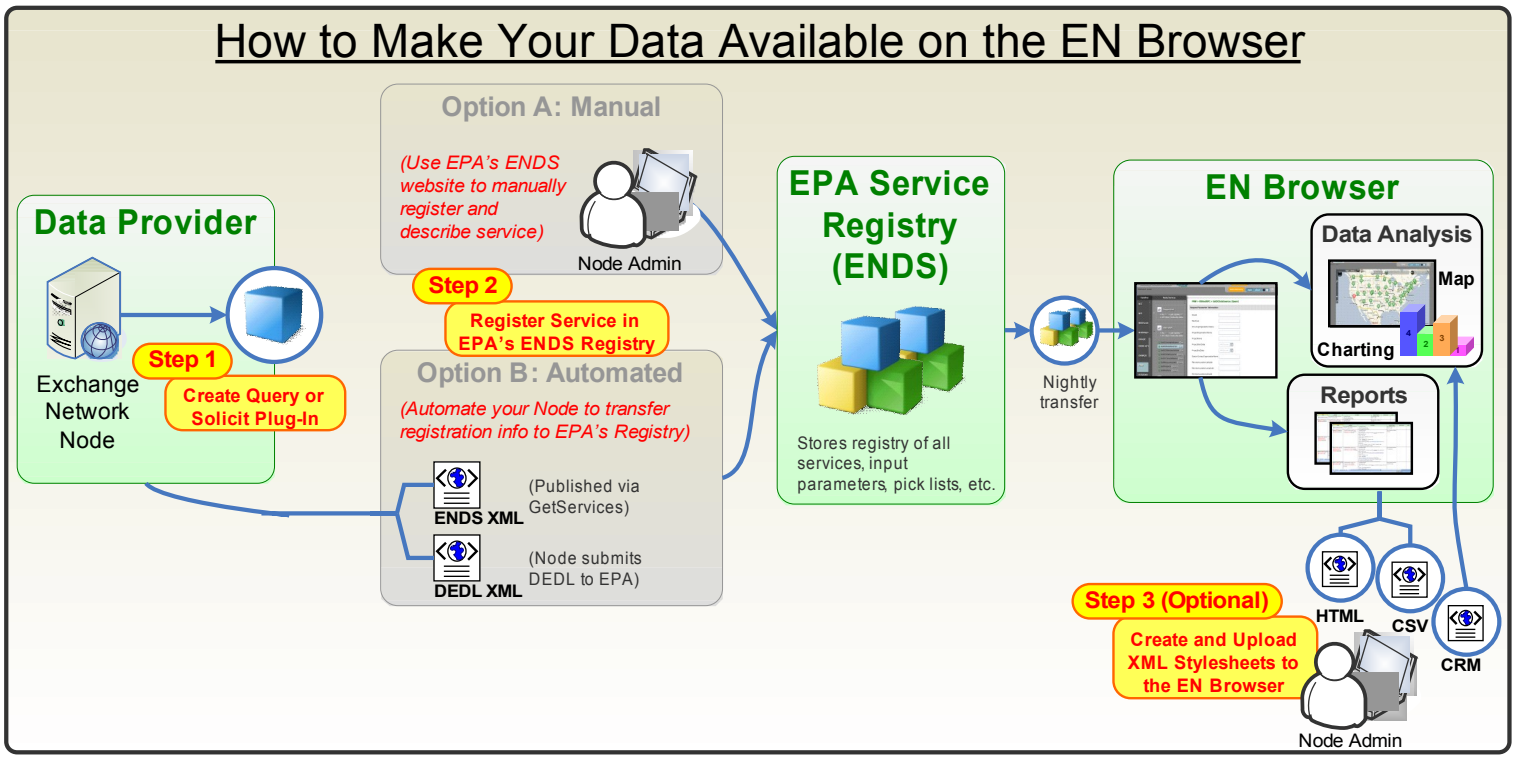
How to Make Your Data Available through the EN Browser

Overview

Making your data available through the EN Browser can be completed in **3 steps**. This document guides you through these steps.



Step 1: Create a Query or Solicit Plug-In

Step 1
Create Query or Solicit Plug-In

Using your Exchange Network Node, create a new Query or Solicit data service. The details on how to complete this step are outside the scope of this document. If you do not know what a Node is or how to create a new Query/Solicit service, please refer to the following links for additional assistance:

- What is an Exchange Network Node?
- How Do I Get Started with the Exchange Network?
- How Do I Build a New Query or Solicit Plug-In?
- …or contact your Node Administrator or Node service provider for additional assistance

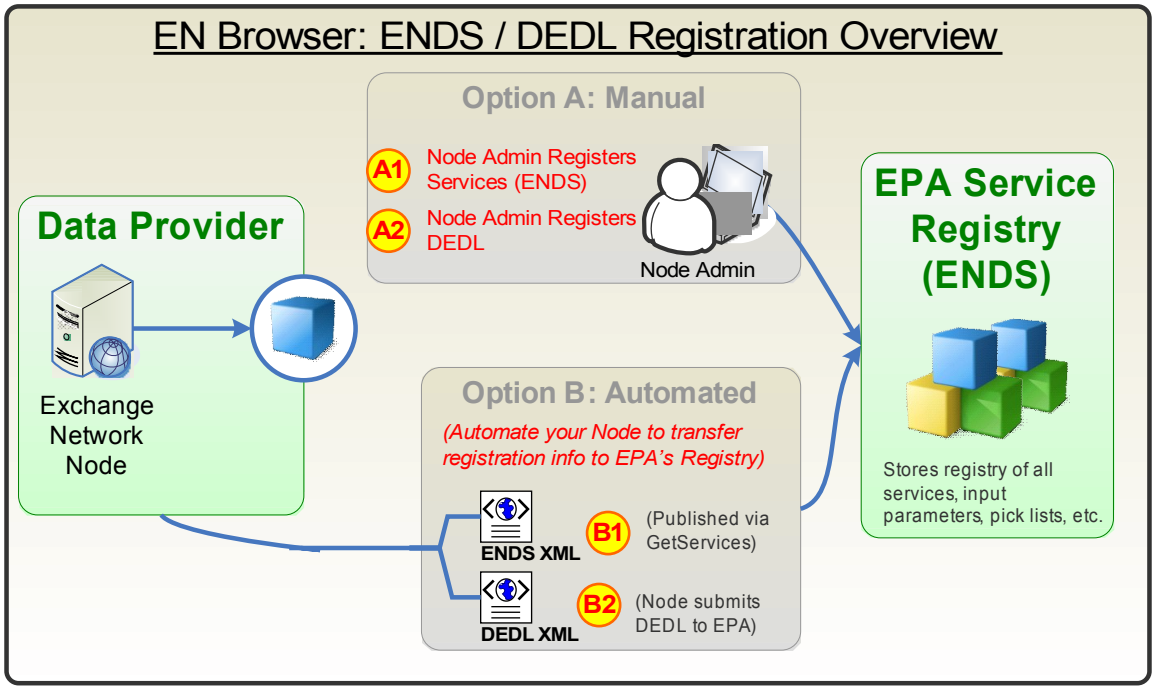
Tip: *When designing your Query or Solicit outbound service, think about how you want people to retrieve your data (for example, do you want them to query data geospatially, by date range, etc.). This will affect which input parameters you include for your Query or Solicit service.*

Step 2

Register Service in EPA's ENDS Registry

Step 2: Register your Query or Solicit Service

In order to discover and retrieve data, data providers must register their data services at EPA's Exchange Network Discovery Service (ENDS). There are two registration options available to data publishers, as outlined here:



Option A: Manual:

A1 A2 Node Administrators can log into the EPA-CDX ENDS User Interface and manually enter in Node, Service, Service Parameter, and Data Element (i.e. DEDL) information. The EPA-CDX ENDS user interface can be accessed here:

- **Test Environment** (used when registering services that use test NAAS): https://ends2.epacdxnode.net
- **Production Environment** (used when registering services that use production NAAS): https://ends2.epa.gov

Screens are provided to allow this entering of information. The more information that is supplied, the better your service will appear in the EN Browser.

The following table outlines the different fields that can be filled out and how they impact how your data service appears in the EN Browser:

Node-Level Fields

Field	Example	Notes (How Used by EN Browser)
Node Identifier	NJ	
Description	NJDEP Test Node	
URL	https://www11-stg.state.nj.us/Node.WebServices/services/NetworkNodeSoap_V11	**Very Important:** the EN Browser will use this to locate where your node is when retrieving data.
Stage	Test or Production	It is important to register only test services in test ENDS and production services in production ENDS, otherwise they will not show up in the EN Browser
Provider	NJ	
Contact	John-doe@test.gov	
Publish Services	No	If this is set to "Yes", then EPA's ENDS service will automatically call your Node's GetServices method to retrieve service registration.
Bounding Coordinates	35 / 29 / -85 / -88	**Very Important:** the EN Browser uses these to plot your Node on the En Browser map.
Node Properties		Not used by EN Browser

Service and Service Parameter Level Fields

Field	Example	Notes
ServiceName	GetWaterQualityInvSolicit	
Description	Solicit Water Quality Inventory (station) data.	
Service Type	Solicit or Query	**If you use a value other than Query or Solicit, it will not appear in the EN Browser.**
Document URL		Not used by EN Browser
Dataflow Name	WQX	
XSLT URL		Not used by EN Browser
ParameterName	OrganizationTypeCode	
Type	String	**Very Important:** the EN Browser uses this to determine which control to display when a user is retrieving data from your service. The following values are supported: • **string:** will display textbox • **date:** will display calendar control • **time:** will display clock control

Field	Example	Notes
		• **map:** will display map control (requires other fields to be registered – see DEDL)
Encoding		Not used by EN Browser
Required Indicator	False	If True, the EN Browser will mark with red asterisk and force user to enter a value
Occur	1	Not used by EN Browser
Descriptor	NJ.WQDEOrgName	This maps a service parameter with a "Data Element definition" (DEDL). This is essentially a foreign key and must exactly match the name of the Data Element definition (see below)

Data Element Definition Fields

Field	Example	Notes
Element Identifier	NJ.WQDEOrgName	Unique ID for the data element definition. This is used to map with a Service Parameter Descriptor (see above) and must match exactly.
Application Domain	Default	Not used by EN Browser
Element Type	String	Not used by EN Browser
Description	Please enter City Code (5 digits)	When the user hovers their mouse over a field in the EN Browser, this will display
Keywords		Not used by EN Browser
Owner		Not used by EN Browser
Element Label	City Code	This is the label that will display to the user for each parameter in your Query/Solicit service
Default Value		If supplied, this will be defaulted in the textbox or other control in the EN Browser.
Data Source fields		Not used by EN Browser
UpperLimit	50000	If supplied, when the user enters a value outside of the valid range, an error message will appear. This is only valid for integer values.
Lower Limit	5	If supplied, when the user enters a value outside of the valid range, an error message will appear. This is only valid for integer values.
Allow Multi-select	true	If set to 'true', then a listbox will appear in the EN Browser for this parameter. **Must be used in conjunction with a MultiSelectSeparator property (see below)**
Add. Values Indicator	True	If set to true, then the user will be able to add values in addition to the drop-down values that are supplied
Optionality		Not used by EN Browser
Wildcard		Not used by EN Browser
Format String		The required format of the data element. This applies mainly to fixed strings such as dates, telephone numbers or social security numbers. For example, a date format could be defined as YYYY-

Field	Example	Notes
		MM-DD. Predefined characters for building FormatString patterns are provided here: YYYY = four-digit year MM = two-digit month (01=January, etc.) DD = two-digit day of month (01 through 31) hh = two digits of hour (00 through 23) (am/pm NOT allowed) mm = two digits of minute (00 through 59) ss = two digits of second (00 through 59) s = one or more digits representing a decimal fraction of a second TZD = time zone designator (Z or +hh:mm or -hh:mm) **Important:** This is required for date-based parameters.
Validation Rules	^([0-9]*|\d*\.\d{1}?\d*)$	A regular expression that can be used by the EN Browser to enforce the formatting of a parameter. For example, it could be used to enforce that the user supplies an Integer for a parameter.
Properties		**Important:** The EN Browser uses a few pre-defined data element property names. These are used to provide functionality beyond what the ENDS/DEDL provides by default. These are entered as name/value pairs. The following table lists supported Property Names: ○ **MultiSelectSeparator:** Identifies how parameter values are separated when a parameter allows the selection of multiple values. ○ **WMSResource:** Provides a link to a Web Map Service used to aid in selecting parameter values ○ **WFSResource:** Provides a link to a Web Feature Service used to return the value after interacting with the WMSResource above ○ **BaseMapLayerName:** Provides a name of a base map to display when rendering the return from the WMS service call ○ **ActiveMapLayerName:** Provides a name of the active layer to display when rendering the return from the WMS service call ○ **OtherDisplayLayerName:** Provides a name of any other layer names from the hosted WMS server that the data provider wishes to allow the user to view. ○ **DisplayProjection:** Projection of the map for display and calculation ○ **Feature:** Name of the feature type from the WFS service for geographic feature retrieving ○ **ReturnFieldValue:** value returned from the WFS service call to be displayed ○ **ReturnFieldID:** value returned from the WFS service call to be recorded as the selected

Field	Example	Notes
		parameter value
Element Value-Name	14	When defining a pick list (i.e. a drop-down list) that a user can choose from when providing the value for a parameter, this is the value that will be passed to your Query/Solicit service.
Element Value-Value	MERCER	When defining a pick list (i.e. a drop-down list) that a user can choose from when providing the value for a parameter, this is the value that the user will see, which may be different from the value that is passed to the Query/Solicit service (see row above).

Option B: Automated:

B1 ENDS:

Automated:

Some Nodes have the option of being able to make available their data service registration information directly from their Nodes via a "GetService" service call. In this scenario, the EPA ENDS server will be able to periodically make a GetService Web service call on the Nodes to get the latest registration information. This will apply to Nodes that have signed up for automated ENDS retrieval. This GetServices call will return an ENDS XML file to EPA-CDX's ENDS service. Check with your Node service provider about the availability of this feature.

Manual XML submission:

If your node doesn't support GetServices (or the information supplied by your Node's GetServices is incorrect), you can still generate an XML file containing your services and submit to ENDS. The ENDS endpoints for submission are:

Test: https://ends2.epacdxnode.net/Node2WS.svc
Production: https://ends2.epa.gov/Node2WS.svc
Submit Dataflow: ENDS_v20
Submit Flow Operation: Refresh

B2 DEDL:

EPA provides a Submit service (DEDL_v1_0) that allows Data Providers to submit their DEDL information directly to EPA-CDX's ENDS. Here are the parameters to use when making the submit call:

Test: https://ends2.epacdxnode.net/Node2WS.svc
Production: https://ends2.epa.gov/Node2WS.svc
Submit Dataflow: DEDL_v1_0
Submit Flow Operation: Refresh

Step 3 (Optional): Create and Upload XML Stylesheets to the EN Browser

Step 3 (Optional)

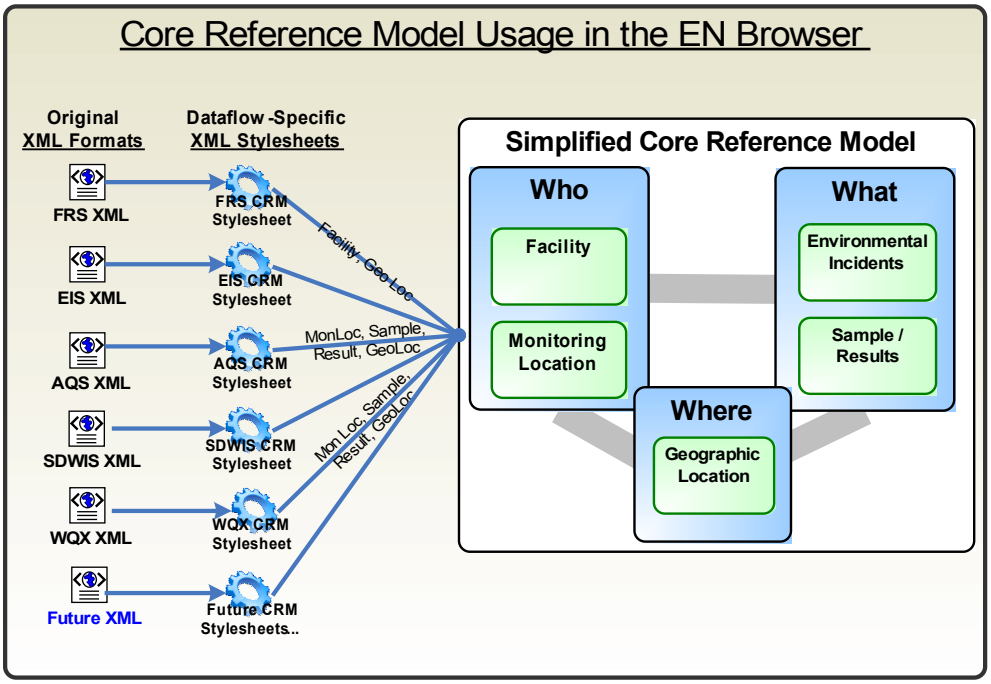
Create and Upload XML Stylesheets to the EN Browser

1.1 Introduction

The EN Browser allows Node Administrators to upload stylesheets that provide additional functionality. These stylesheets are not required in order for the EN Browser to be able to interact with your Query or Solicit service, but provide additional features after data has been retrieved. There are two major categories of stylesheets supported by the EN Browser:

- **Data Transformation (HTML, CSV, etc):** You can develop an XML stylesheet to convert the XML files generated from your Query/Solicit service into any text-based data format supported by XSLT transformation. This could be HTML, CSV, another XML file, GML, KML, etc.

- **Core Reference Model Transformation:** The EN Browser is able to plot your returned data on a map or display on a graph (e.g. time series graph). The way this is accomplished is by transforming the original XML file into a standard XML format. This standard XML format is called Core Reference Model (CRM). It provides a very simple basic representation of environmental data that spans all different environmental program areas. Therefore, in order for your dataflow to be able to be plotted on maps or charts in the EN Browser, you must create and upload a CRM stylesheet for your dataflow (unless another data provider publishing the same data format has already done this step).



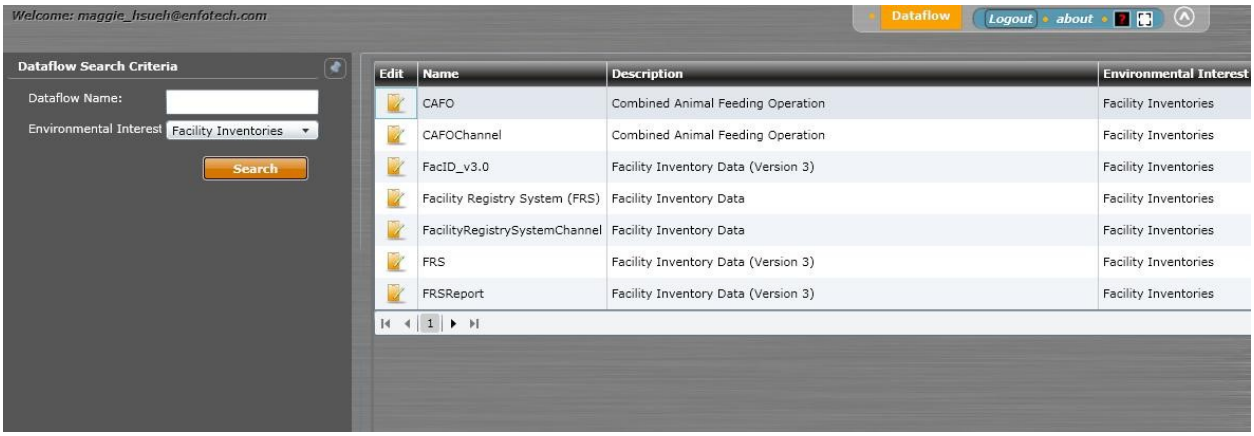
1.2 Creating Stylesheets

- **Data Transformation (HTML, CSV, etc):** This is relatively straightforward. You can develop the stylesheet however you wish, as long as the output is a valid text-based file without errors.
- **Core Reference Model Transformation:** The EN Browser dictates an output format for CRM stylesheets. The stypesheet must convert the original XML file to a format that validates against the CRM XML schema, which is posted in the Appendix of this document.

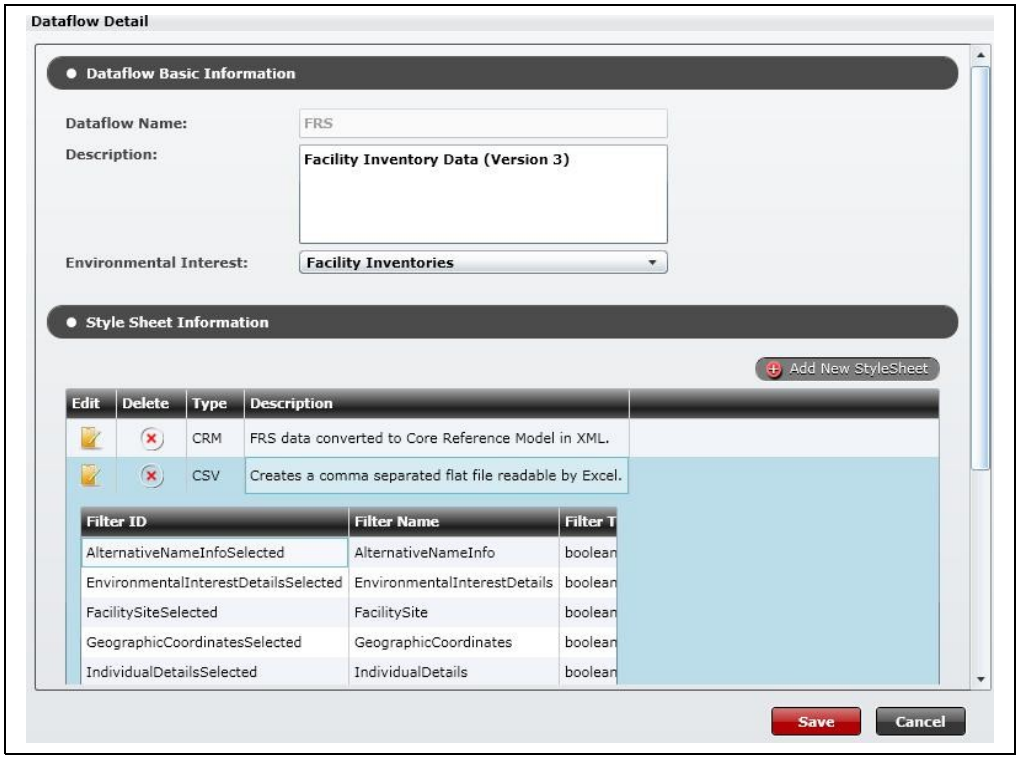
1.3 Uploading Stylesheets to the EN Browser:

Step 1: Log in to the EN Browser as a Node Administrator. This is a NAAS user that has been granted 'Admin' rights by CDX. Each Node has one or two Node Administrator accounts.

Step 2: When a Node Administrator logs into the EN Browser, an additional menu option will appear called "Dataflow Admin" as shown here:



The Node Admin can click on the icon for a row to edit the dataflow, as shown here:



The following information can be updated:

- **Dataflow Level Information:** Allows the EN Browser administrator to add, update, or delete essential dataflow information. This includes a user-friendly Dataflow Name and Dataflow Description for an unlimited number of dataflows. The reason these are managed within the EN Browser as opposed to being defined by each Node data provider is because this information spans multiple data providers and needs to be consistent across multiple providers (i.e. states) in order to provide consistency expected by the tool.) This will consist of 3 columns:

    - **DataflowDescription:** provides a user friendly data flow description
    - **Environmental Interest:** identifies the Environmental Interest under which this Dataflow is categorized. This will affect where it shows up on the Environmental Interest map in the EN Browser.
    - **Dataflow Stylesheets:**
        - Each stylesheet that is registered in the system has the following attributes:
            - Dataflow
            - XSLT Type (HTML, XML, CSV, GML, CRM)
            - XSLT Description (to provide a meaningful description of the stylesheet)

Stylesheet Variables: The system also supports stylesheet variables. When the stylesheet is registered, stylesheet variables can be defined. When the user selects a particular stylesheet to apply transformation, the system will prompt to supply values for each variable. This can be

used for a variety of purposes, including filtering a stylesheet to only include certain pieces of an XML file for transformation.

Appendix A: CRM XML schema

```
<?xml version="1.0" encoding="utf-8"?>
<xsd:schema xmlns:crm="http://www.exchangenetwork.net/schema/crm/1" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.exchangenetwork.net/schema/crm/1" elementFormDefault="qualified"
attributeFormDefault="unqualified" version="1.0">
	<xsd:annotation>
		<xsd:documentation>
	Schema Name   : CRM_CRM_v1.0.xsd
	Description       : Core Reference Model Schema
	Point of Contact :  Doug Timms
	Primary schema element: CRM
		</xsd:documentation>
	</xsd:annotation>
	<xsd:element name="CRM" type="crm:CRMDataType">
		<xsd:annotation>
			<xsd:documentation>Main Schema used to transfer processing reports.</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="CRMDataType">
		<xsd:sequence>
			<xsd:element ref="crm:Facilities" minOccurs="0"/>
			<xsd:element ref="crm:MonitoringLocations" minOccurs="0"/>
			<xsd:element ref="crm:EnvironmentalIncidents" minOccurs="0"/>
			<xsd:element ref="crm:Samples" minOccurs="0"/>
			<xsd:element ref="crm:GeographicLocations" minOccurs="0"/>
		</xsd:sequence>
	</xsd:complexType>
	<xsd:element name="Facilities" type="crm:FacilitiesDataType">
		<xsd:annotation>
			<xsd:documentation>Provides a listing of facilities.</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="FacilitiesDataType">
		<xsd:sequence>
			<xsd:element ref="crm:Facility" minOccurs="0" maxOccurs="unbounded"/>
		</xsd:sequence>
	</xsd:complexType>
	<xsd:element name="Facility" type="crm:FacilityDataType">
		<xsd:annotation>
			<xsd:documentation>A Facility.</xsd:documentation>
		</xsd:annotation>
	</xsd:element>
	<xsd:complexType name="FacilityDataType">
		<xsd:sequence>
			<xsd:element name="FacilitySiteIdentifier" type="xsd:string"/>
			<xsd:element name="FacilitySiteName" type="xsd:string" minOccurs="0"/>
			<xsd:element name="FacilitySiteType" type="xsd:string" minOccurs="0"/>
			<xsd:element name="FederalFacilityIndicator" type="xsd:string" minOccurs="0"/>
			<xsd:element name="GeographicLocationIdentifier" type="xsd:string" minOccurs="0"/>
			<xsd:element ref="crm:FacilityProperties" minOccurs="0"/>
			<xsd:element ref="crm:FacilityChemicalStorageReports" minOccurs="0"/>
		</xsd:sequence>
	</xsd:complexType>
	<!-- FACILITY PROPERTIES -->
	<xsd:element name="FacilityProperties" type="crm:FacilityPropertiesDataType">
		<xsd:annotation>
			<xsd:documentation>A listing of facility properties.</xsd:documentation>
```

```
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="FacilityPropertiesDataType">
            <xsd:sequence>
                    <xsd:element ref="crm:FacilityProperty" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="FacilityProperty" type="crm:FacilityPropertyDataType">
            <xsd:annotation>
                    <xsd:documentation>A facility property.</xsd:documentation>
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="FacilityPropertyDataType">
            <xsd:attribute name="name" type="xsd:string" use="required"/>
            <xsd:attribute name="value" type="xsd:string" use="optional"/>
    </xsd:complexType>
    <!-- FACILITY CHEMICAL STORAGE -->
    <xsd:element name="FacilityChemicalStorageReports" type="crm:FacilityChemicalStorageReportsDataType">
            <xsd:annotation>
                    <xsd:documentation>A listing of chemicals storage reports.</xsd:documentation>
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="FacilityChemicalStorageReportsDataType">
            <xsd:sequence>
                    <xsd:element ref="crm:FacilityChemicalStorageReport" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="FacilityChemicalStorageReport" type="crm:FacilityChemicalStorageReportDataType">
            <xsd:annotation>
                    <xsd:documentation>A facility chemical storage report.</xsd:documentation>
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="FacilityChemicalStorageReportDataType">
            <xsd:sequence>
                    <xsd:element name="ReportIdentifier" type="xsd:string"/>
                    <xsd:element name="ReportReceivedDate" type="xsd:date" minOccurs="0"/>
                    <xsd:element name="ReportRecipientName" type="xsd:date" minOccurs="0"/>
                    <xsd:element name="ReportingPeriodStartDate" type="xsd:date" minOccurs="0"/>
                    <xsd:element name="ReportingPeriodEndDate" type="xsd:date" minOccurs="0"/>
                    <xsd:element name="SubstanceIdentifier" type="xsd:string" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    <!-- MONITORING LOCATIONS -->
    <xsd:element name="MonitoringLocations" type="crm:MonitoringLocationsDataType">
            <xsd:annotation>
                    <xsd:documentation>Provides a listing of monitoring locations.</xsd:documentation>
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="MonitoringLocationsDataType">
            <xsd:sequence>
                    <xsd:element ref="crm:MonitoringLocation" minOccurs="0" maxOccurs="unbounded"/>
            </xsd:sequence>
    </xsd:complexType>
    <xsd:element name="MonitoringLocation" type="crm:MonitoringLocationDataType">
            <xsd:annotation>
                    <xsd:documentation>A monitoring location.</xsd:documentation>
            </xsd:annotation>
    </xsd:element>
    <xsd:complexType name="MonitoringLocationDataType">
            <xsd:sequence>
                    <xsd:element name="MonitoringLocationIdentifier" type="xsd:string"/>
                    <xsd:element name="FacilitySiteIdentifier" type="xsd:string" minOccurs="0"/>
```

```
                <xsd:element name="MonitoringLocationName" type="xsd:string" minOccurs="0"/>
                <xsd:element name="MonitoringLocationTypeName" type="xsd:string" minOccurs="0"/>
                <xsd:element name="MonitoringLocationDescriptionText" type="xsd:string" minOccurs="0"/>
                <xsd:element name="GeographicLocationIdentifier" type="xsd:string" minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>
<!-- SAMPLES -->
<xsd:element name="Samples" type="crm:SamplesDataType">
        <xsd:annotation>
                <xsd:documentation>Provides a listing of samples.</xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SamplesDataType">
        <xsd:sequence>
                <xsd:element ref="crm:Sample" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
</xsd:complexType>
<xsd:element name="Sample" type="crm:SampleDataType">
        <xsd:annotation>
                <xsd:documentation>A sample.</xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SampleDataType">
        <xsd:sequence>
                <xsd:element name="SampleIdentifier" type="xsd:string"/>
                <xsd:element name="SampleMedia" type="xsd:string" minOccurs="0"/>
                <xsd:element name="SampleStartDate" type="xsd:date" minOccurs="0"/>
                <xsd:element name="SampleStartTime" type="xsd:time" minOccurs="0"/>
                <xsd:element name="SampleEndDate" type="xsd:date" minOccurs="0"/>
                <xsd:element name="SampleEndTime" type="xsd:time" minOccurs="0"/>
                <xsd:element name="FacilitySiteIdentifier" type="xsd:string" minOccurs="0"/>
                <xsd:element name="MonitoringLocationIdentifier" type="xsd:string" minOccurs="0"/>
                <xsd:element ref="crm:SampleProperties" minOccurs="0"/>
                <xsd:element ref="crm:Results" minOccurs="0"/>
        </xsd:sequence>
</xsd:complexType>
<!-- SAMPLE PROPERTIES -->
<xsd:element name="SampleProperties" type="crm:SamplePropertiesDataType">
        <xsd:annotation>
                <xsd:documentation>A listing of sample properties.</xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SamplePropertiesDataType">
        <xsd:sequence>
                <xsd:element ref="crm:SampleProperty" minOccurs="0" maxOccurs="unbounded"/>
        </xsd:sequence>
</xsd:complexType>
<xsd:element name="SampleProperty" type="crm:SamplePropertyDataType">
        <xsd:annotation>
                <xsd:documentation>An sample property.</xsd:documentation>
        </xsd:annotation>
</xsd:element>
<xsd:complexType name="SamplePropertyDataType">
        <xsd:attribute name="name" type="xsd:string" use="required"/>
        <xsd:attribute name="value" type="xsd:string" use="optional"/>
</xsd:complexType>
<!-- RESULTS -->
<xsd:element name="Results" type="crm:ResultsDataType">
        <xsd:annotation>
                <xsd:documentation>A listing of results.</xsd:documentation>
        </xsd:annotation>
</xsd:element>
```

```
<xsd:complexType name="ResultsDataType">
	<xsd:sequence>
		<xsd:element ref="crm:Result" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="Result" type="crm:ResultDataType">
	<xsd:annotation>
		<xsd:documentation>A result.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="ResultDataType">
	<xsd:sequence>
		<xsd:element name="SubstanceIdentifier" type="xsd:string"/>
		<xsd:element name="SubstanceName" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ResultMeasureValue" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ResultMeasureUnitCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ResultBasisName" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ResultStatusCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="StatisticalBaseCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="DetectionLimitMeasure" type="xsd:string" minOccurs="0"/>
		<xsd:element name="DetectionLimitUnitCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ReportingLimitMeasure" type="xsd:string" minOccurs="0"/>
		<xsd:element name="ReportingLimitUnitCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="AnalyticalMethodName" type="xsd:string" minOccurs="0"/>
		<xsd:element ref="crm:ResultProperties" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<!-- ENV INC PROPERTIES -->
<xsd:element name="ResultProperties" type="crm:ResultPropertiesDataType">
	<xsd:annotation>
		<xsd:documentation>A listing of result properties.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="ResultPropertiesDataType">
	<xsd:sequence>
		<xsd:element ref="crm:ResultProperty" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="ResultProperty" type="crm:ResultPropertyDataType">
	<xsd:annotation>
		<xsd:documentation>A result property.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="ResultPropertyDataType">
	<xsd:attribute name="name" type="xsd:string" use="required"/>
	<xsd:attribute name="value" type="xsd:string" use="optional"/>
</xsd:complexType>
<!-- ENV INCIDENTS -->
<xsd:element name="EnvironmentalIncidents" type="crm:EnvironmentalIncidentsDataType">
	<xsd:annotation>
		<xsd:documentation>A listing of environmental incidents.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="EnvironmentalIncidentsDataType">
	<xsd:sequence>
		<xsd:element ref="crm:EnvironmentalIncident" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="EnvironmentalIncident" type="crm:EnvironmentalIncidentDataType">
	<xsd:annotation>
		<xsd:documentation>An environmental incident.</xsd:documentation>
	</xsd:annotation>
```

```
</xsd:element>
<xsd:complexType name="EnvironmentalIncidentDataType">
	<xsd:sequence>
		<xsd:element name="IncidentIdentifier" type="xsd:string"/>
		<xsd:element name="IncidentCategory" type="xsd:string" minOccurs="0"/>
		<xsd:element name="IncidentType" type="xsd:string" minOccurs="0"/>
		<xsd:element name="IncidentDescription" type="xsd:string" minOccurs="0"/>
		<xsd:element name="IncidentOccurredDate" type="xsd:date" minOccurs="0"/>
		<xsd:element name="IncidentReportedDate" type="xsd:date" minOccurs="0"/>
		<xsd:element name="IncidentStatus" type="xsd:string" minOccurs="0"/>
		<xsd:element name="GeographicLocationIdentifier" type="xsd:string" minOccurs="0"/>
		<xsd:element ref="crm:EnvironmentalIncidentProperties" minOccurs="0"/>
	</xsd:sequence>
</xsd:complexType>
<!-- ENV INC PROPERTIES -->
<xsd:element name="EnvironmentalIncidentProperties" type="crm:EnvironmentalIncidentPropertiesDataType">
	<xsd:annotation>
		<xsd:documentation>A listing of environmental incident properties.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="EnvironmentalIncidentPropertiesDataType">
	<xsd:sequence>
		<xsd:element ref="crm:EnvironmentalIncidentProperty" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="EnvironmentalIncidentProperty" type="crm:EnvironmentalIncidentPropertyDataType">
	<xsd:annotation>
		<xsd:documentation>An environmental incident property.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="EnvironmentalIncidentPropertyDataType">
	<xsd:attribute name="name" type="xsd:string" use="required"/>
	<xsd:attribute name="value" type="xsd:string" use="optional"/>
</xsd:complexType>
<!-- GEOGRAPHIC LOCATIONS -->
<xsd:element name="GeographicLocations" type="crm:GeographicLocationsDataType">
	<xsd:annotation>
		<xsd:documentation>A listing of geographic locations.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="GeographicLocationsDataType">
	<xsd:sequence>
		<xsd:element ref="crm:GeographicLocation" minOccurs="0" maxOccurs="unbounded"/>
	</xsd:sequence>
</xsd:complexType>
<xsd:element name="GeographicLocation" type="crm:GeographicLocationDataType">
	<xsd:annotation>
		<xsd:documentation>An geographic location.</xsd:documentation>
	</xsd:annotation>
</xsd:element>
<xsd:complexType name="GeographicLocationDataType">
	<xsd:sequence>
		<xsd:element name="GeographicLocationIdentifier" type="xsd:string"/>
		<xsd:element name="LatitudeMeasure" type="xsd:string" minOccurs="0"/>
		<xsd:element name="LongitudeMeasure" type="xsd:string" minOccurs="0"/>
		<xsd:element name="AddressText" type="xsd:string" minOccurs="0"/>
		<xsd:element name="SupplementalAddressText" type="xsd:string" minOccurs="0"/>
		<xsd:element name="LocalityName" type="xsd:string" minOccurs="0"/>
		<xsd:element name="StateCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="AddressPostalCode" type="xsd:string" minOccurs="0"/>
		<xsd:element name="CountryIdentity" type="xsd:string" minOccurs="0"/>
		<xsd:element name="CountyIdentity" type="xsd:string" minOccurs="0"/>
```

```
                <xsd:element name="TribalLandName" type="xsd:string" minOccurs="0"/>
                <xsd:element name="TribalLandIndicator" type="xsd:string" minOccurs="0"/>
                <xsd:element name="LocationDescriptionText" type="xsd:string" minOccurs="0"/>
                <xsd:element name="SourceMapScaleNumber" type="xsd:string" minOccurs="0"/>
                <xsd:element name="HorizontalAccuracyMeasure" type="xsd:string" minOccurs="0"/>
                <xsd:element name="HorizontalCollectionMethod" type="xsd:string" minOccurs="0"/>
                <xsd:element name="GeographicReferencePoint" type="xsd:string" minOccurs="0"/>
                <xsd:element name="HorizontalReferenceDatum" type="xsd:string" minOccurs="0"/>
                <xsd:element name="DataCollectionDate" type="xsd:date" minOccurs="0"/>
                <xsd:element name="VerticalMeasure" type="xsd:string" minOccurs="0"/>
                <xsd:element name="VerticalCollectionMethod" type="xsd:string" minOccurs="0"/>
                <xsd:element name="VerticalReferenceDatum" type="xsd:string" minOccurs="0"/>
                <xsd:element name="VerificationMethod" type="xsd:string" minOccurs="0"/>
                <xsd:element name="CoordinateDataSource" type="xsd:string" minOccurs="0"/>
                <xsd:element name="GeometricType" type="xsd:string" minOccurs="0"/>
            </xsd:sequence>
        </xsd:complexType>
</xsd:schema>
```