

RCS Web Services Developer’s Guide

Environmental Protection Agency (EPA)

Delivery Order # GS00Q09BGD0022

Task Order # EP-G11H-00154

Project # TDD 10.09

Version 1.1

November 14, 2016

Table of Contents

List of Exhibits iv

Revision Log v

1 Introduction 1

1.1 Document Purpose 1

1.2 Project Overview 1

2 EPA Reusable Component Services Web Services Overview 2

2.1 Access and Security 3

3 Data Publishing 4

3.1 GetRCSResourcesByType\_v1.0 4

3.1.1 Request 4

3.1.2 Response 5

3.2 GetRCSResourceByIdentfier\_v1.0 5

3.2.1 Request 5

3.2.2 Response 6

3.3 GetRCSResourceByCategoryTag\_v1.0 7

3.3.1 Request 7

3.3.2 Response 7

3.4 GetRCSResourceByKeyword\_v1.0 8

3.4.1 Request 8

3.4.2 Response 8

4 Using the RCS Web Service 9

4.1 Creating REST URLs 9

4.2 Example RCS Web Service URLs 9

4.3 Example Usages 10

4.3.1 Search on an resource by the resource type name 10

4.4 Business Rules 10

4.5 Use Cases 10

4.5.1 Use Case 1: Get Resources from Resource Type Name 11

4.5.2 Use Case 2: Get a Resource from Resource Identifier 11

5 Appendix A: Schema Information 12

6 Appendix B: Sample SOAP Requests and Responses 17

List of Exhibits

[Exhibit 2‑1 RCS Query Data Service Processing 3](#_Toc420677963)

[Exhibit 3‑1 GetRCSResourcesByType\_v1.0 Query Parameter 5](#_Toc420677964)

[Exhibit 3‑2 GetRCSResourceByIdentifier\_v1.0 Query Parameter 6](#_Toc420677964)

[Exhibit 4‑1 URL Parameter Description 7](#_Toc420677965)

[Exhibit 4‑2 Example RCS Web Service URLs 8](#_Toc420677966)

[Exhibit 7‑1 GetRCSResourcesByType\_v1.0 Sample Request 1](#_Toc420677968)5

[Exhibit 7‑2 GetRCSResourcesByType\_v1.0 Sample Response 1](#_Toc420677969)5

[Exhibit 7‑3 GetRCSResourcesByType\_v1.0 Sample Request 1](#_Toc420677970)6

[Exhibit 7‑4 GetRCSResourcesByType\_v1.0 Sample Response 1](#_Toc420677971)6

[Exhibit 7‑5 GetRCSResourceByIdentifier\_v1.0 Sample Response 20](#_Toc420677971)

[Exhibit 7‑6 GetRCSResourceByIdentifier\_v1.0 Sample Response 20](#_Toc420677971)

Revision Log

| **Date** | **Version No.** | **Description** | **Author** | **Reviewer** | **Review Date** |
| --- | --- | --- | --- | --- | --- |
| 5/28/2015 | 0.01 | Initial Draft | G. Woodring | K. Garrett | 05/29/15 |
| 7/10/2015 | 1.0 | Update | T. Bizal |  |  |
| 8/20/2015 | 1.0 | Addition of service “GetRCSResourceByIdentifier\_v1.0” | T. Bizal |  |  |
| 11/14/2016 | 1.1 | Addition of services:  “GetRCSResourcesByCategoryTag\_v1.0”,  “GetRCSResourcesByKeyword\_v1.0” | T. Bizal |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# Introduction

## Document Purpose

This document serves as a guide for developers, both internal and external, for the Environmental Protection Agency (EPA) who may use EPA’s new Reusable Component Services (RCS) web services. This guide outlines the data available through the web service and presents several use cases to assist developers in making the best use of the RCS data.

## Project Overview

RCS provides a central point of access to a broad range of IT resources, components, and services used in various EPA and partner systems. It enables reuse of these resources, which helps reduce cost, quicken development, and produce higher quality systems and applications.

RCS allows users to catalog reusable information resources for EPA and its partners, load resource metadata, harvest metadata from other Agency registries and repositories, relate services and data assets to relevant systems and applications, and access a single interface through which users can search for all resources regardless of type and location.

By enabling and promoting reusability, RCS results in the development of applications that meet standards, ensure quality, reduce cost, and minimize development time.

# EPA Reusable Component Services 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. Web service queries return data requests in XML format only and do not display the data in a user interface. Returning data in standardized XML format allows independent software applications to receive the data in a consistent manner which can be adjusted to suite their respective needs. All services offered by EPA RCS Web Services are query web services.

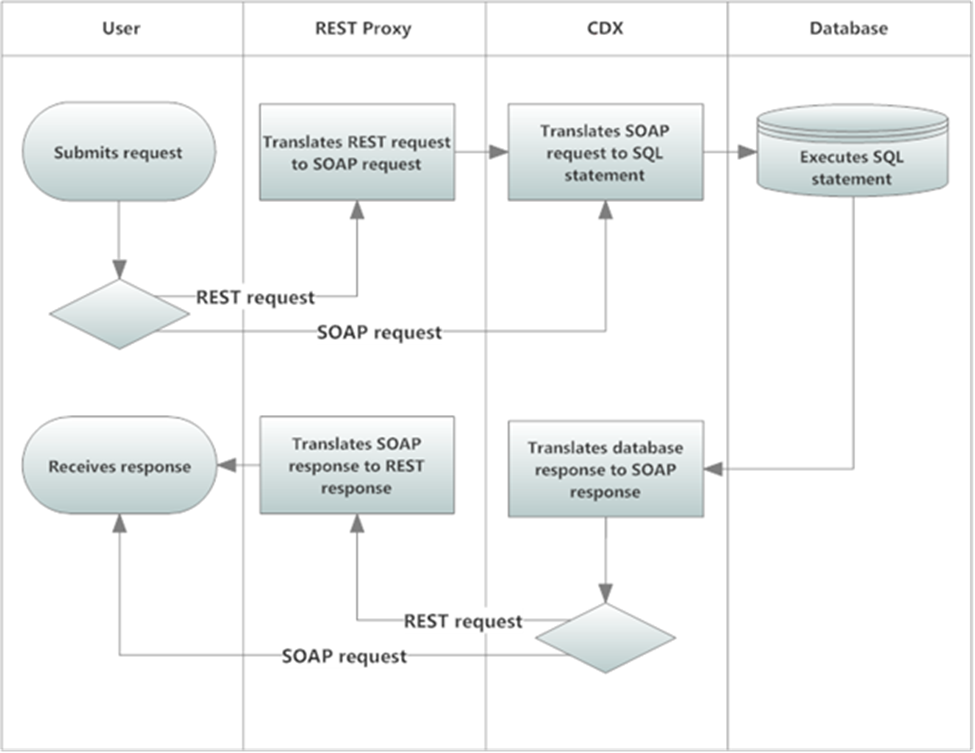
Web services generally run on either the Simple Object Access Protocol (SOAP) or Representational State Transfer (REST) protocol. The RCS Web Services support both SOAP and REST standards.

Use of a SOAP web service generally consists of XML message-passing between client and server. To invoke a typical web service with the SOAP protocol, the client sends an XML-formatted document that specifies the desired data service, necessary authentication credentials, and service-specific parameters. The service receives this request and responds with the requested data in XML.

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 a RCS Web Service is depicted in **Exhibit 2‑1** below.

Exhibit 2‑1 RCS Query Data Service Processing



## Access and Security

The REST services offered by EPA RCS Web Services are publicly available through the CDX REST Proxy and do not require any additional security authorizations to access.

All requests to SOAP services must be accompanied by a valid Network Authentication and Authorization Services (NAAS) security token per the [Exchange Network’s Node 2.0 specifications](http://www.exchangenetwork.net/node-2/) (<http://www.exchangenetwork.net/node-2/> ). All partners must be authorized to NAAS and receive a valid security token before the RCS data services can be invoked by a SOAP request.

In order to acquire a valid security token the user must have the appropriate NAAS security policies in place and associated with EPA’s CDX node and EPA RCS Web Services data flow. To obtain access, contact the CDX help desk for more information (<http://www.epa.gov/cdx/contact.htm>).

# Data Publishing

The RCS data flow offers two query data services at this time. The service GetRCSResourcesByType\_v1.0 allows users to list all metadata about resources registered in RCS by a given resource type. The types of resources available in RCS include: APIs, web services, XML schemas, blocks of code, code libraries, data models, software tools, and more. The metadata includes basic information such as: resource ID, resource name, description, contact individuals and organizations. It also includes information or links to documentation about access to resources, examples, and instructions for use. The service GetRCSResourceByIdentifier\_v1.0 allows users to list all metadata about a single resource in RCS by a given resource identifier. The sections below provide sample retrieval data publishing; the queries are designed to be performed in the order listed.

Please note that the parameters rowId and maxRows are currently accepted only when using SOAP query data services. These parameters are expected to be operational in the future, at which time this documentation will be updated accordingly.

## GetRCSResourcesByType\_v1.0

**Description:** This data service allows users to view a single or multiple resources based on their type. All associated metadata to the resource or resources will also be included. Zero or more codes may be returned by this service.

**Type:** Query

**Data Service-level Business Rules:** Not applicable

**XML Header Usage:** Not applicable

### Request

**Dataflow**: RCS

**Request**: GetRCSResourcesByType\_v1.0

**Endpoint:**

*Production -*https://node2.epa.gov/Node2WS.svc

*Test -*https://cdxnode2.epacdxnode.net/Node2WS.svc

**rowId**: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one resource returned by the query (one instance of a Resource XML tag and all of its child tags).

**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 resources (rows) to be returned for this invocation of the query service.

Exhibit 3‑1 GetRCSResourcesByType\_v1.0 Query Parameter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Required** | **Max Length** | **Occurrences** | **Wildcard Behavior** | **Notes** |
| ResourceTypeName | String | Yes | N/A | 1 | None | The name of the resource type. |

### Response

**Response:** See schema RCS\_Result\_v1.1.xsd in **Exhibit 7‑2** in Appendix B.

**rowId:** The integer representing the position of the first resource 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 Resource returned by the query.

**RowCount:** The integer representing the total number of resources returned for this query. This may not be all of the resources 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 RCS application itself. Users can continue to issue the query, increasing the rowID input parameter each time to retrieve all entities selected by the query. See the Node 2.0 specification for more information.

## GetRCSResourceByIdentifier\_v1.0

**Description:** This data service allows users to view a single resource from the provided resource identifier. All associated metadata to the resource will be included.

**Type:** Query

**Data Service-level Business Rules:** Not applicable

**XML Header Usage:** Not applicable

### Request

**Dataflow**: RCS

**Request**: GetRCSResourceByIdentifier\_v1.0

**Endpoint:**

*Production -*https://node2.epa.gov/Node2WS.svc

*Test -*https://cdxnode2.epacdxnode.net/Node2WS.svc

**rowId**: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one resource returned by the query (one instance of a Resource XML tag and all of its child tags).

**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 resources (rows) to be returned for this invocation of the query service.

Exhibit 3‑2 GetRCSResourcesByType\_v1.0 Query Parameter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Required** | **Max Length** | **Occurrences** | **Wildcard Behavior** | **Notes** |
| ResourceIdentifier | String | Yes | N/A | 1 | None | The resource identifier. |

### Response

**Response:** See schema RCS\_Result\_v1.1.xsd in **Exhibit 7‑6** in Appendix B.

**rowId:** The integer representing the position of the first resource 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 Resource returned by the query.

**RowCount:** The integer representing the total number of resources returned for this query. This may not be all of the resources 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 RCS application itself. Users can continue to issue the query, increasing the rowID input parameter each time to retrieve all entities selected by the query. See the Node 2.0 specification for more information.

## GetRCSResourceByCategoryTag\_v1.0

**Description:** This data service allows users to list all metadata about resources registered in RCS by a given category tag. All associated metadata to the resource will be included.

**Type:** Query

**Data Service-level Business Rules:** Not applicable

**XML Header Usage:** Not applicable

### Request

**Dataflow**: RCS

**Request**: GetRCSResourceByCategoryTag\_v1.0

**Endpoint:**

*Production -*https://node2.epa.gov/Node2WS.svc

*Test -*https://cdxnode2.epacdxnode.net/Node2WS.svc

**rowId**: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one resource returned by the query (one instance of a Resource XML tag and all of its child tags).

**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 resources (rows) to be returned for this invocation of the query service.

Exhibit 3‑2 GetRCSResourcesByType\_v1.0 Query Parameter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Required** | **Max Length** | **Occurrences** | **Wildcard Behavior** | **Notes** |
| CategoryTag | String | Yes | N/A | 1 | None | A specified category tag. |

### Response

**Response:** See schema RCS\_Result\_v1.1.xsd in **Exhibit 7‑6** in Appendix B.

**rowId:** The integer representing the position of the first resource 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 Resource returned by the query.

**RowCount:** The integer representing the total number of resources returned for this query. This may not be all of the resources 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 RCS application itself. Users can continue to issue the query, increasing the rowID input parameter each time to retrieve all entities selected by the query. See the Node 2.0 specification for more information.

## GetRCSResourceByKeyword\_v1.0

**Description:** This data service allows users to list all metadata about resources registered in RCS by a given keyword. All associated metadata to the resource will be included.

**Type:** Query

**Data Service-level Business Rules:** Not applicable

**XML Header Usage:** Not applicable

### Request

**Dataflow**: RCS

**Request**: GetRCSResourceByCategoryTag\_v1.0

**Endpoint:**

*Production -*https://node2.epa.gov/Node2WS.svc

*Test -*https://cdxnode2.epacdxnode.net/Node2WS.svc

**rowId**: Any valid rowID value per the Node specification WSDL (e.g., -1, 0, or positive integer). For this query, each row represents one resource returned by the query (one instance of a Resource XML tag and all of its child tags).

**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 resources (rows) to be returned for this invocation of the query service.

Exhibit 3‑2 GetRCSResourcesByType\_v1.0 Query Parameter

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Data Type** | **Required** | **Max Length** | **Occurrences** | **Wildcard Behavior** | **Notes** |
| Keyword | String | Yes | N/A | 1 | None | A specified keyword |

### Response

**Response:** See schema RCS\_Result\_v1.1.xsd in **Exhibit 7‑6** in Appendix B.

**rowId:** The integer representing the position of the first resource 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 Resource returned by the query.

**RowCount:** The integer representing the total number of resources returned for this query. This may not be all of the resources 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 RCS application itself. Users can continue to issue the query, increasing the rowID input parameter each time to retrieve all entities selected by the query. See the Node 2.0 specification for more information.

# Using the RCS Web Service

## Creating REST URLs

EPA RCS REST web services are exposed through EPA's CDX Exchange Network REST Proxy (ENRP). The URL template for EPA RCS REST services as generated by ENRP is as follows.

https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request=**{request}**&Params=**{parameters}**&rowId=**{rowId}**&MaxRows=**{maxRows}**

The URL parameters are detailed in **Exhibit 4‑1** below. One main point to remember when using this web service is that maxRows and rowId are parameters are only available when using the SOAP services and currently do not work within REST. The capitalization is also very important when using the parameters and must be copied identically from the way they appear in this document.

Exhibit 4‑1 URL Parameter Description

|  |  |
| --- | --- |
| URL Parameter | Description |
| request | The name of the service request: GetRCSResourcesByType\_v1.0, GetRCSResourceByIdentifier\_v1.0, GetRCSResourcesByCategoryTag\_v1.0, or GetRCSResourcesByKeyword\_v1.0 |
| 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. This parameter may be omitted if no Query service parameters need be supplied. |

Example REST URLs are provided in section 4.2.

## Example RCS 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 4‑2 Example RCS Web Service URLs

| **URL** | **WS REST Web Service URL** | **Results** |
| --- | --- | --- |
| 1 | https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request= GetRCSResourcesByType\_v1.0&Params=ResourceTypeName|database | List all resources of type Database |
| 2 | https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request= GetRCSResourcesByType\_v1.0&Params=ResourceTypeName|blocks of code | List all resources of type Blocks of Code |
| 3 | https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request= GetRCSResourceByIdentifier\_v1.0&Params=ResourceIdentifier|afbe711a-9e91-11e1-9e06-9da7dfd68bcf | List metadata for the resource with identifier “afbe711a-9e91-11e1-9e06-9da7dfd68bcf” |
| 4 | https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request=GetRCSResourcesByCategoryTag\_v1.0&Params=CategoryTag|Local%20Government | List all resources with Category Tag of “Local Government” |
| 5 | https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request=GetRCSResourcesByKeyword\_v1.0&Params=Keyword|cromerr | List all resources with keyword “cromerr” |

## Example Usages

Provided below are example searches that will provide guidance on how to use the RCS Web Service. These examples are of special cases that could potentially cause confusion when searching.

Please refer to Appendix B: Sample SOAP Requests and Responses for example SOAP XML responses.

### Search on an resource by the resource type

**Scenario:**

Looking for all SOAP Web Services registered in RCS.

**Result:**

Performing a search on this resource type using the resource type name of “SOAP Web Service”, the service will list all SOAP Web Service Resources.

**Explanation/Rule:**

The service will simply retrieve all publically available resources of the provided resource type in RCS.

## Business Rules

The following business rules and clarifications apply to the RCS data contained in EPA RCS Web Services:

* rowId represents the row number with which you would like the search results to begin. The default value is zero, which will return the very first matching result.
* maxRows represents the maximum number of rows to be returned from the search.

## Use Cases

EPA RCS Web Services data flow offers two query data services. The service ‘GetRCSResourcesByType\_v1.0’ allows users to get detailed information on a particular resource or resources based on a resource type parameter. The service ‘GetRCSResourceByIdentifier\_v1.0’ allows users to get detailed information on one particular resource based on its resource identifier.

Each of the use cases below are provided to assist the user of this Web Service with some of the most common searches. Provided are the REST URL that would be utilized, along with the expected inputs and outputs.

### Use Case 1: Get Resources from Resource Type Name

**URL # 1:** https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request= GetRCSResourcesByType\_v1.0&Params=ResourceTypeName|database

**Input:** The ResourceTypeName of ‘database’

**Output:** User receives a list of Database-type resources.

### Use Case 2: Get a Resource from Resource Identifier

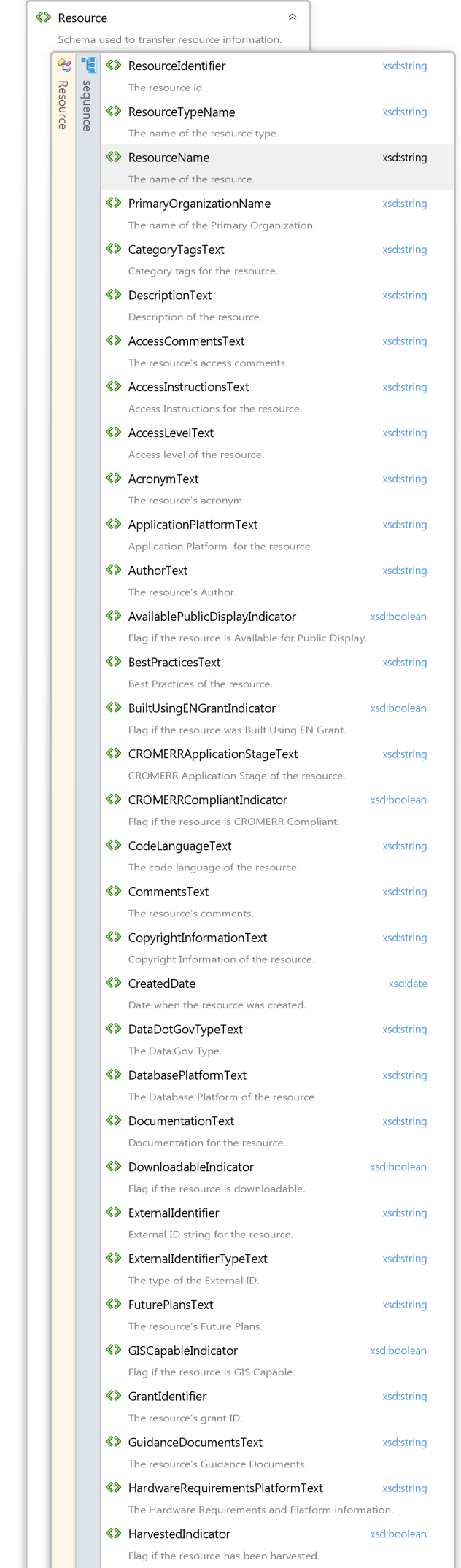
**URL # 1:** https://ends2.epa.gov/RestProxy/Query?Node=.NetNode2&Dataflow=RCS&Request= GetRCSResourceByIdentifier\_v1.0&Params=ResourceIdentifier|6bd8a515-dd80-11df-832e-92ce4b1f0c29

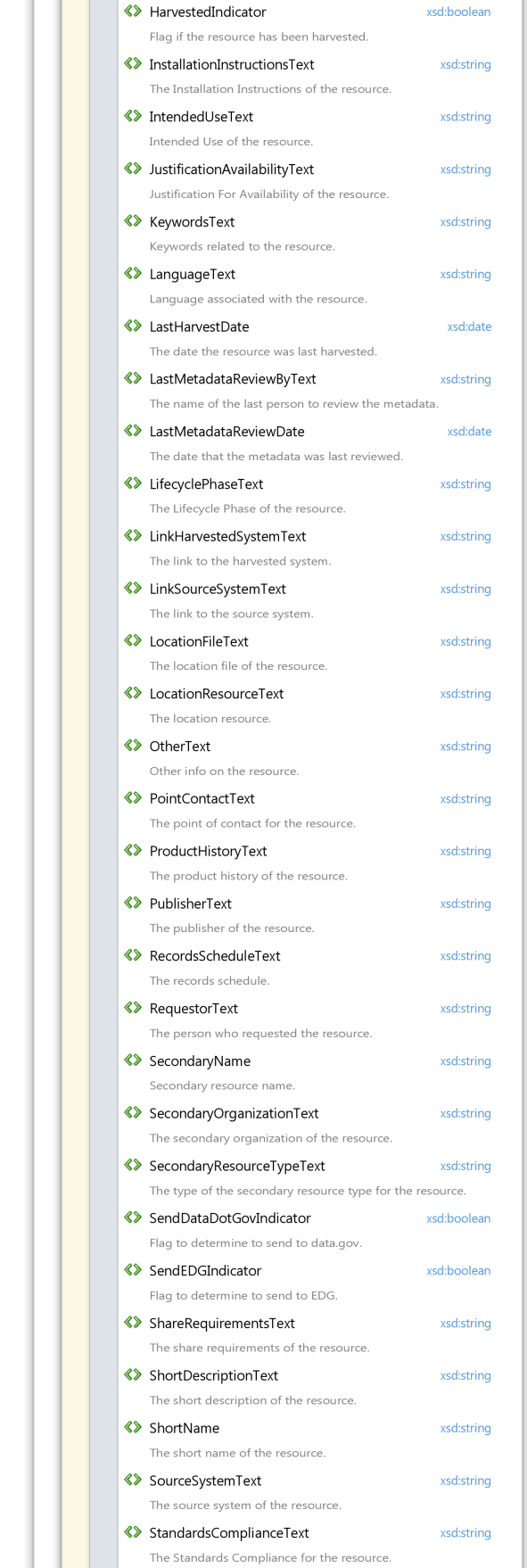
**Input:** The ResourceIdentifier ‘6bd8a515-dd80-11df-832e-92ce4b1f0c29’

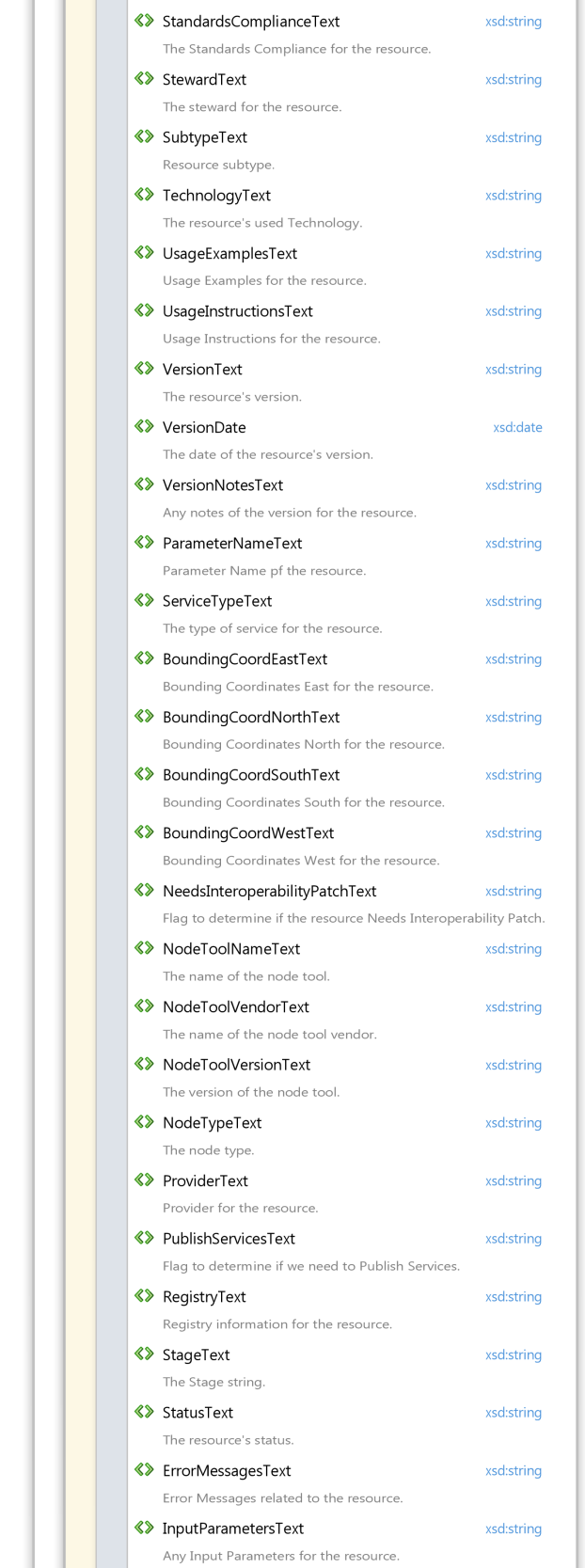
**Output:** User receives the metadata for the resource corresponding to identifier ‘6bd8a515-dd80-11df-832e-92ce4b1f0c29’, which is the System-type resource Reusable Component Services.

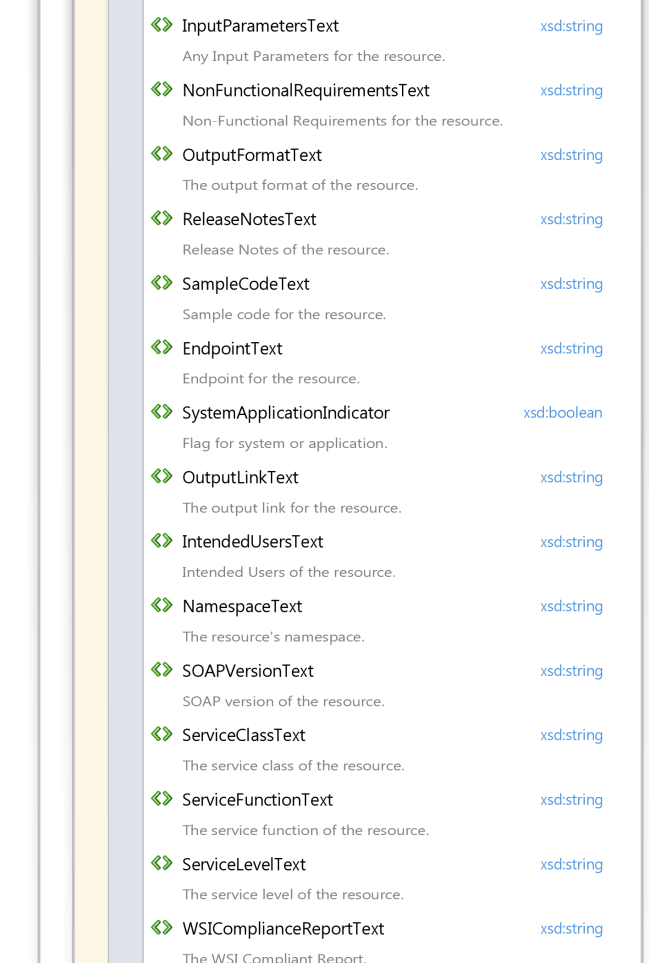
# Appendix A: Schema Information

The following diagram depicts a high-level view of the RCS schema. A single resource entity is represented by the Resource tag.









# Appendix B: Sample SOAP Requests and Responses

Exhibit 7‑1 GetRCSResourcesByType\_v1.0 Sample Request

<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:Query>

<ns:securityToken>

<!-- securityToken removed from example request -->

</ns:securityToken>

<ns:dataflow>RCS</ns:dataflow>

<ns:request>GetRCSResourcesByType\_v1.0</ns:request>

<ns:parameters parameterName="ResourceTypeName">database</ns:parameters>

</ns:Query>

</soap:Body>

</soap:Envelope>

Exhibit 7‑2 GetRCSResourcesByType\_v1.0 Sample Response

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<QueryResponse xmlns="http://www.exchangenetwork.net/schema/node/2">

<rowId>0</rowId>

<rowCount>2</rowCount>

<lastSet>false</lastSet>

<results>

<NewDataSet xmlns="http://www.exchangenetwork.net/schema/RCS/1">

<Resource>

<ResourceIdentifier>9ec30f37-0fd0-11e1-a787-bf5575d3f06a</ResourceIdentifier>

<ResourceTypeText>Database</ResourceTypeText>

<ResourceName>Deep Water Horizon Oil Spill Field Dataset</ResourceName>

<PrimaryOrganizationText>USEPA - United States Environmental Protection Agency</PrimaryOrganizationText>

<DescriptionText>The Deep Water Horizon Oil Spill Field Dataset contains the sampling and monitoring data that EPA collected during the incident. It is stored in an Oracle database.</DescriptionText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CreatedDate>2011-11-15T00:00:00-05:00</CreatedDate>

<GISCapableIndicator>false</GISCapableIndicator>

<LifecyclePhaseText>Operations and Maintenance</LifecyclePhaseText>

<SubtypeText>Lotus-Notes</SubtypeText>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>The Deep Water Horizon Oil Spill Field Dataset contains the sampling and monitoring data that EPA collected during the incident. It is stored in an Oracle database.</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>0ff6a242-c82f-11e0-904e-a4a3a3d6ae76</ResourceIdentifier>

<ResourceTypeText>Database</ResourceTypeText>

<ResourceName>WATERS10</ResourceName>

<PrimaryOrganizationText>USEPA - United States Environmental Protection Agency</PrimaryOrganizationText>

<DescriptionText>Allows applications to directly access the NHD event indexing service via a stored procedure interface</DescriptionText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CommentsText>Must have a schema account on the EPA WATERS10 database instance</CommentsText>

<CreatedDate>2011-08-16T00:00:00-04:00</CreatedDate>

<GISCapableIndicator>false</GISCapableIndicator>

<KeywordsText>NHD</KeywordsText>

<LifecyclePhaseText>Operations and Maintenance</LifecyclePhaseText>

<SecondaryName>Event Indexing service</SecondaryName>

<SubtypeText>Relational</SubtypeText>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>Allows applications to directly access the NHD event indexing service via a stored procedure interface</Description>

<LinkSourceSystemText>www.epa.gov/waters/geoservices/docs/event\_indexing\_service.html</LinkSourceSystemText>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

</NewDataSet>

</results>

</QueryResponse>

</s:Body>

</s:Envelope>

Exhibit 7‑3 GetRCSResourcesByType\_v1.0 Sample Request

<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:Query>

<ns:securityToken>

<!-- securityToken removed from example request -->

</ns:securityToken>

<ns:dataflow>RCS</ns:dataflow>

<ns:request>GetRCSResourcesByType\_v1.0</ns:request>

<ns:parameters parameterName="ResourceTypeName">blocks of code</ns:parameters>

</ns:Query>

</soap:Body>

</soap:Envelope>

Exhibit 7‑4 GetTribes Sample Response

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<QueryResponse xmlns="http://www.exchangenetwork.net/schema/node/2">

<rowId>0</rowId>

<rowCount>8</rowCount>

<lastSet>false</lastSet>

<results>

<NewDataSet xmlns="http://www.exchangenetwork.net/schema/RCS/1">

<Resource>

<ResourceIdentifier>45d34781-8d75-11e1-afb0-ec6a440d2a94</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>CDX to ARCS SQL</ResourceName>

<PrimaryOrganizationText>EPA / OW - Office of Water</PrimaryOrganizationText>

<DescriptionText>CDX to ARCS SQL code</DescriptionText>

<AccessLevelText>Everybody</AccessLevelText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CodeLanguageText>sql</CodeLanguageText>

<CreatedDate>2012-04-23T00:00:00-04:00</CreatedDate>

<LocationFileText>instantclient-sqlplus-nt-11.2.0.4.0.zip</LocationFileText>

<OtherText>https://rcs.epa.gov:443/CentraSite/CentraSite/ino:dav/ino:dav/projects/SupportingDocumentsLibrary/uddi:bcf3a8ad-61c5-11df-bd49-9af1668c06ac/CDX\_TO\_ARCS.sql</OtherText>

<AvailablePublicIndicator>false</AvailablePublicIndicator>

<Description>CDX to ARCS SQL code</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>04176671-8d78-11e1-afb0-88c3531df5d8</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>Create public APEX Page</ResourceName>

<PrimaryOrganizationText>EPA / OW - Office of Water</PrimaryOrganizationText>

<DescriptionText>Create public APEX page that will take the ID as parameter and a process to look up the the CDX data and token values and pass the user through APEX authentication if token checks out.</DescriptionText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CodeLanguageText>sql</CodeLanguageText>

<CreatedDate>2012-04-23T00:00:00-04:00</CreatedDate>

<OtherText>https://rcs.epa.gov:443/CentraSite/CentraSite/ino:dav/ino:dav/projects/SupportingDocumentsLibrary/uddi:bcf3a8ad-61c5-11df-bd49-9af1668c06ac/HANDOFF.sql</OtherText>

<AvailablePublicIndicator>false</AvailablePublicIndicator>

<Description>Create public APEX page that will take the ID as parameter and a process to look up the the CDX data and token values and pass the user through APEX authentication if token checks out.</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>b4bd5f52-98e0-11e3-a646-a56b3585dd38</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>FRS REST API Code Samples</ResourceName>

<DescriptionText>Github repository of simple code samples demonstrating use of EPA Facility Registry Service Facility Lookup API via HTML, JavaScript and jQuery.</DescriptionText>

<AuthorText>David G. Smith - smith.davidg@epa.gov</AuthorText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CodeLanguageText>HTML/JavaScript</CodeLanguageText>

<CreatedDate>2014-02-18T00:00:00-05:00</CreatedDate>

<DataDotGovTypeText>Tool Catalog</DataDotGovTypeText>

<DatabasePlatformText>N/A - web client</DatabasePlatformText>

<DocumentationText>http://www.epa.gov/enviro/html/fii/FRS\_REST\_Services.html</DocumentationText>

<DownloadableIndicator>true</DownloadableIndicator>

<GuidanceDocumentsText>https://github.com/USEPA/FRS-getfacilities-samples/blob/master/README.md</GuidanceDocumentsText>

<KeywordsText>EPA, Facilities, FRS, REST, jQuery, API</KeywordsText>

<LocationResourceText>https://github.com/USEPA/FRS-getfacilities-samples</LocationResourceText>

<PublisherText>David G. Smith - smith.davidg@epa.gov</PublisherText>

<SendDataDotGovIndicator>false</SendDataDotGovIndicator>

<SendEDGIndicator>false</SendEDGIndicator>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>Github repository of simple code samples demonstrating use of EPA Facility Registry Service Facility Lookup API via HTML, JavaScript and jQuery.</Description>

<HardwareRequirementsPlatformTe>N/A - web client</HardwareRequirementsPlatformTe>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>3fd58701-8d64-11e1-afb0-9612e87d14ff</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>Javascript Code for ARCS APEX tool</ResourceName>

<PrimaryOrganizationText>EPA / OW - Office of Water</PrimaryOrganizationText>

<DescriptionText>Javascript code ARCS APEX tool</DescriptionText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CreatedDate>2012-04-23T00:00:00-04:00</CreatedDate>

<OtherText>https://rcs.epa.gov:443/CentraSite/CentraSite/ino:dav/ino:dav/projects/SupportingDocumentsLibrary/uddi:bcf3a8ad-61c5-11df-bd49-9af1668c06ac/thickbox</OtherText>

<AvailablePublicIndicator>false</AvailablePublicIndicator>

<Description>Javascript code ARCS APEX tool</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>366</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>RCS New Resource test6585</ResourceName>

<PrimaryOrganizationText>EPA / OPPTS - Office of Prevention, Pesticides, and Toxic Substances</PrimaryOrganizationText>

<DescriptionText>Description 1</DescriptionText>

<CreatedDate>2014-10-28T00:00:00-04:00</CreatedDate>

<RequestorText>Timothy Bizal</RequestorText>

<StewardText>Timothy Bizal</StewardText>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>Description 1</Description>

</Resource>

<Resource>

<ResourceIdentifier>386</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>RCS New Resource test9363</ResourceName>

<PrimaryOrganizationText>EPA / OPPTS - Office of Prevention, Pesticides, and Toxic Substances</PrimaryOrganizationText>

<DescriptionText>Description 1</DescriptionText>

<CreatedDate>2014-12-08T00:00:00-05:00</CreatedDate>

<RequestorText>Timothy Bizal</RequestorText>

<StewardText>Timothy Bizal</StewardText>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>Description 1</Description>

</Resource>

<Resource>

<ResourceIdentifier>f8284304-5281-11e1-8c35-9686dd36974f</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>SDE Layer Menu Add-in to ArcGIS 10</ResourceName>

<PrimaryOrganizationText>EPA / USEPA / Region 6 - R6</PrimaryOrganizationText>

<DescriptionText>An add-in to ArcGIS 10 that allows ArcMap users to turn on GIS layers stored in SDE by simply checking boxes next to layers listed on a pop up menu. Adding layers is quick and easy as the user does not have to manually hunt for layers in SDE.</DescriptionText>

<AuthorText>Brittany Roche</AuthorText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CodeLanguageText>Visual Basic .NET</CodeLanguageText>

<CreatedDate>2012-02-08T00:00:00-05:00</CreatedDate>

<PublisherText>EPA Region 6</PublisherText>

<SourceSystemText>EPA Region 6</SourceSystemText>

<StewardText>Jeff Danielson</StewardText>

<AvailablePublicIndicator>false</AvailablePublicIndicator>

<Description>An add-in to ArcGIS 10 that allows ArcMap users to turn on GIS layers stored in SDE by simply checking boxes next to layers listed on a pop up menu. Adding layers is quick and easy as the user does not have to manually hunt for layers in SDE.</Description>

<HardwareRequirementsPlatformTe>ArcGIS 10</HardwareRequirementsPlatformTe>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

<Resource>

<ResourceIdentifier>b7d63d37-a379-11e2-bda6-e52fabc5b1c1</ResourceIdentifier>

<ResourceTypeText>Blocks of Code</ResourceTypeText>

<ResourceName>Term Lookup Tooltip Example Code</ResourceName>

<PrimaryOrganizationText>EPA / OEI / OIAA - Office of Information Analysis and Access</PrimaryOrganizationText>

<DescriptionText>Example code for how terms should be displayed when using the term lookup service if displaying the definitions in a tooltip.</DescriptionText>

<AuthorText>CGI Federal</AuthorText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CodeLanguageText>HTML, Javascript and CSS</CodeLanguageText>

<CreatedDate>2013-04-12T00:00:00-04:00</CreatedDate>

<DataDotGovTypeText>Tool Catalog</DataDotGovTypeText>

<DownloadableIndicator>true</DownloadableIndicator>

<LifecyclePhaseText>Operations and Maintenance</LifecyclePhaseText>

<OtherText>https://rcs.epa.gov:443/CentraSite/CentraSite/ino:dav/ino:dav/projects/SupportingDocumentsLibrary/uddi:bcf3a8ad-61c5-11df-bd49-9af1668c06ac/Term Lookup Tooltip Demo Code</OtherText>

<PublisherText>EPA</PublisherText>

<SendDataDotGovIndicator>false</SendDataDotGovIndicator>

<SendEDGIndicator>false</SendEDGIndicator>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>Example code for how terms should be displayed when using the term lookup service if displaying the definitions in a tooltip.</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

</NewDataSet>

</results>

</QueryResponse>

</s:Body>

</s:Envelope>

Exhibit 7‑5 GetRCSResourceByIdentifier\_v1.0 Sample Request

<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:Query>

<ns:securityToken>

<!-- securityToken removed from example request -->

</ns:securityToken>

<ns:dataflow>RCS</ns:dataflow> <ns:request>GetRCSResourceByIdentifier\_v1.0</ns:request>

<ns:parameters parameterName="ResourceIdentifier">9ec30f37-0fd0-11e1-a787-bf5575d3f06a</ns:parameters>

</ns:Query>

</soap:Body>

</soap:Envelope>

Exhibit 7‑6 GetRCSResourceByIdentifier\_v1.0 Sample Response

<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope">

<s:Body xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<QueryResponse xmlns="http://www.exchangenetwork.net/schema/node/2">

<rowId>0</rowId>

<rowCount>1</rowCount>

<lastSet>false</lastSet>

<results>

<NewDataSet xmlns="http://www.exchangenetwork.net/schema/RCS/1">

<Resource>

<ResourceIdentifier>9ec30f37-0fd0-11e1-a787-bf5575d3f06a</ResourceIdentifier>

<ResourceTypeText>Database</ResourceTypeText>

<ResourceName>Deep Water Horizon Oil Spill Field Dataset</ResourceName>

<PrimaryOrganizationText>USEPA - United States Environmental Protection Agency</PrimaryOrganizationText>

<DescriptionText>The Deep Water Horizon Oil Spill Field Dataset contains the sampling and monitoring data that EPA collected during the incident. It is stored in an Oracle database.</DescriptionText>

<BuiltUsingENGrantIndicator>false</BuiltUsingENGrantIndicator>

<CreatedDate>2011-11-15T00:00:00-05:00</CreatedDate>

<GISCapableIndicator>false</GISCapableIndicator>

<LifecyclePhaseText>Operations and Maintenance</LifecyclePhaseText>

<SubtypeText>Lotus-Notes</SubtypeText>

<AvailablePublicIndicator>true</AvailablePublicIndicator>

<Description>The Deep Water Horizon Oil Spill Field Dataset contains the sampling and monitoring data that EPA collected during the incident. It is stored in an Oracle database.</Description>

<VersiosubnText>1.0</VersiosubnText>

</Resource>

</NewDataSet>

</results>

</QueryResponse>

</s:Body>

</s:Envelope>