machine-to-machine interfaces to ensure substance information maintained in external systems is up-to-date with the latest EPA central registry of substances (SRS).

The SRS data services are provided through the EPA's Exchange Network (www.exchangenetwork.net) and are accessed through the Exchange Network's Central Data Exchange Node (the "CDX node") using the Exchange Network's node web service specifications. Version 3.0 of the SRS data services supports version 2.0 of the Exchange Network's node web service specifications. Version 3.0 of the SRS data services are also backward compatible with version 1.1 of the Exchange Network's node web service specifications; however, it is strongly encouraged that trading partners utilize Exchange Network node client or node technology that supports version 2.0 of the specifications.

Please note: Version 3.0 and later of the SRS data services are NOT backward compatible with any prior version of the SRS data services.

The data services provided by the SRS data flow are grouped into two primary categories as identified in the following table (Table 2-1):

Table 2-1 SRS Data Service Categories

Category	Description
Data Processing Services	Services are provided to maintain substance list and synonym information within the SRS system by authorized users through the submission of an XML document containing the maintenance request. At present, only the addition of pre-defined substances to pre-defined lists within the SRS system is supported. Future releases of this data flow may provide enhanced list maintenance services.
Data Publishing Services	Services are provided to retrieve substance information including standard substance identifiers as defined by the Chemical Identification Data Standard for chemicals and the Biological Taxonomy Data Standard for biological organisms (as well as identifiers defined by this flow for physical property, miscellaneous object, and unknown substance information maintained by the SRS system), as well as substance synonym / list membership information.

	For chemical and biological substances, the XML that is returned for a substance conforms to an extension of the Exchange Network’s Shared Schema Components Version 2.0 for chemical and biological substances.

2.4.1 Flow Service Architecture

The Substance Registry Services (SRS) data flow is provided through the EPA's Central Data Exchange (CDX) and is implemented as an integration of a number of system components as depicted in the following architecture diagram (Figure 2-1):

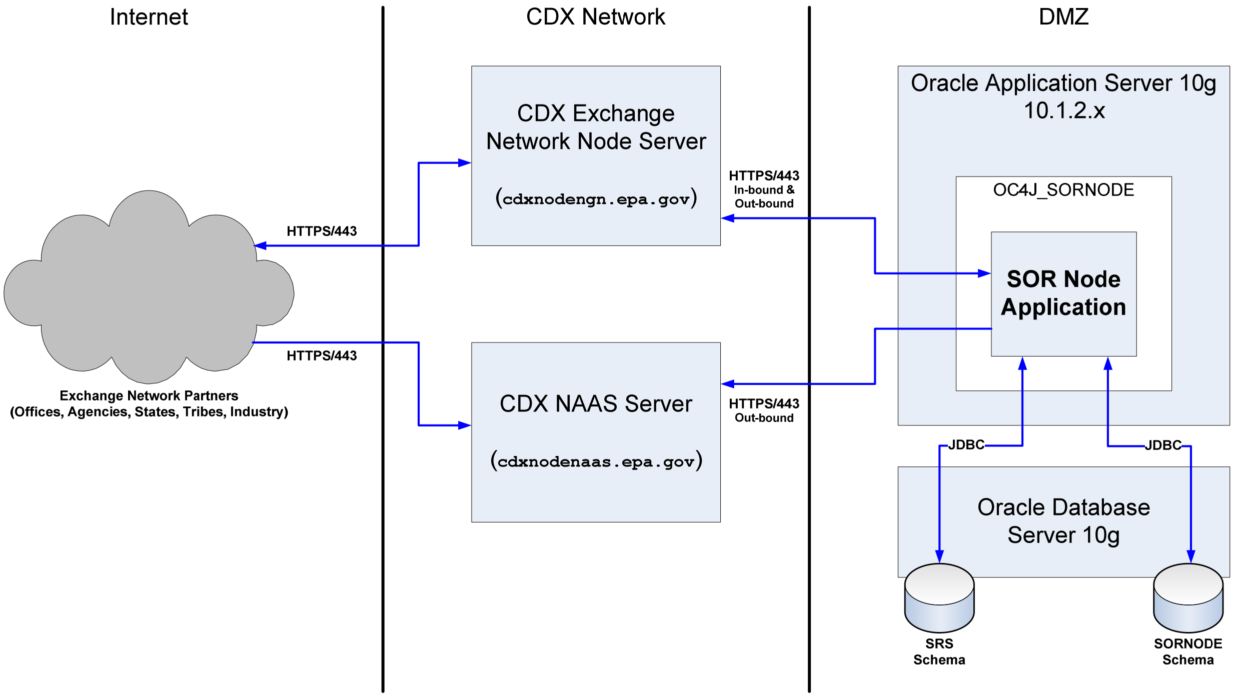

Internet
CDX Network
DMZ
CDX Exchange Network Node Server
(cdxnodengn.epa.gov)
CDX NAAS Server
(cdxnodenaas.epa.gov)
HTTPS/443
HTTPS/443
HTTPS/443 In-bound & Out-bound
HTTPS/443 Out-bound
Exchange Network Partners (Offices, Agencies, States, Tribes, Industry)
Oracle Application Server 10g 10.1.2.x
OC4J_SORNODE
SOR Node Application
JDBC
JDBC
Oracle Database Server 10g
SRS Schema
SORNODE Schema

Figure 2-1 SRS Flow Service Architecture Diagram

Partner systems and users (via node client or node technologies) interface directly with the CDX Exchange Network Node (the "CDX node") and the CDX NAAS (Network Authentication and Authorization Services) Server to access the Substance Registry Services data services. All interactions are performed using web services that conform to the Exchange Network's node web service specifications.

The CDX node handles the appropriate communication with the SRS system (i.e., the SOR Node Application) to satisfy the data service request and respond to the requestor. This includes handling all multi-step operations used by requestors to determine the status of a prior request and download its results.

2.4.2 Flow Service Processing – Data Processing

Data processing services are asynchronous services. The requestor submits the request and later determines the results of the request. The following diagram (Figure 2-2) depicts the general flow of processing performed when submitting data to CDX for processing by the SRS data service:

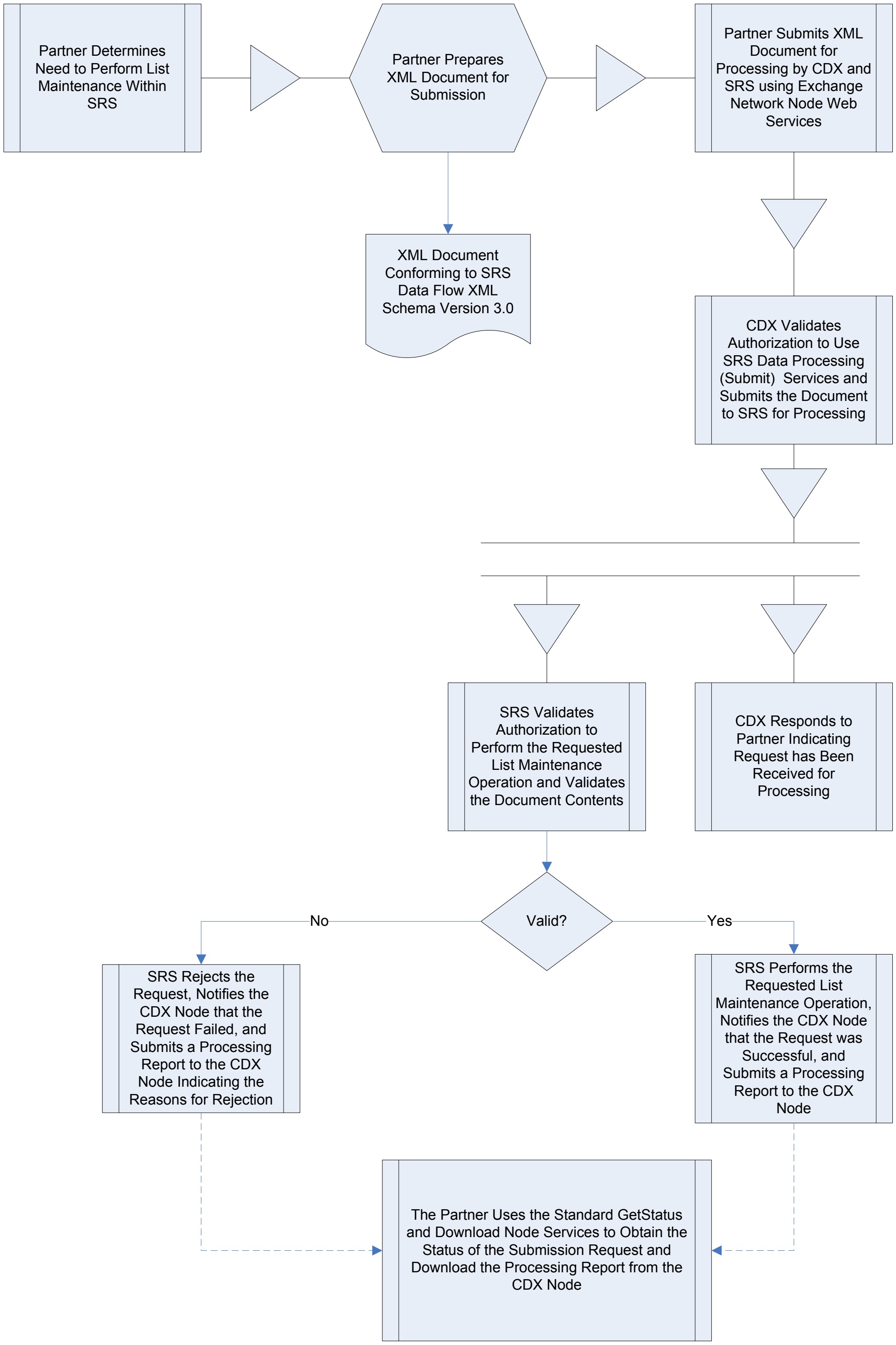

Partner Determines Need to Perform List Maintenance Within SRS
Partner Prepares XML Document for Submission
Partner Submits XML Document for Processing by CDX and SRS using Exchange Network Node Web Services
XML Document Conforming to SRS Data Flow XML Schema Version 3.0
CDX Validates Authorization to Use SRS Data Processing (Submit) Services and Submits the Document to SRS for Processing
SRS Validates Authorization to Perform the Requested List Maintenance Operation and Validates the Document Contents
CDX Responds to Partner Indicating Request has Been Received for Processing
Valid?
No
Yes
SRS Rejects the Request, Notifies the CDX Node that the Request Failed, and Submits a Processing Report to the CDX Node Indicating the Reasons for Rejection
SRS Performs the Requested List Maintenance Operation, Notifies the CDX Node that the Request was Successful, and Submits a Processing Report to the CDX Node
The Partner Uses the Standard GetStatus and Download Node Services to Obtain the Status of the Submission Request and Download the Processing Report from the CDX Node

Figure 2-2 SRS List Maintenance Data Processing Flow

2.4.3 Flow Service Processing – Data Publishing

The SRS data flow provides both Query and Solicit data publishing services. The SRS data services defined as Query data services are designed to return substance information about one or a relatively few substances in a single invocation of the data service. These Query services are synchronous data services: the service requestor (node client) waits until the operation is completed and receives the results of the data service in response to the initial request. The following diagram (Figure 2-3) depicts the general flow of processing performed when issuing a query data service to CDX for processing by the appropriate SRS data service:

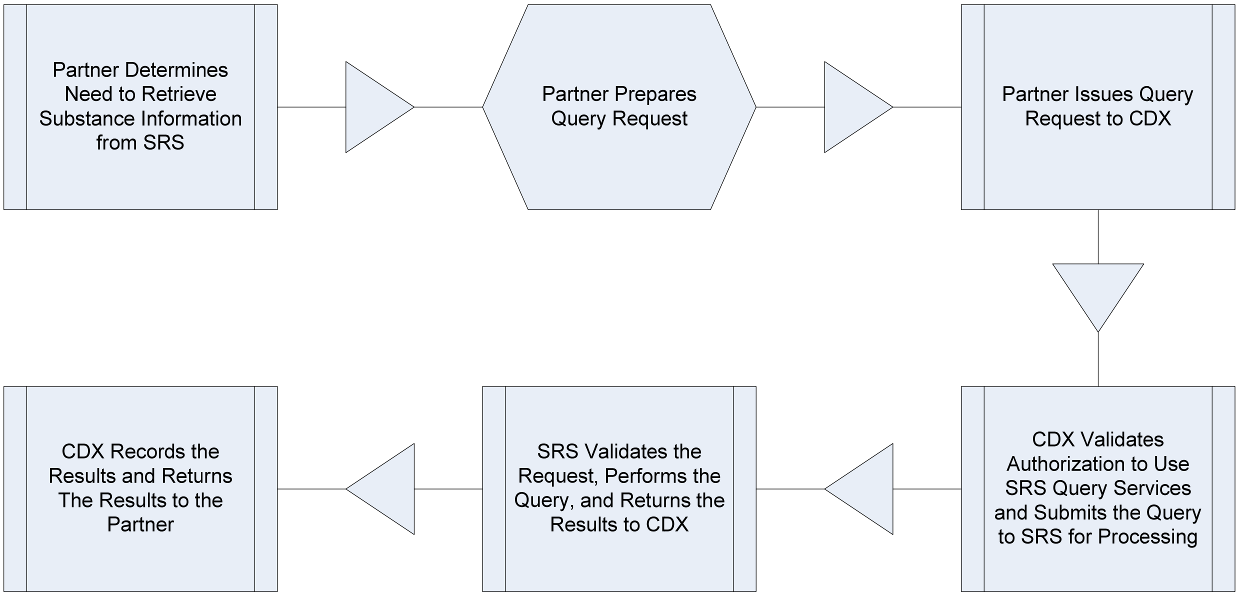

Partner Determines Need to Retrieve Substance Information from SRS
Partner Prepares Query Request
Partner Issues Query Request to CDX
CDX Validates Authorization to Use SRS Query Services and Submits the Query to SRS for Processing
SRS Validates the Request, Performs the Query, and Returns the Results to CDX
CDX Records the Results and Returns The Results to the Partner

Figure 2-3 SRS Query Data Service Processing

The SRS data services defined as Solicit data services are designed to return substance information for any number of substances. Solicit data services are designed to operate asynchronously: the service requestor receives a response from CDX that their request has been received along with a transaction identifier for their request. Their request is then submitted to SRS for processing. Later, the requestor can use standard Node web services (GetStatus and Download) to determine the status of the request and download the results of the request. The following diagram (Figure 2-4) depicts the general flow of processing performed when issuing a query data service to CDX for processing by the appropriate SRS data service:

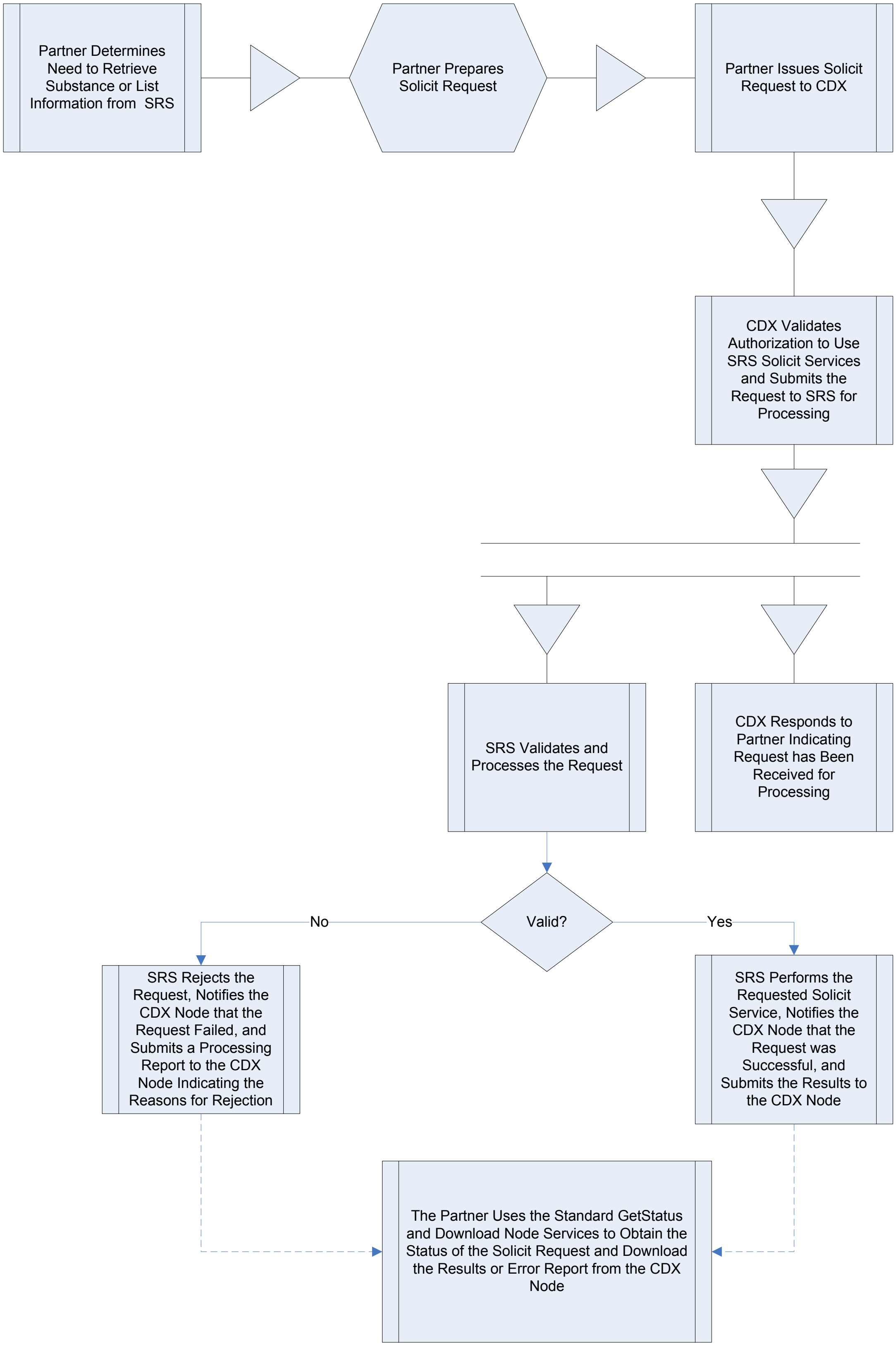

Partner Determines Need to Retrieve Substance or List Information from SRS
Partner Prepares Solicit Request
Partner Issues Solicit Request to CDX
CDX Validates Authorization to Use SRS Solicit Services and Submits the Request to SRS for Processing
SRS Validates and Processes the Request
CDX Responds to Partner Indicating Request has Been Received for Processing
Valid?
No
Yes
SRS Rejects the Request, Notifies the CDX Node that the Request Failed, and Submits a Processing Report to the CDX Node Indicating the Reasons for Rejection
SRS Performs the Requested Solicit Service, Notifies the CDX Node that the Request was Successful, and Submits the Results to the CDX Node
The Partner Uses the Standard GetStatus and Download Node Services to Obtain the Status of the Solicit Request and Download the Results or Error Report from the CDX Node

Figure 2-4 SRS Solicit Data Service Processing

2.5 Flow Access and Security

All service requests must be accompanied by a valid NAAS security token per the Exchange Network's Node 2.0 specifications. All partners must be authorized to NAAS and receive a valid security token before any of the SRS data services can be invoked.

In order to acquire a valid security token (via the partner's Node client technology using the Node specification's authenticate operation), the user must have the appropriate NAAS security policies in place and associated with the EPA's CDX node and the SRS data flow. To obtain access:

- Contact the CDX help desk for more information.
- Contact the SRS flow owner for access (see section 2.1).

2.5.1 Access to Data Processing Services

In order to utilize a SRS list maintenance data processing service (see section 4.1), two authorizations must be acquired by the partner for the partner's NAAS account:

- A NAAS security policy for the NAAS account (NAAS 3.0 for Node 2.0 clients) that permits the account to perform node Submit operations to the CDX node for the SRS data flow.

- The same NAAS account user ID must also be authorized within the SRS Node Application system to perform the specified list maintenance operation (e.g., ListSubstanceAddition) against the substance list (specified as the ListAcronym element in submitted XML documents) specified in the request.

2.5.2 Access to Data Publishing Services

In order to utilize a SRS data publishing service (see section 5), the partner's NAAS account must be authorized to perform the desired data publishing service:

- A NAAS security policy for the NAAS account (NAAS 3.0 for Node 2.0 clients) that permits the account to perform node Query or Solicit operations to the CDX node for the SRS data flow and the desired service (request) name.

2.6 Flow-level Business Rules

Current Business Rules: Not applicable. See the sections that follow for relevant descriptions of business rules associated with the SRS data services.

Fault Follow-up Actions: Each data service currently defined as part of the SRS data service is atomic and re-runable if an error condition is detected. Data publishing services query data from SRS; if the request fails for any reason, the request can be re-submitted. If a problem persists, contact the CDX help desk. For the list maintenance data publishing services, a submission can be re-submitted if a prior submission fails.

2.7 Additional Flow Tools and Resources

There are no additional tools required or provided at this time.

3 Web Services Overview

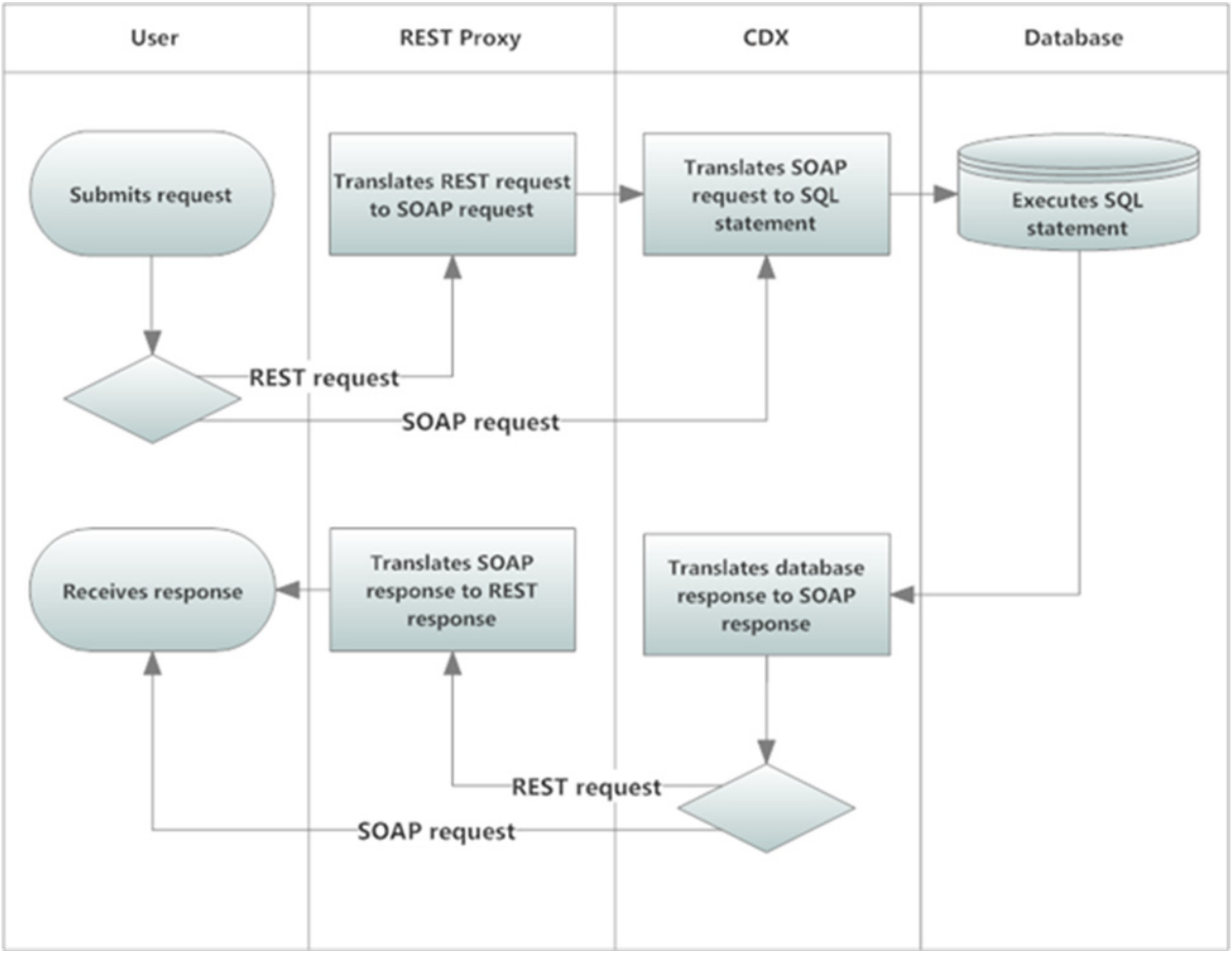
A web service is a method of communication between software applications using open protocols over the web. A common function of a web service is to query data from a centralized data source. Such query web services return data requests in XML format only; they do not display the data in a user interface. Data returned in a standardized XML format allows independent software applications to consume the same data in a consistent manner suited to their respective needs.

Web services generally run on either the Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) protocol. **While SOAP is supported for all SRS Data Flow services, REST support is only available for query data services. In other words, query data services are supported by both SOAP and REST, while all other services are SOAP-only.**

A REST web service can be called by entering the web service's URL into the browser. All service parameters can be set via URL parameters. The server interprets the data in the URL and responds with the requested data in XML.

The general flow of processing performed when issuing a request to an SRS Web Service is depicted in Exhibit 3-1 below.

Exhibit 3-1 SRS Services Query Data Service Processing


User
REST Proxy
CDX
Database
Submits request
Translates REST request to SOAP request
Translates SOAP request to SQL statement
Executes SQL statement
REST request
SOAP request
Receives response
Translates SOAP response to REST response
Translates database response to SOAP response
REST request
SOAP request

4 Data Processing

The SRS data processing services are designed to allow authorized users to update information in SRS by submitting XML documents containing updated SRS information to the SRS Node Application using the Exchange Network via the CDX Node. The data submission process is an asynchronous process; partners attach and submit their XML document using node client technology and receive a response indicating that their document has been received for processing. Later, the partner can use the standard GetStatus and Download node operations to check on the status of their submission (e.g., Pending, Processing, Completed or Failed).

If a submission fails with errors, the partner can download the Node20.Error document for submission to review the processing errors. Refer to section 2.4.2 for more information.

4.1 List Maintenance Data Processing

The SRS list maintenance data processing services are designed to allow authorized users to maintain substance lists within SRS. Users must be have NAAS authorization as well as SRS authorization in order to perform these services. Refer to section 2.5 for more information.

4.1.1 ListSubstanceAddition

The ListSubstanceAddition data service allows an authorized partner to add a substance that is already defined in SRS to a substance list that is also already defined in SRS. Adding a substance to a list in SRS results in the creation of a synonym for that substance; the synonym is associated with the list, not the substance directly. In order to create the synonym, the appropriate fields must be supplied. See the business rules section below for more information.

Type: Submit

XML Header Usage: The Exchange Network Header version 2.0 is required for documents submitted with this service. Refer to the Exchange Network documentation for the header for more information.

Header Elements, Description and Examples

Element	Required?	Example	EN Header Version
AuthorName	Yes	Harman.John@epa.gov	2.0
OrganizationName	Yes	SOR	2.0
DocumentTitle	Yes	ListSubstanceAddition	2.0
CreationDateTime	Yes	2008-10-29T00:52:14	2.0

Custom Header Elements, Description and Examples
There are no custom header elements for this service.

Header Payload Operation Attribute Values
The payload operation attribute is not required for this service.

Data Service-level Business Rules:

The following table defines the business rules associated with each XML field in the XML document payload for this data service. See Figure 4-1 for an example document. The format of the document must conform to and is validated using the SRS XML schema.

Field Name (XML Tag)	Description / Business Rules
`ListAcronym`	Mandatory List acronym must already exist in SRS. The submission will be rejected if the list is not already defined in SRS.
`EPAIdentifier`	Mandatory Provide a value for only one these identifier fields. A substance with this identifier value must already exist in SRS. For CASRegistryNumber, there must be a substance that has the specified value as its current CAS registry number.
`InternalTrackingNumber`	
`CASRegistryNumber`	
`SynonymName`	Mandatory The name for the substance that will be associated with the specified substance list. Name cannot exceed 2000 characters.
`AlternateIdentifierList`	Optional An XML tag containing one or more `AlternateIdentifier` chlld tags containing an alternate identifier value for the synonym / substance. If an `AlternateIdentifier` tag is specified, it must also specify the `AlternateIdentifierContext` attribute whose value is a context that is already defined in SRS.
`SynonymEffectiveStartDate`	Optional Date the synonym becomes effective. Format: YYYY/MM/DD
`SynonymEffectiveEndDate`	Optional Date the synonym is no longer effective. Format: YYYY/MM/DD
`RegulationReasonText`	Mandatory The reason the substance is being regulated (or added to the list).
`NameType`	Optional Synonym name type. If supplied, must be either "EPA Registry" or "Systematic"
`SynonymCommentText`	Characters cannot exceed 2000 characters.

4.1.1.1 Request

Dataflow Name: SRS

Flow Operation: ListSubstanceAddition

Recipient: Not currently supported.

NotificationURI: Not currently supported.

Input: See the flow schemas SRS_ListMaintenance_v3.0 and SRS_ListSubstanceAddition_v3.0. The input document format supplied to this service (attached to the Node submit operation) may be XML, or the same XML document in ZIP format. Multiple `ListSubstanceAddition` tags (i.e., multiple substance additions to lists) may be specified in a single `Payload` tag. Only the first `Payload` tag is processed.

```
<hdr:Document xmlns:hdr="http://www.exchangenetwork.net/schema/header/2"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation=”http://www.exchangenetwork.net/schema/header/2
  http://www.exchangenetwork.net/schema/header/2”
  id="ID01">
  <hdr:Header>
    <hdr:AuthorName>Harman.John@epa.gov</hdr:AuthorName>
    <hdr:OrganizationName>SOR</hdr:OrganizationName>
    <hdr:DocumentTitle>ListSubstanceAddition</hdr:DocumentTitle>
    <hdr:CreationDateTime>2008-10-29T00:52:14</hdr:CreationDateTime>
  </hdr:Header>
  <hdr:Payload>
    <SRS:ListMaintenance
      xmlns:SRS="http://www.exchangenetwork.net/schema/SRS/3"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="http://www.exchangenetwork.net/schema/SRS/3
      http://www.exchangenetwork.net/schema/SRS/3">
      <SRS:ListSubstanceAddition>
        <SRS:ListAcronym>SAMPLE LIST</SRS:ListAcronym>
        <SRS:EPAIdentifier>E60</SRS:EPAIdentifier>
        <SRS:SynonymName>SAMPLE SYNONYM A</SRS:SynonymName>
        <SRS:RegulationReasonText/>
        <SRS:AlternateIdentifierList>
          <SRS:AlternateIdentifier
            AlternateIdentifierContext="SAMPLE">12345</SRS:AlternateIdentifier>
        </SRS:AlternateIdentifierList>
      </SRS:ListSubstanceAddition>
      <SRS:ListSubstanceAddition>
        <SRS:ListAcronym>SAMPLE LIST</SRS:ListAcronym>
        <SRS:EPAIdentifier>E60</SRS:EPAIdentifier>
        <SRS:SynonymName>SAMPLE SYNONYM B</SRS:SynonymName>
        <SRS:RegulationReasonText/>
        <SRS:AlternateIdentifierList>
        <SRS:AlternateIdentifier
          AlternateIdentifierContext="SAMPLE">12345</SRS:AlternateIdentifier>
        </SRS:AlternateIdentifierList>
       </SRS:ListSubstanceAddition>
    </SRS:ListMaintenance>
  </hdr:Payload>
</hdr:Document>
```

Figure 4-1 SRS Sample ListSubstanceAddition XML Document Submission

4.1.1.2 Response

Return:

The standard Node response to the initial submit request will be the standard transaction ID generated for the request, along with a status of the request (typically this will be 'Received').

Submission requests are processed asynchronously. The requestor can provide the transaction ID for the request to the standard Node GetStatus and Download operations to determine the processing status of the submission request. If errors are detected (i.e., the status of the submit request is Failed), the requestor can download the list of errors detected while processing the request by using the Download operation and downloading the document named 'Node20.Error'.

Error Conditions and Return:

If SRS detects errors related to the business rules for this service defined in section 4.1.1, it will record the errors in a document named 'Node20.Error'. This document can be downloaded using the standard Node Download operation supplying the original transaction ID and the document named 'Node20.Error' (FLAT format).

5 Data Publishing

The SRS data publishing services are designed primarily to publish information about substances defined within the SRS system. Substance information is published as an XML document. A published XML document may contain information about zero, one or more substances as defined in SRS based upon the criteria (parameters) provided by the partner when requesting the publication of substance information (i.e., when invoking an SRS data publishing data service). Published XML documents conform to the XML schema defined for the SRS data flow; see section 7 for more information.

The data publishing services provided as part of the SRS data flow are invoked either synchronously (Query) or asynchronously (Solicit) based upon which service is being invoked by the partner; refer to section 2.4.3 for more information.

Each instance of a substance published by the SRS data flow's data publishing services conforms to a common structure, regardless of the substance type (i.e., Chemical, Biological, etc.), in the XML document that is published as a result of the service request. Figure 5-1 below depicts this structure.

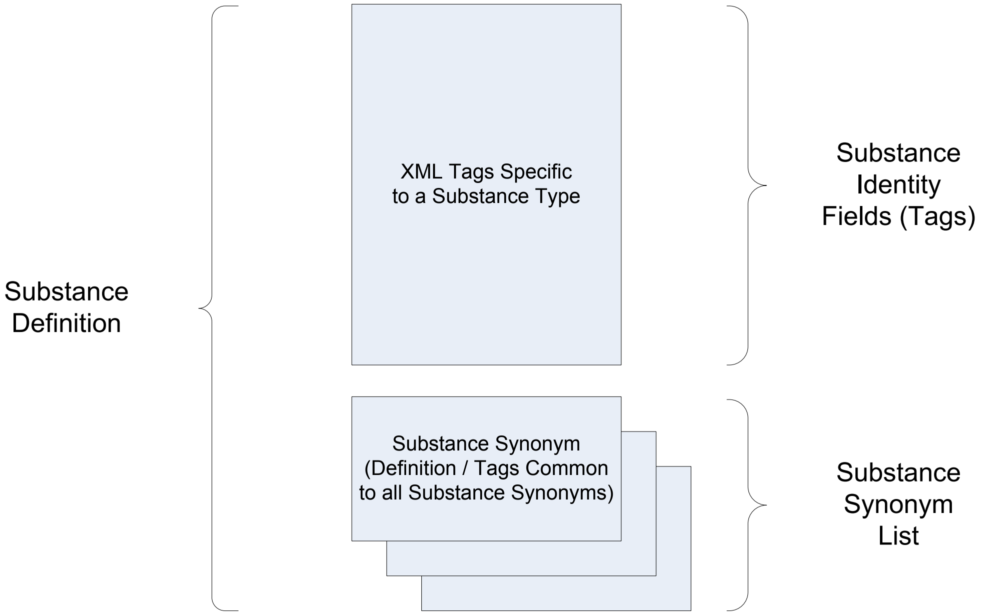


Figure 5-1 SRS Common Substance XML Structure

In the resulting XML document, each substance is contained within an outer "substance" tag (the name of the tag depends on the substance type). Within the substance tag, there is one and only one substance identity XML structure, and a list of zero or more substance synonym XML structures. See section 7.2.2 for more information. An example of this common XML structure is depicted in Figure 5-2.

```
<!-- Chemical Substance -->
<ChemicalSubstance>

      <!-- Identity -->
      <ChemicalSubstanceIdentification>
            <identity-field1>…</identity-field1>
            <identity-field2>…</identity-field2>
            Etc...
      </ChemicalSubstanceIdentification>

      <!-- Synonyms -->
      <ChemicalSubstanceSynonymList>

            <!-- Synonym 1 -->
            <ChemicalSubstanceSynonym>
                  <synonym-field1>…</synonym-field1>
                  <synonym-field2>…</synonym-field2>
                  Etc...
            </ChemicalSubstanceSynonym>

            <!-- Synonym 2 -->
            <ChemicalSubstanceSynonym>
                  <synonym-field1>…</synonym-field1>
                  <synonym-field2>…</synonym-field2>
                  Etc...
            </ChemicalSubstanceSynonym>

            Etc... (more synonyms)

      </ChemicalSubstanceSynonymList>

</ChemicalSubstance>
```

Figure 5-2 SRS Example Substance XML Structure

Please note: XML comments such as "`<!-- Identity -->`" depicted in Figure 5-2 are NOT present in the actual documents published. They are included here for clarification only.

For chemical substances published by the SRS services, the Exchange Network's Shared Schema Component for chemical identification provides for the use of common identifiers for all chemical substances regulated or monitored by environmental programs. Version 2.0 of the Chemical Substance Identity definition is located at the following URL:

http://exchangenetwork.net/schema/SC/SC_ChemicalSubstanceIdentity_v2.0.xsd

```
<xsd:complexType name="ChemicalSubstanceIdentityDataType">
       <xsd:sequence>
              <xsd:element ref="EPAChemicalInternalNumber" minOccurs="0" />
              <xsd:element ref="CASRegistryNumber" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceSystematicName" minOccurs="0" />
              <xsd:element ref="EPAChemicalRegistryName" minOccurs="0" />
              <xsd:element ref="EPAChemicalIdentifier" minOccurs="0" />
              <xsd:element ref="ChemicalNameContextName" minOccurs="0" />
              <xsd:element ref="EPAChemicalRegistryNameSourceText" minOccurs="0" />
              <xsd:element ref="EPAChemicalRegistryNameContextName" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceDefinitionText" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceCommentText" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceSynonymName" minOccurs="0" />
              <xsd:element ref="MolecularFormulaCode" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceFormulaWeightQuantity" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceTypeName" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceLinearStructureCode" minOccurs="0" />
              <xsd:element ref="ChemicalStructureGraphicalDiagram" minOccurs="0" />
              <xsd:element ref="ChemicalSubstanceClassificationName" minOccurs="0" />
              <xsd:element ref="ChemicalSynonymStatusName" minOccurs="0" />
              <xsd:element ref="ChemicalSynonymSourceName" minOccurs="0" />
       </xsd:sequence>
</xsd:complexType>
```

Figure 5-3 Chemical Substance Identity Shared Schema Component v2.0

Figure 5-3 depicts the definition of the shared schema component for chemical identities. This XML data type (data structure) is used for publishing chemical substance identifiers; it is tailored through standard XML schema extension mechanisms as needed to function with SRS.

For biological substances published by the SRS services, the Exchange Network's Shared Schema Component for biological identification provides for the use of common identifiers for all biological substances regulated or monitored by environmental programs. Version 2.0 of the Biological Substance Identity definition is located at the following URL:

http://exchangenetwork.net/schema/SC/SC_BiologicalSubstanceIdentity_v2.0.xsd

```
<xsd:complexType name="BiologicalSubstanceIdentityDataType">
    <xsd:sequence>
        <xsd:element minOccurs="0" ref="ITISTaxonomicSerialNumber" />
        <xsd:element minOccurs="0" ref="ICTVdBTaxonIdentifier" />
        <xsd:element minOccurs="0" ref="EPABiologicalIdentifier" />
        <xsd:element minOccurs="0" ref="BiologicalSystematicName" />
        <xsd:element minOccurs="0" ref="BiologicalVernacularName" />
        <xsd:element minOccurs="0" ref="BiologicalGroupName" />
        <xsd:element minOccurs="0" ref="BiologicalKingdomName" />
        <xsd:element minOccurs="0" ref="BiologicalDivisionPhylumName" />
        <xsd:element minOccurs="0" ref="BiologicalSubdivisionSubphylumName" />
        <xsd:element minOccurs="0" ref="BiologicalClassName" />
        <xsd:element minOccurs="0" ref="BiologicalSubclassName" />
        <xsd:element minOccurs="0" ref="BiologicalInfraclassName" />
        <xsd:element minOccurs="0" ref="BiologicalOrderName" />
        <xsd:element minOccurs="0" ref="BiologicalSuborderName" />
        <xsd:element minOccurs="0" ref="BiologicalFamilyName" />
        <xsd:element minOccurs="0" ref="BiologicalSubfamilyName" />
        <xsd:element minOccurs="0" ref="BiologicalGenusName" />
        <xsd:element minOccurs="0" ref="BiologicalSpeciesName" />
        <xsd:element minOccurs="0" ref="BiologicalSubspeciesName" />
        <xsd:element minOccurs="0" ref="BiologicalSynonymName" />
        <xsd:element minOccurs="0" ref="ProgramSystemBiologicalName" />
        <xsd:element minOccurs="0" ref="BiologicalGroupDescriptionText" />
        <xsd:element minOccurs="0" ref="ProgramSystemAbbreviatedName" />
        <xsd:element minOccurs="0" ref="ProgramSystemName" />
        <xsd:element minOccurs="0" ref="CitationReferenceCode" />
    </xsd:sequence>
</xsd:complexType>
```

Figure 5-4 Biological Substance Identity Shared Schema Component v2.0

Figure 5-4 depicts the definition of the shared schema component for biological identities. This XML data type (data structure) is used for publishing biological substance identifiers; it is tailored through standard XML schema extension mechanisms as needed to function with SRS.

The XML schema provided with the SRS data flow defines the substance identity XML structures for the remaining substances types published from SRS (i.e., physical property, miscellaneous, and not known substance types).

5.1 Substance Retrieval Data Publishing (Query)

5.1.1 GetSubstanceByALTIDListAcronym

This data service allows partners to select substances based upon an alternate identifier for the substance and whether the substance is a member of a specified list within SRS. Based upon qualification made to the alternate ID parameter (see notes below), zero or more substances may be returned by this service.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.1.1 Request

Dataflow: SRS

Request: GetSubstanceByALTIDListAcronym

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`AlternateIdentifier`	String	Yes	50	See 5.3.1	1
`ListAcronym`	String	Yes	100	See 5.3.1	2
`AlternateIdentifierQualification`	String	Yes	8	See 5.3.3	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.1.1.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.2 GetSubstanceByCAS

This data service allows partners to select substances based upon the CAS registry number associated with the substance in SRS. This service will select substances that match either the substance's current CAS registry number, or one of the CAS registry numbers in its lists of incorrectly used or formerly used CAS registry numbers.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.2.1 Request

Dataflow: SRS

Request: GetSubstanceByCAS

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`CASRegistryNumber`	String	Yes	15	See 5.3.2	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.1.2.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.3 GetSubstanceByEPAID

This data service allows partners to select substances based upon the EPA identifier associated with the substance in SRS. The selection is based upon an exact match of the specified parameter.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.3.1 Request

Dataflow: SRS

Request: GetSubstanceByEPAID

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`EPAIdentifier`	String	Yes	10	See 5.3.1	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.1.3.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.4 GetSubstanceByInternalTrackingNumber

This data service allows partners to select substances based upon the SRS internal tracking number associated with the substance in SRS. The selection is based upon an exact match of the specified parameter.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.4.1 Request

Dataflow: SRS

Request: GetSubstanceByInternalTrackingNumber

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`InternalTrackingNumber`	String	Yes	9	See 5.3.2	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.1.4.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.5 GetSubstanceByName

This data service allows partners to select substances based upon a substance name for the substance. The substance name will match based upon a substance's name, systematic name, or one of its synonym names. Based upon qualification made to the substance name parameter (see notes below), zero or more substances may be returned by this service.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.5.1 Request

Dataflow: SRS

Request: GetSubstanceByName

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`SubstanceName`	String	Yes	4000	See 5.3.1	1
`SubstanceNameQualification`	String	Yes	8	See 5.3.3	2
`IncludeList`	XML	Yes	4000	See 5.3.4.1	3
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	4
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	6

5.1.5.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.6 GetSubstanceByTSN

This data service allows partners to select substances based upon the taxonomic serial number associated with the substance in SRS. The selection is based upon an exact match of the specified parameter.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.6.1 Request

Dataflow: SRS

Request: GetSubstanceByTSN

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`TaxonomicSerialNumber`	String	Yes	12	See 5.3.2	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.1.6.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2).

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.1.7 GetSubstanceMatchByName

This data service allows partners to determine whether a substance exists within SRS based upon a substance name for the substance. The substance name will match based upon a substance's name, systematic name, or one of its synonym names. Based upon qualification made to the substance name parameter (see notes below), zero or more substances may be selected by this service. For this service, substance definitions are not returned; in this case, an indicator is returned in the XML: `<Match>Found</Match>` or `<Match>Not Found</Match>` indicating whether the query selected (matched) any substances in SRS.

Type: Query

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.1.7.1 Request

Dataflow: SRS

Request: GetSubstanceMatchByName

RowID: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one substance returned by the query (one instance of a substance XML tag and all of its child tags; see section 7.2.2). This parameter is the starting row number to return for this invocation of the query service.

maxRows: Any valid maxRows value per the Node specification WSDL (e.g., -1, or positive integer). For this query, this is the maximum number of substances (rows) to be returned for this invocation of the query service.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`SubstanceName`	String	Yes	4000	See 5.3.1	1
`SubstanceNameQualification`	String	Yes	8	See 5.3.3	2
`IncludeList`	XML	Yes	4000	See 5.3.4.1	3
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	4
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	6

5.1.7.2 Response

Response: See schema SRS_SubstanceInformation_v3.0 (XML)

RowID: The integer representing the position of the first substance returned in the full result set of the query. See the Node 2.0 specification for more information. For this query, each row represents a match indicator for each substance selected by the query.

RowCount: The integer representing the total number of substances returned for this query. This may not be all of the substances selected for this query based upon parameters passed to the query; the total number returned may be limited by the maxRows input parameter, or by the SRS application itself. Partners can continue to issue the query, increasing the rowID input parameter each time to retrieve all substances selected by the query. See the Node 2.0 specification for more information.

LastSet: Boolean indicating whether there are any more rows (substances) to return.

5.2 Substance Retrieval Data Publishing (Solicit)

5.2.1 GetSubstancesByCAS

This asynchronous data service allows partners to select substances based upon the CAS registry number associated with the substance in SRS. This service takes a list of CAS registry numbers and will select substances that match any of the specified CAS registry numbers where; a match is based upon either the substance's current CAS registry number, or one of the CAS registry numbers in its lists of incorrectly used or formerly used CAS registry numbers.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.1.1 Request

Dataflow: SRS

Request: GetSubstancesByCAS

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
CASRegistryNumberList	XML	Yes	4000	See 5.3.4.3	1
IncludeList	XML	Yes	4000	See 5.3.4.1	2
ExcludeList	XML	Yes	4000	See 5.3.4.1	3
IncludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	4
ExcludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	5

5.2.1.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is 'Completed'. The resulting document to download will be named 'solicitResponse.xml' and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.2 GetSubstancesByCategoryClassifications

This asynchronous data service allows partners to select substances based upon the classifications associated with the substance in SRS. This service takes a list of classifications and will select substances that have been classified with SRS as any of the specified classifications.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.2.1 Request

Dataflow: SRS

Request: GetSubstancesByCategoryClassifications

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`ClassificationList`	XML	Yes	4000	See 5.3.4.3	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.2.2.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.3 GetSubstancesByCharacteristics

This asynchronous data service allows partners to select substances based upon the characteristics associated with the substance in SRS. This service takes a list of characteristics and will select substances that have been specified with SRS as having any of the specified characteristics.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.3.1 Request

Dataflow: SRS

Request: GetSubstancesByCharacteristics

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`CharacteristicList`	XML	Yes	4000	See 5.3.4.3	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.2.3.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.4 GetSubstancesByEPAIDs

This asynchronous data service allows partners to select substances based upon the EPA identifier associated with the substance in SRS. This service takes a list of EPA identifiers and will select substances that match any of the specified EPA identifiers in SRS.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.4.1 Request

Dataflow: SRS

Request: GetSubstancesByEPAIDs

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
EPAIdentifierList	XML	Yes	4000	See 5.3.4.3	1
IncludeList	XML	Yes	4000	See 5.3.4.1	2
ExcludeList	XML	Yes	4000	See 5.3.4.1	3
IncludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	4
ExcludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	5

5.2.4.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.5 GetSubstancesByInternalTrackingNumber

This asynchronous data service allows partners to select substances based upon the SRS internal tracking number associated with the substance in SRS. This service takes a list of internal tracking numbers and will select substances that match any of the specified internal tracking numbers in SRS.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.5.1 Request

Dataflow: SRS

Request: GetSubstancesByInternalTrackingNumber

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`InternalTrackingNumberList`	XML	Yes	4000	See 5.3.4.3	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.2.5.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.6 GetSubstancesByListAcronym

This asynchronous data service allows partners to select all substances that are a member of the specified list in SRS. This service takes a list acronym to match within SRS.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.6.1 Request

Dataflow: SRS

Request: GetSubstancesByListAcronym

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
ListAcronym	String	Yes	100	See 5.3.1	1
IncludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	2
ExcludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	3

5.2.6.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.7 GetSubstancesByNamesExact

This asynchronous data service allows partners to select substances based upon a substance name for the substance. The substance name will match based upon an exact match of the substance's name, systematic name, or one of its synonym names.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.7.1 Request

Dataflow: SRS

Request: GetSubstancesByNamesExact

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
`SubstanceNameList`	XML	Yes	4000	See 5.3.4.3	1
`IncludeList`	XML	Yes	4000	See 5.3.4.1	2
`ExcludeList`	XML	Yes	4000	See 5.3.4.1	3
`IncludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	4
`ExcludeSubstanceTypeList`	XML	Yes	4000	See 5.3.4.2	5

5.2.7.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is 'Completed'. The resulting document to download will be named 'solicitResponse.xml' and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be 'Failed'; if failed, the partner can download a processing report using the returned transaction ID and the document name 'Node20.Report'.

5.2.8 GetSubstanceCASByListAcronym

This asynchronous data service allows partners to select only the CAS registry numbers of all substances that are a member of the specified list in SRS. This service takes a list acronym to match within SRS. This service is equivalent to the GetSubstancsByListAcronym except that it only returns CAS registry numbers instead of full substance definitions.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.8.1 Request

Dataflow: SRS

Request: GetSubstanceCASByListAcronym

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
ListAcronym	String	Yes	100	See 5.3.1	1
IncludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	2
ExcludeSubstanceTypeList	XML	Yes	4000	See 5.3.4.2	3

5.2.8.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0. In this case, only a list of <SRS:CASRegistryNumber> tags will be returned in the <SRS:CASRegistryNumberList> tag of the <SRS:SubstanceInformation> tag.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.9 GetSubstanceListAcronymsByAcronym

This asynchronous data service allows partners to determine substance list acronyms that exist in SRS. A list of zero or more SRS substance list acronyms will be returned based upon the specified acronym parameter value and its qualification.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.9.1 Request

Dataflow: SRS

Request: GetSubstanceListAcronymsByAcronym

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
ListAcronym	String	Yes	100	See 5.3.1	1
ListAcronymQualification	String	Yes	8	See 5.3.3	2

5.2.9.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_ListMaintenance_v3.0. In this case, only a list of <SRS:ListAcronym> tags will be returned in the <SRS:ListMaintenance> tag.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.2.10 GetPMNACCInformation

This asynchronous data service allows partners to select substances that are on the CBI (Confidential Business Information) section of the TSCA Inventory based upon various public information that the TSCA Inventory publishes for these substances. This information includes fields such as PMN Number, Accession Number, and Generic Name. This service takes a list of criteria and will select substances that match all of the specified criteria.

Type: Solicit

Data Service-level Business Rules: Not applicable.

XML Header Usage: Not applicable.

5.2.10.1 Request

Dataflow: SRS

Request: GetPMNACCInformation

Recipient Addresses: Not currently supported.

Notification URI: Not currently supported.

Parameters:

Name	Data Type	Required	Max Length	Notes	Position *(Node 1.1 Only)*
GenericName	String	No	4000	See 5.3.1	1
GenericNameQualification	String	No	8	See 5.3.3	2
Identifier	String	No	4000	See 5.3.1	3
IdentifierQualification	String	No	8	See 5.3.3	4
IdentifierTypeList	XML	No	4000	See 5.3.4.3	5
EPAInventoryFlagList	XML	No	4000	See 5.3.4.3	6

5.2.10.2 Response

Response: A transaction ID (String). The returned transaction ID can be used to determine the processing status of the Solicit request using the standard GetStatus node service, and to download the resulting XML document when the transaction status returned is ‘Completed’. The resulting document to download will be named ‘solicitResponse.xml’ and will be an XML document that conforms to the SRS schema SRS_SubstanceInformation_v3.0.

If an error occurs during processing, the transaction status will be ‘Failed’; if failed, the partner can download a processing report using the returned transaction ID and the document name ‘Node20.Report’.

5.3 Data Publishing Service Parameters

Because many of the parameters defined for the SRS data publishing services are used on multiple services and/or follow the same validation or business rules, the definition and rules for the these parameters are provided in a single place in this section.

5.3.1 String Parameters – Alphanumeric

These parameters are string values (Node 2.0 “None” encoding) that can contain any valid alphanumeric characters. Wild-card characters are NOT recognized; for example, typical wild-card characters such as “*” (asterisk) are not recognized as wild-card characters but are treated as simple characters. Some services accept value qualification parameters (see section 5.3.3). In these cases, the value of the corresponding string parameter will be used as a partial value; otherwise, the value of this parameter is used as an exact match value.

5.3.2 String Parameters – Numeric

These parameters are string values (Node 2.0 “None” encoding) that can contain any valid alphanumeric characters; however, all non-numeric characters are removed from the parameter during processing, and the resulting value **must** be a numeric integer value. This allows values for parameters such as the CASRegistryNumber parameter to be input either with its customary dashes (‘-‘) in the appropriate place, or without them. Like alphanumeric string parameters, wild-card characters are NOT recognized.

5.3.3 Parameter-Value Qualification Parameters

Several SRS data publishing services accept parameters that qualify the value provided in another parameter (of the same name without the “Qualification” suffix). For example, the service GetSubstanceByName accepts parameters named SubstanceName and SubstanceNameQualification.

The valid values for “qualification” parameters are: “exact”, “begins”, or “contains”. One of those values must be supplied. The meaning of each value is below. If the “qualification” parameter for a particular service is optional, and the parameter is not supplied, the default “qualification” will be exact.

Qualification Parameter Value	Effect
`exact`	The value specified in the corresponding parameter is used as-is (exact matching) during processing of the data service request
`begins`	The value specified in the corresponding parameter is used to match any value that “begins with” the value specified in the corresponding parameter during processing of the data service request
`contains`	The value specified in the corresponding parameter is used to match any value that “contains” the value specified in the corresponding parameter during processing of the data service request

5.3.4 XML Parameters

The Node 2.0 specification supports parameters that are XML-encoded (i.e., they are snippets of well-formed XML or are full-fledged XML documents). The SRS data flow provides a number of data publishing services that make use of XML parameters to allow variable lists of values to be passed to the service in a semantically well-defined, industry standard manner (i.e., as XML). For Node 1.1 clients, these XML parameters can be passed to the CDX node in the standard 1.1 parameter array of strings as a string containing XML; however, it is strongly encouraged that partners utilize Node 2.0 client technology.

In all cases, if it is desired that an XML parameter value have no effect on a data service request, an “empty” XML string should be passed as the XML parameter value, consisting of an open and close root XML tag only. See the following sections for more information.

5.3.4.1 Including and Excluding Lists

Many of the SRS data services provide the ability to search for substances based upon their inclusion or exclusion from specific lists within SRS. In these cases, the parameter names are: `IncludeList` and `ExcludeList`. The format of the XML to use for these parameter values is:

XML Parameter Name	Parameter Value XML Format

`IncludeList`	The format of the XML passed as a parameter value should be:
	`<IncludeList>`
	`<List>acronym1</List>`
	`<List>acronym2</List>`
	…
	`</IncludeList>`
	If this parameter is passed, only substances that are members of the specified lists will be published in response to the service request.

`ExcludeList`	The format of the XML passed as a parameter value should be:
	`<ExcludeList>`
	`<List>acronym1</List>`
	`<List>acronym2</List>`
	…
	`</ExcludeList>`
	If this parameter is passed, substances that are members of the specified lists will NOT be published in response to the service request.

5.3.4.2 Including and Excluding Substance Types

Many of the SRS data services provide the ability to search for substances based upon the type of substance (e.g., Chemical, Biological, Physical Property, etc.). In these cases, the parameter names are: `IncludeSubstanceTypeList` and `IncludeSubstanceTypeList`. The format of the XML to use for these parameter values is:

XML Parameter Name	Parameter Value XML Format

`IncludeSubstanceTypeList`	The format of the XML passed as a parameter value should be:
	`<IncludeSubstanceTypeList>`
	`  <SubstanceType>type1</SubstanceType>`
	`  <SubstanceType>type2</SubstanceType>`
	`  …`
	`</IncludeSubstanceTypeList>`
	If this parameter is passed, only substances that are of the specified type will be published in response to the service request. Valid values for the SubstanceType parameter are: Chemical Substance, Biological Organism, Physical Property, Miscellaneous Object, Not Known

`ExcludeSubstanceTypeList`	The format of the XML passed as a parameter value should be:
	`<ExcludeSubstanceTypeList>`
	`  <SubstanceType>type1</SubstanceType>`
	`  <SubstanceType>type2</SubstanceType>`
	`  …`
	`</ExcludeSubstanceTypeList>`
	If this parameter is passed, substances that are of the specified type will NOT be published in response to the service request. Valid values for the SubstanceType parameter are: Chemical Substance, Biological Organism, Physical Property, Miscellaneous Object, Not Known

5.3.4.3 Value-List XML Parameters

Where used, XML parameters to the SRS data services are designed to allow lists of values to be supplied to the data service in a semantically well-defined way (i.e., as XML). This section documents the XML parameters used by the SRS data services that are not defined in the previous sections.

XML Parameter Name	Parameter Value XML Format

`CASRegistryNumberList`	The format of the XML passed as a parameter value should be:
	`<CASRegistryNumberList>`
	`  <CASRegistryNumber>num1</CASRegistryNumber>`
	`  <CASRegistryNumber>num2</CASRegistryNumber>`
	`  …`
	`</CASRegistryNumberList>`
	Only substances that match the specified CAS registry numbers will be published in response to the service request.

`CharacteristicList`	The format of the XML passed as a parameter value should be:
	`<CharacteristicList>`
	`  <Characteristic>char1</Characteristic>`
	`  <Characteristic>char2</Characteristic>`
	`  …`
	`</CharacteristicList>`
	Only substances that match have the specified characteristics in SRS will be published in response to the service request.

`ClassificationList`	The format of the XML passed as a parameter value should be:
	`<ClassificationList>`
	`  <Classification>class1</Classification>`
	`  <Classification>class2</Classification>`
	`  …`
	`</ClassificationList>`
	Only substances that are classified as one of the provided classifications in SRS will be published in response to the service request.

`EPAIdentifierList`	The format of the XML passed as a parameter value should be:
	`<EPAIdentifierList>`
	`  <EPAIdentifier>id1</EPAIdentifier>`
	`  <EPAIdentifier>id2</EPAIdentifier>`
	`  …`
	`</EPAIdentifierList>`
	Only substances that match (have) the specified EPA identifiers in SRS will be published in response to the service request.

XML Parameter Name	Parameter Value XML Format

`InternalTrackingNumberList`	The format of the XML passed as a parameter value should be:
	`<InternalTrackingNumberList>`
	` <InternalTrackingNumber>id1</InternalTrackingNumber>`
	` <InternalTrackingNumber>id2</InternalTrackingNumber>`
	 …
	`</InternalTrackingNumberList>`
	Only substances that match (have) the specified SRS internal tracking numbers in SRS will be published in response to the service request.

`IdentifierTypeList`	The format of the XML passed as a parameter value should be:
	`  <IdentifierTypeList>`
	`    <IdentifierType>type1</IdentifierType>`
	`    <IdentifierType>type2</IdentifierType>`
	    …
	`  </IdentifierTypeList>`
	For substances on the CBI section of the TSCA Inventory, when using the Identifier parameter, only identifiers that belong to one of the specified types will be searched when publishing the response to the service request. Valid values for the IdentifierType parameter are: PMN Number, Accession Number

`EPAInventoryFlagList`	The format of the XML passed as a parameter value should be:
	`  <EPAInventoryFlagList>`
	`    <EPAInventoryFlag>flag1</EPAInventoryFlag>`
	`    <EPAInventoryFlag>flag2</EPAInventoryFlag>`
	    …
	`  </EPAInventoryFlagList>`
	For substances on the CBI section of the TSCA Inventory, only substances that match (have) one of the specified EPA Inventory Flags will be published in response to the service request. Valid values for the EPAInventoryFlag parameter are:
	• TSCA 5(e) Consent Orders
	• TSCA 5(f) Specific Labeling
	• TSCA Free-Radical Initiated Polymers
	• TSCA Commenced PMN
	• TSCA 6 Unreasonable Risk
	• TSCA 5A SNUR
	• TSCA 4 Tests
	• TSCA IUR Exempt
	• TSCA Exempt Polymers with NAMW &gt;= 1,000
	• TSCA Exempt Polyester Polymers

6 Using the REST Web Services

6.1 Creating REST URLs

REST web services are available for the query data services appearing in section 5.1. They are not available for solicit or publishing data services. The REST services are exposed through EPA's CDX Exchange Network REST Proxy (ENRP). The URL template for SRS REST services as generated by ENRP is as follows.

```
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request={r
equest}&Params={parameters}
```

The URL parameters are detailed in Exhibit 6-1 below.

Exhibit 6-1 URL Parameter Description

URL Parameter	Description
request	The name of the service request (e.g., GetSubstanceByName).
parameters	A list of all parameters for the Query service (described in 3.1.1 and 3.2.1) in the form of: param1|value1;param2|value2;... where parameter name/value pairs are separated with a pipe (|) delimiter and parameters are separated with a semi-colon (;) delimiter. XML-format service parameters may be provided in plaintext. This parameter may be omitted if no query data service parameters need be supplied.

Example REST URLs are provided in section 6.2.

6.2 Example SRS REST Web Service URLs

Below is a table of example URLs that would be used to call the REST service and the results returned by each.

Exhibit 6-2 Example Tribal Web Service URLs

REST Web Service URL	Results
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request=GetSubstanceByName&Params=SubstanceName|Methylolurea	Get Substance(s) with exact name “Methulolurea”.
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request=GetSubstanceByName&Params=SubstanceName|Meth;SubstanceNameQualification|begins	Get Substance(s) with name that begins with “Meth”.
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request=GetSubstanceByName&Params=SubstanceName|Meth;SubstanceNameQualification|begins;ExcludeSubstanceTypeList|<ExcludeSubstanceTypeList><SubstanceType>Chemical Substance</SubstanceType><SubstanceType></SubstanceType></ExcludeSubstanceTypeList>	Get Substance(s) with name that begins with “Meth”, excluding chemical substances.

REST Web Service URL	Results
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request=GetSubstanceByInternalTrackingNumber&Params=InternalTrackingNumber|164442	Get Substance with Internal Tracking Number of “164442”.
https://ends2.epacdxnode.net/RestProxy/query?Node=NGNTest2.0&Dataflow=SRS&Request=GetSubstanceByCAS&Params=CASRegistryNumber|1000-82-4-1	Get Substance with CAS Registry Number of “1000-82-4-1”.
https://ends2.epacdxnode.net/RestProxy/query?Node=NGNTest2.0&Dataflow=SRS&Request=GetSubstanceByEPAID&Params=EPAIdentifier|E1894187	Get Substance with EPA Identifier of “E1894187”.
https://ends2.epacdxnode.net/RestProxy/query?Node=NGNTest2.0&Dataflow=SRS&Request=GetSubstanceByTSN&Params=TaxonomicSerialNumber|94719	Get Substance with Taxonomical Serial Number of “92719”.
https://ends2.epa.gov/RestProxy/Query?Node=NGNProd2.0&Dataflow=SRS&Request=GetSubstanceMatchByName&Params=SubstanceName|Urea, N-(hydroxymethyl)-	Check whether substance with name “Urea, N-(hydroxymethyl)-“ exists in SRS.

7 Schema Information

7.1 Schema Structure

The Substance Registry Services (SRS) data flow XML schema is implemented using a modular architecture as defined in the Exchange Network XML Schema Design Rules and Conventions document version 1.1 (http://www.exchangenetwork.net/dev_schema/drc.htm). The SRS schema architecture is depicted in the following diagram (Figure 7-1):

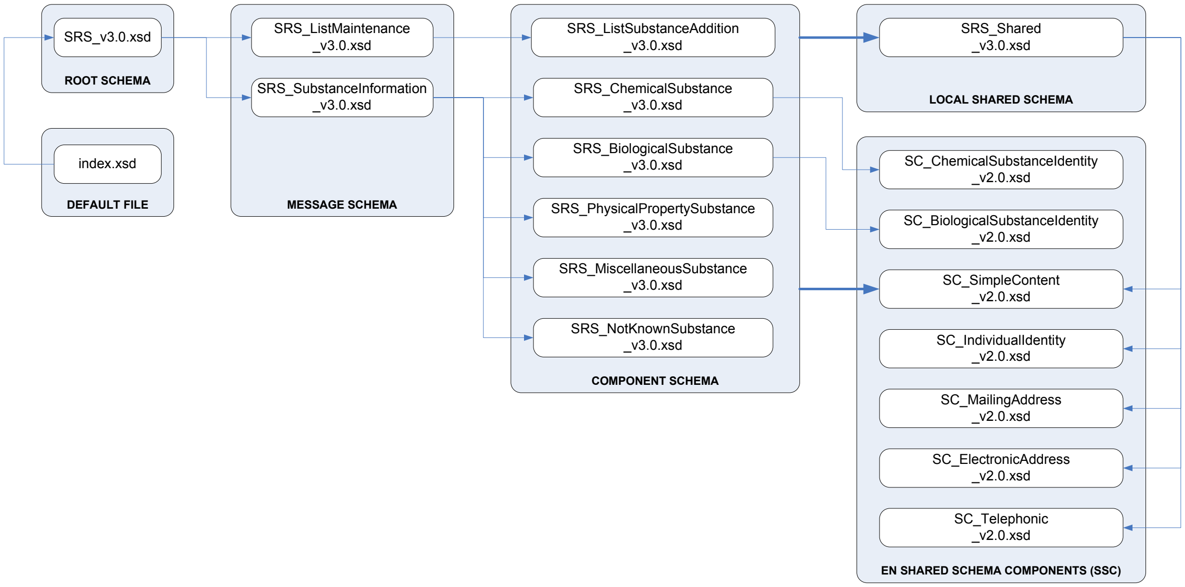

SRS_v3.0.xsd
ROOT SCHEMA
index.xsd
DEFAULT FILE
SRS_ListMaintenance _v3.0.xsd
SRS_SubstanceInformation _v3.0.xsd
MESSAGE SCHEMA
SRS_ListSubstanceAddition _v3.0.xsd
SRS_ChemicalSubstance _v3.0.xsd
SRS_BiologicalSubstance _v3.0.xsd
SRS_PhysicalPropertySubstance _v3.0.xsd
SRS_MiscellaneousSubstance _v3.0.xsd
SRS_NotKnownSubstance _v3.0.xsd
COMPONENT SCHEMA
SRS_Shared _v3.0.xsd
LOCAL SHARED SCHEMA
SC_ChemicalSubstanceIdentity _v2.0.xsd
SC_BiologicalSubstanceIdentity _v2.0.xsd
SC_SimpleContent _v2.0.xsd
SC_IndividualIdentity _v2.0.xsd
SC_MailingAddress _v2.0.xsd
SC_ElectronicAddress _v2.0.xsd
SC_Telephonic _v2.0.xsd
EN SHARED SCHEMA COMPONENTS (SSC)

Figure 7-1 SRS Data Flow XML Schema Architecture

The SRS schema architecture makes use of a local shared schema that is specific to the SRS data services, as well as a number of the Exchange Network's Shared Schema Components. Use of the shared schema components helps ensure that data exchanged using the SRS data services conform to existing EPA and Exchange Network data standards. Please note: in Figure 7-1, all of the component schemas make use of the SRS local shared schema, as well as the Simple Content shared schema component; to simplify the diagram, a connection is only shown between the outer "Component Schema" box and the two shared components used by all of the component schemas.

Version 3.0 of the SRS XML schema can be located at the following URL:

http://exchangenetwork.net/schema/SRS/3

7.2 Schema Components

The SRS XML schema is organized around the two categories of data services: data processing and data publishing. The data processing services provided by SRS are currently designed to support list maintenance within SRS. The data publishing services provided by SRS are designed to support substance information retrieval. The major components of the SRS schema are described in the following sections.

7.2.1 Data Processing Schema Components

The SRS data processing services all conform to make use of the Exchange Network's Header version 2.0 (http://www.exchangenetwork.net/dev_schema/HeaderSpecification_v2.0.doc). The Message (or payload) Schema contained in the Exchange Network's payload tag for SRS data processing services is the SRS_ListMaintenance_v3.0.xsd schema. The List Maintenance schema itself includes one or more List Substance Addition data blocks (defined by the SRS_ListSubstanceAddition_v3.0 schema). This allows one or more substance additions to be performed per document submission. Figure 7-2 depicts this architecture.

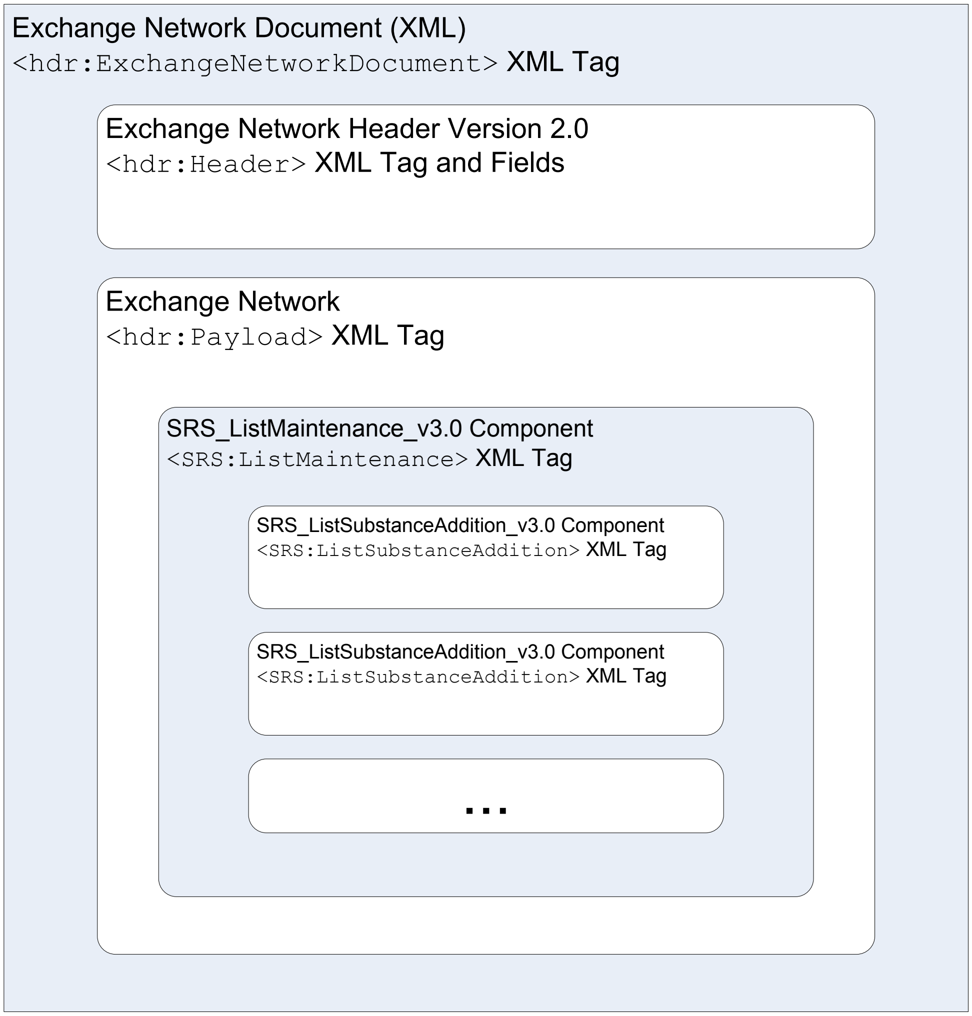

Exchange Network Document (XML)
<hdr:ExchangeNetworkDocument> XML Tag
Exchange Network Header Version 2.0
<hdr:Header> XML Tag and Fields
Exchange Network
<hdr:Payload> XML Tag
SRS_ListMaintenance_v3.0 Component
<SRS:ListMaintenance> XML Tag
SRS_ListSubstanceAddition_v3.0 Component
<SRS:ListSubstanceAddition> XML Tag
SRS_ListSubstanceAddition_v3.0 Component
<SRS:ListSubstanceAddition> XML Tag
...

Figure 7-2 SRS Data Processing Schema Components

7.2.2 Data Publishing Schema Components

The SRS data publishing services are designed primarily to publish substance information for one or more substances. The SRS data publishing services do not make use of the Exchange Network Header. The relationship of the major schema components used for data publishing is depicted in Figure 7-3.

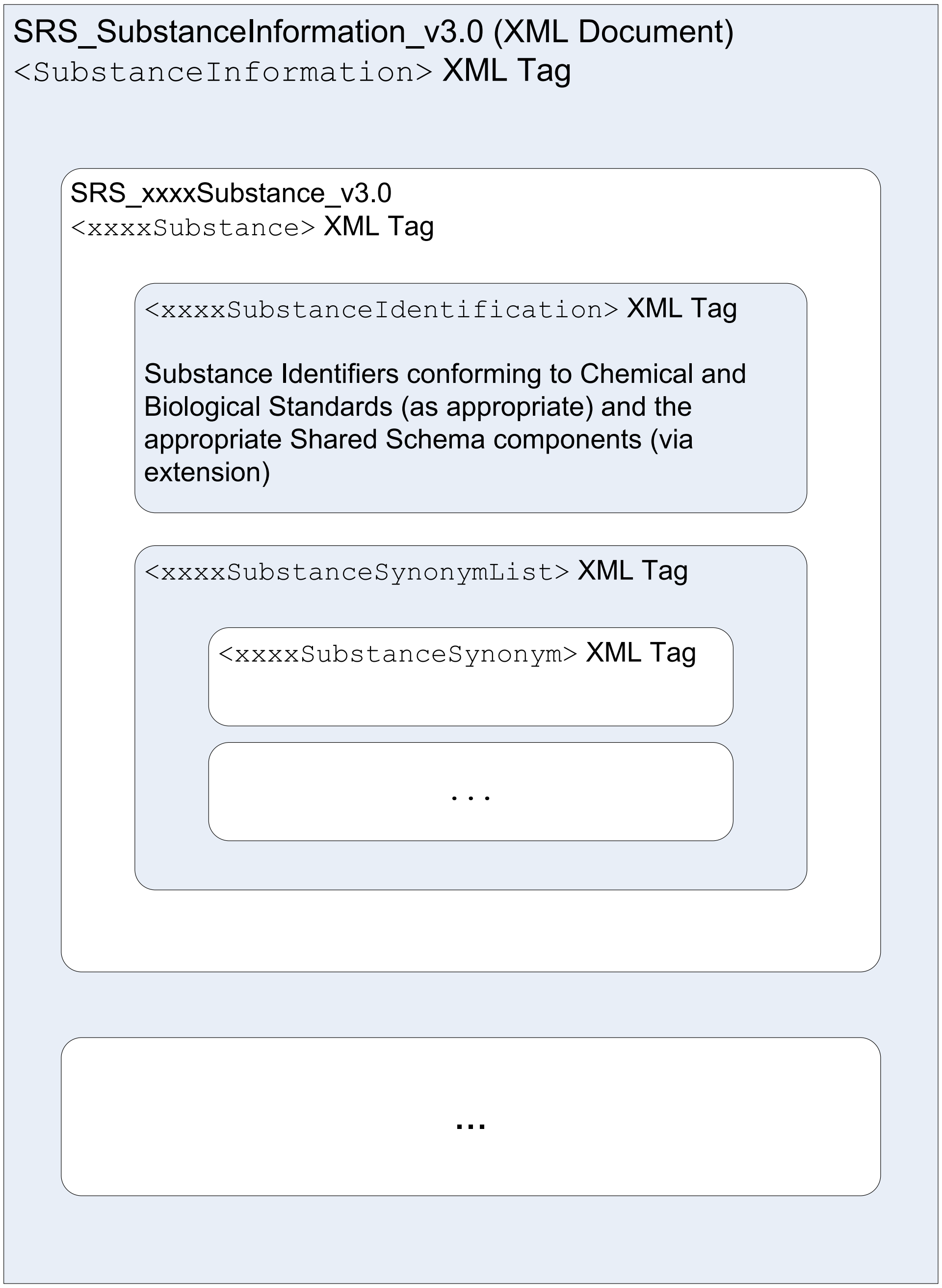

SRS_SubstanceInformation_v3.0 (XML Document)
<SubstanceInformation> XML Tag
SRS_xxxxSubstance_v3.0
<xxxxSubstance> XML Tag
<xxxxSubstanceIdentification> XML Tag
Substance Identifiers conforming to Chemical and Biological Standards (as appropriate) and the appropriate Shared Schema components (via extension)
<xxxxSubstanceSynonymList> XML Tag
<xxxxSubstanceSynonym> XML Tag
. . .
...

Figure 7-3 SRS Data Publishing Schema Components

Please note: in Figure 7-3, “xxxx” should be substituted with any one of the following values: Chemical, Biological, PhysicalProperty, Miscellaneous, or NotKnown. The XML document that results from a query or solicit data service can return zero, one or more of any of these substance types. Each substance type contains a substance identification data block, and one or more substance synonym data blocks.