MITS Report API

Introduction

This document describes how to use the MITS Report Web Service API to programmatically perform the following actions from outside of the MITS Report user interface:

- Import a previously exported Report Source Definition XML File
- Create a Report Library
- Delete a Report
- List Report Sources
- List Report Libraries
- List Reports in a Library
- List Available Columns in a Report Source
- Update Report Sources (Dynamic Change Update or Full Update)
- List Users
- Create a New User
- Modify Security Settings for a User

Conventions

Each section of this document will be presented in the following manner:

1. Description
2. Required Information
3. URL
4. Request Example
5. Response Example

This document utilizes different typefaces for different purposes:

- Program code, commands, and XML examples appear in a `monospaced font`
- Placeholders (words or characters that should be replaced in program code, commands, and examples) appear in an *`italicized monospace font`*.

Using the MITS Report API

Posting Method

Report Source API requests are are sent via HTTP (using the POST method) to the *webServices/* subdirectory of the MITS Report Server. For example, if MITS Report was installed on the local machine with a port setting of 8080, and the user was looking to import a Report Source, the url would look like this:

http://localhost:8080/webServices/importExportServices

Authentication Requirements

A MITS Report User account will be needed to use this API. Web Service functionality is dependent on User account security settings. (Creating a Report Library will require certain permissions, etc.)

NOTE: More information on security settings can be found in the *MITS Report Administration Guide* and the *MITS Report Source Reference Guide*.

Authentication is performed using the base64 encoding provided by *HTTP Basic Authentication*. The following line will need to be included in the header of each request:

```
Authorization: Basic username:password
```

For example: `bdonis:ilikemits`

Encoded: `YmRvbmlzOmlsaWtlbWl0cw==`

Import Report Source Definition

Description

Imports a Report Source that has been exported from within MITS Report.

Required Information

A Report Source exported from within MITS Report. It will be in an XML file format.

URL

```
http://{pathToMitsReport}/webServices/importExportServices
```

Request Example

Use the Report Source Definition XML file, exported from MITS Report.

Response Example

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

Create Report Library

Description

Creates new Report Libraries for use in MITS Report

Required Information

URL

```
http://pathToMitsReport/webServices/reportLibraryServices
```

Request Example

```
<createReportLibrary>
  <label>Managers</label>
  <description>Managers Report Library</description>
</createReportLibrary>
```

Response Example

```
<library>
  <name>RptLib_1195069334890</name>
  <id>6</id>
  <reports/>
</library>
```

Delete Report

Description

Remove a single report from MITS Report.

Required Information

You will need the report ID of the report you want to delete. This is obtained from within MITS Report by selecting the desired report. The report ID is located at the end of the URL. Here is an example:

```
http://localhost:8080/displayReport.html?REPORT_ID=24
```

URL

```
http://pathToMitsReport/webServices/reportServices
```

Request Example

```
<deleteMitsReport>
  <reportId>202</reportId>
</deleteMitsReport>
```

Response Example

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

List Report Sources (MultiValue)

Description

Lists all Report Sources created from a particular source file in the MultiValue source database account along with their internal IDs. The Report Source IDs are used for updating Report Sources.

WARNING: The returned list may be influenced by User security settings.

Required Information

- Absolute path to the database account where the source database file is located
- Name of the source file for the Report Source

URL

```
http://pathToMitsReport/webServices/reportSourceService
```

Request Example

```
<listReportSources>
  <listFilter>
    <sourceAccount>C:\IBM\UV\MITS.REPORT</sourceAccount>
    <fileName>INVOICES</fileName>
  </listFilter>
</listReportSources>
```

Response Example

```
<mitsReportSources>
  <mitsReportSource>
    <mitsReportSourceInternalId>42</mitsReportSourceInternalId>
    <sourceAccount>C:\IBM\UV\MITS.REPORT</sourceAccount>
    <fileName>INVOICES</fileName>
    <label>Invoice Small</label>
    <description>This has a small sample.</description>
    <currentStatus>Available</currentStatus>
  </mistReportSource>
</mitsReportSources>
```

List Report Sources (Relational)

Description

Lists all Report Sources created from a particular source file in the relational source database account along with their internal IDs. The Report Source IDs are used for updating Report Sources.

WARNING: The returned list may be influenced by User security settings.

Required Information

- Absolute path to the database account where the source database file is located
- Name of the source file for the Report Source

URL

```
http://pathToMitsReport/webServices/relationalSourceService
```

Request Example

```
<listReportSources>
  <listFilter>
    <sourceAccount>C:\IBM\UV\MITS.REPORT</sourceAccount>
  </listFilter>
</listReportSources>
```

Response Example

```
<mitsReportSources>
  <mitsReportSource>
    <mitsReportSourceInternalId>42</mitsReportSourceInternalId>
    <sourceAccount>C:\IBM\UV\MITS.REPORT</sourceAccount>
    <tableName>INVOICES</tableName>
    <label>Invoice Small</label>
    <description>This has a small sample.</description>
  </mistReportSource>
</mitsReportSources>
```

List Report Libraries

Description

Lists all Report Libraries along with their internal IDs.

WARNING: The returned list may be influenced by User security settings.

Required Information

None.

URL

```
http://{pathToMitsReport}/webServices/reportLibraryServices
```

Request Example

```
<listReportLibraries/>
```

Response Example

```
<reportLibraries>
  <library>
    <name>Common Report Library</name>
    <id>2</id>
    <reports/>
  </library>
  <library>
    <name>Invoices Small</name>
    <id>3</id>
    <reports/>
  </library>
</reportLibraries>
```

List Reports in a Library

Description

Lists all Reports or only those created within a particular Report Library along with their internal IDs.

WARNING: The returned list may be influenced by User security settings.

Required Information

The ID of the Report Library

URL

```
http://pathToMitsReport/webServices/reportLibraryServices
```

Request Example 1

Returns all reports in every Report Library

```
<listReportsInLibrary/>
```

Response Example 1

```
<reportLibraries>
  <library>
    <name>Common Report Library</name>
    <id>2</id>
    <reports>
      <report>
        <id>22</id>
        <name>RELATIONAL REPORT</name>
        <sourceId>47</sourceId>
        <sourceName>Invoices2</sourceName>
      </report>
    </reports>
  </library>
  <library>
    <name>Invoices Small</name>
    <id>3</id>
    <reports>
      <report>
        <id>28</id>
        <name>Invoices Report</name>
        <sourceId>44</sourceId>
        <sourceName>Invoices</sourceName>
      </report>
    </reports>
  </library>
</reportLibraries>
```

Request Example 2

Returns all reports from a particular Report Library

```
<listReportsInLibrary>
  <libraryId>2</libraryId>
</listReportsInLibrary>
```

Response Example 2

```
<reportLibraries>
  <library>
    <name>Common Report Library</name>
    <id>2</id>
    <reports>
      <report>
        <id>22</id>
        <name>RELATIONAL REPORT</name>
        <sourceId>47</sourceId>
        <sourceName>Invoices2</sourceName>
      </report>
      <report>
        <id>28</id>
        <name>Invoices Report</name>
        <sourceId>44</sourceId>
        <sourceName>Invoices</sourceName>
      </report>
    </reports>
  </library>
</reportLibraries>
```

List Columns

Description

This will list all columns associated with a Report Source. The column information returned can be used when performing User security updates.

WARNING: The returned list may be influenced by User security settings.

Information Required

Internal ID for a Report Source.

URL

```
http://pathToMitsReport/webServices/reportSourceService
```

Request Example

```
<listReportSourceColumns>
  <mitsReportSourceInternalId>1675</mitsReportSourceInternalId>
</listReportSourceColumns>
```

Response Example

```
<mitsReportSourceColumns>
  <mitsReportSourceColumn>
    <title>Extended Cost</title>
    <dictId>EXTENDED.COST</dictId>
    <securityId>2964</securityId>
  </mitsReportSourceColumn>
  <mitsReportSourceColumn>
    <title>Extended Sales</title>
    <dictId>EXTENDED.SALES</dictId>
    <securityId>2965</securityId>
  </mitsReportSourceColumn>
</mitsReportSourceColumns>
```

Report Source Update

Full Report Source Update

Description

This will perform a full update on a Report Source.

WARNING: User security settings may prevent a Report Source from updating.

NOTE: This will clear all data from the Report Source before updating. Some Report Sources can take a long time to update or can cause significant system load. Be sure you are aware of the potential impact before initiating an update of this kind.

Information Required

Internal ID for a Report Source.

URL

```
http://pathToMitsReport/webServices/reportSourceService
```

Request Example

```
<updateReportSource>
  <mitsReportSourceInternalId>44</mitsReportSourceInternalId>
</updateReportSource>
```

Response Example

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

Dynamic Change Update

Description

This will update a Report Source using the *Dynamic Change Update* feature. See the *MITS Report Source Reference Guide* for more information about Dynamic Change Update.

WARNING: User security settings may prevent a Report Source from updating.

Information Required

Internal ID for a Report Source.

URL

```
http://pathToMitsReport/webServices/reportSourceService
```

Request Example

```
<changeDataCaptureUpdateReportSource>
  <mitsReportSourceInternalId>44</mitsReportSourceInternalId>
</changeDataCaptureUpdateReportSource>
```

Response Example

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

List Users

Description

This will return all MITS Report Users as well as their personal information and associated Groups, as defined on the **Administration** tab of MITS Report.

Required Information

None

URL

```
http://pathToMitsReport/webServices/userAndGroupSecurity
```

Request Example

```
<listUsers />
```

Response Example

```
<users>
  <user>
    <reportUserId>7</reportUserId>
    <emailAddress>bobd@mits.com</emailAddress>
    <firstName>Bob</firstName>
    <lastName>Donis</lastName>
    <password>*********</lastName>
    <username>bdonis</username>
    <roles class="list">
      <role>
        <roleID>6</roleID>
        <systemRole>true</systemRole>
        <name>Report Library Reader</name>
      </role>
      <role>
        <roleID>4</roleID>
        <systemRole>true</systemRole>
        <name>Report Source Reader</name>
      </role>
    </roles>
    <enabled>true</enabled>
  </user>
</users>
```

Create a New User

Description

This will create a new MITS Report User.

Required Information

MITS Report login credentials for the User (username and password)

URL

```
http://pathToMitsReport/webServices/userAndGroupSecurity
```

Request Example

```
<userSecurityUpdate>
  <users>
    <user>
      <username>bdonis</username>
      <password>ilikemits</password>
      <firstName>Bob</firstName>
      <lastName>Donis</lastName>
      <emailAddress>bobd@mits.com</emailAddress>
      <roles>
        <roleId>4</roleId>
      </roles>
      <sources>
        <source>
          <sourceId>44</sourceId>
          <sourceRestriction>
            <access>true</access>
            <configuration>false</configuration>
            <updateData>false</updateData>
          </sourceRestriction>
          <columnAccessRestrictions>
            <columnAccessRestriction>1691</columnAccessRestriction>
          </columnAccessRestrictions>
          <columnValueRestrictions>
            <booleanOperator>AND</booloeanOperator>
            <columnValueRestriction>
              <booleanOperator>AND</booleanOperator>
              <columnId>1677</columnId>
              <valueRestriction>
                <expression>EQUAL</expression>
                <value>BONIFACE BINGO</value>
              </valueRestriction>
            </columnValueRestriction>
          <columnValueRestrictions>
        </source>
      </sources>
    </user>
  </users>
</userSecurityUpdate>
```

Response Example (Success)

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

Response Example (Failure)

```
<failureResponse>
  <failureReason>unknown</failureReason>
  <errorMessage>java.net.SocketException: Read timed out</errorMessage>
</failureResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

Modifying a Report User

Description

This will update an existing MITS Report User.

Required Information

MITS Report login credentials for the User (username and password)

URL

```
http://pathToMitsReport/webServices/userAndGroupSecurity
```

Request Example

```
<userSecurityUpdate>
  <users>
    <user>
      <username>bdonis</username>
      <password>ilikemits</password>
      <firstName></firstName>
      <lastName></lastName>
      <emailAddress></emailAddress>
      <roles>
        <roleId>5</roleId>
        <roleId>3</roleId>
      </roles>
      <sources>
        <source>
          <sourceId>44</sourceId>
          <sourceRestriction>
            <access>true</access>
            <configuration>false</configuration>
            <updateData>false</updateData>
          </sourceRestriction>
          <columnAccessRestrictions>
            <columnAccessRestriction>1691</columnAccessRestriction>
          </columnAccessRestrictions>
          <columnValueRestrictions>
            <booleanOperator>AND</booloeanOperator>
            <columnValueRestriction>
              <booleanOperator>AND</booleanOperator>
              <columnId>1677</columnId>
              <valueRestriction>
                <expression>EQUAL</expression>
                <value>BONIFACE BINGO</value>
              </valueRestriction>
            </columnValueRestriction>
          <columnValueRestrictions>
        </source>
      </sources>
    </user>
  </users>
</userSecurityUpdate>
```

Response Example (Success)

```
<successResponse>
  <responseTime>2007-06-22 09:45:32.5</responseTime>
</successResponse>
```

Response Example (Failure)

```
<failureResponse>
  <failureReason>unknown</failureReason>
  <errorMessage>java.net.SocketException: Read timed out</errorMessage>
</failureResponse>
```

NOTE: A success response does not refer to the operation succesfully completing, but that the request for the operation was successfully received.

User Security API Schema Reference

```
<?xml version="1.0" encoding="UTF-8"?>

<!--
  XML Schema for User Security API Calls
-->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">


  <!--Requests-->
  <!--
      Request xml for updating/creating users and permissions
  -->
  <xs:element name="userSecurityUpdate">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="users">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="user" type="userType" minOccurs="1" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <!--
      Request xml to retrieve a list of Report Users (without the permissions)
  -->
  <xs:element name="listUsers" type="xs:element" />


  <!--Response Types-->
  <!--
  Sucess Response for an update
  -->
```

```
<xs:element name="successResponse" type="xs:string" />


<!--
Failure response for an update
-->
<xs:element name="failureResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="failureReason" type="failureReasonEnumType"/>
      <xs:element name="errorMessage" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>


<!--Referenced Types-->
<xs:complexType name="userType">
  <xs:sequence>
    <xs:element name="roles" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="role" type="xs:integer" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="sources" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="source" type="sourceType" minOccurs="1" maxOccurs="unbounded"/>
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="username" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="password" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="firstName" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="lastName" type="xs:string" minOccurs="1" maxOccurs="1" />
    <xs:element name="emailAddress" type="xs:string" minOccurs="1" maxOccurs="1" />
```

```
  </xs:sequence>
</xs:complexType>

<xs:complexType name="sourceType">
  <xs:sequence>
    <xs:element name="sourceRestriction" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="access" type="xs:boolean" minOccurs="1" maxOccurs="1" />
          <xs:element name="configure" type="xs:boolean" minOccurs="1" maxOccurs="1" />
          <xs:element name="updateData" type="xs:boolean" minOccurs="1" maxOccurs="1" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="columnAccessRestrictions" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="columnAccessRestriction" minOccurs="1" maxOccurs="unbounded" type="xs:integer" />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="columnValueRestrictions" minOccurs="0" maxOccurs="1">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="columnValueRestriction" type="columnValueRestrictionType" minOccurs="1"
             maxOccurs="unbounded" />
          <xs:element name="booleanOperator" type="booleanOperatorEnumType" minOccurs="1" maxOccurs="1"  />
        </xs:sequence>
      </xs:complexType>
    </xs:element>

    <xs:element name="sourceId" type="xs:integer" minOccurs="1" maxOccurs="1" />
  </xs:sequence>
</xs:complexType>
```

```
<xs:complexType name="columnValueRestrictionType">
  <xs:sequence>
    <xs:element name="valueRestriction" minOccurs="1" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <xs:element name="expression" type="expressionEnumType" minOccurs="1" maxOccurs="1"  />
          <xs:element name="value" type="xs:string" minOccurs="1" maxOccurs="1"  />
        </xs:sequence>
      </xs:complexType>
    </xs:element>
    <xs:element name="booleanOperator" type="booleanOperatorEnumType" minOccurs="1" maxOccurs="1"  />
    <xs:element name="columnId" type="xs:integer" minOccurs="1" maxOccurs="1"  />
  </xs:sequence>
</xs:complexType>

<xs:simpleType name="booleanOperatorEnumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AND"/>
    <xs:enumeration value="OR"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="expressionEnumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQUAL"/>
    <xs:enumeration value="NOT_EQUAL"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="failureReasonEnumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="MitsReportSourceNotFound"/>
    <xs:enumeration value="ColumnNotFound"/>
    <xs:enumeration value="RoleNotFound"/>

    <!-- Occurs when adding a user that already exists -->
```

```
      <xs:enumeration value="UserAlreadyExists"/>

      <!-- Occurs when updating a user that does not already exist -->
      <xs:enumeration value="UserNotFound"/>

      <!-- Occurs when adding a Text value restriction to a Number Column -->
      <xs:enumeration value="MismatchedValueRestrictionType"/>
      <xs:enumeration value="Unknown"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

Report Source API Schema Reference

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <!--published methods-->


  <!--
  Depending on the data types given listReportSources will return a list of report sources.
  Data types: sourceAccount and/or fileName. Types are defined by listFilterType.
  -->
  <xs:element name="listReportSources">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="listFilter" type="listFilterType" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--
  Given an id of a report source this method will return a list of the Report Source's columns.
  -->
  <xs:element name="listReportSourceColumns">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="mitsReportSourceInternalId" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!--
  Given an id of a report source this method will return a mitsReportSourceType.
  -->
  <xs:element name="getReportSource">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="mitsReportSourceInternalId" type="xs:integer"/>
      </xs:sequence>
```

```
  </xs:complexType>
</xs:element>
<!--
Updates a report source based on its id.
-->
<xs:element name="updateReportSource">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="mitsReportSourceInternalId" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--return types-->
<xs:element name="mitsReportSource" type="mitsReportSourceType"/>

<!--
    A list of Report Sources
-->
<xs:element name="mitsReportSources">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="mitsReportSource" type="mitsReportSourceType" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
    A list of Columns in a Report Source
-->
<xs:element name="mitsReportSourceColumns">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="mitsReportSourceColumn" type="mitsReportSourceColumnType" minOccurs="0"
         maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

```
<!--
Sucess Response for an update
-->
<xs:element name="successResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="responseTime" type="xs:dateTime" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<!--
Failure response for an update
-->
<xs:element name="failureResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="failureReason" type="failureReasonEnumType"/>
      <xs:element name="errorMessage" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

<!--data types-->
<xs:complexType name="listFilterType">
  <xs:sequence>
    <xs:element name="sourceAccount" type="xs:string" minOccurs="0"/>
    <xs:element name="fileName" type="xs:string" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>

<!--
This defines what a mits report source that will be returned by the published retrieval methods.
-->
<xs:complexType name="mitsReportSourceType">
  <xs:sequence>
    <xs:element name="mitsReportSourceInternalId" type="xs:integer"/>
```

```
    <xs:element name="sourceAccount" type="xs:string"/>
    <xs:element name="fileName" type="xs:string"/>
    <xs:element name="label" type="xs:string"/>
    <xs:element name="description" type="xs:string"/>
    <xs:element name="currentStatus" type="currentStatusEnumType"/>
  </xs:sequence>
</xs:complexType>

<!--
Defines a MitsReportSourceColumn that will be returned by the published retrieval methods
-->
<xs:complexType name="mitsReportSourceColumnType">
  <xs:sequence>
    <!-- The title of the Column (As displayed in the Reports) -->
    <xs:element name="title" type="xs:string"/>
    <!--The Dictionary Item's ID-->
    <xs:element name="dictId" type="xs:string"/>
    <!--The Security ID to use when setting permissions-->
    <xs:element name="securityId" type="xs:integer"/>
  </xs:sequence>
</xs:complexType>

<!--
The current status of a report source.
-->
<xs:simpleType name="currentStatusEnumType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="Available"/>
    <xs:enumeration value="Updating"/>
    <xs:enumeration value="Failed"/>
    <xs:enumeration value="Unavailable"/>
  </xs:restriction>
</xs:simpleType>

<xs:simpleType name="failureReasonEnumType">
  <xs:restriction base="xs:string">
```

```
      <xs:enumeration value="MitsReportSourceNotFound"/>
      <xs:enumeration value="Unknown"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">


 <!-- Either the response from, or the request to the Import/Export services -->

  <xs:element name="list">

    <xs:complexType>

      <xs:sequence>

        <!-- NOTE: When sending a request, only one or the other of these tags

             can be included.  You can only import/export multivalue OR

             relational, one at a time. -->

        <xs:element name="multiValueReporSource" type="multiValueReportSource" minOccurs="0"
maxOccurs="unbounded"/>

        <xs:element name="relationalReportSource" type="relationalReportSource" minOccurs="0"
maxOccurs="unbounded"/>

      </xs:sequence>

    </xs:complexType>

  </xs:element>


  <!-- Multivalue Report Source Type -->

  <!-- Use this tag when importing a MultiValue Connection based Report

       Source -->


    <xs:complexType name="multiValueReportSource">

      <xs:sequence>

        <xs:element name="sourceAccountName" type="xs:string"/>

        <xs:element name="sourceFileName" type="xs:string"/>

        <xs:element name="selectFilter" type="xs:string"/>

        <xs:element name="dictionaryItems" type="dictionaryItems"/>

        <xs:element name="filters" type="sourceFilters"/>

        <xs:element name="databaseType" type="databaseTypeEnum"/>

        <xs:element name="filterRelationship" type="filterRelationshipEnum"/>

        <xs:element name="label" type="xs:string"/>

        <xs:element name="description" type="xs:string"/>

        <xs:element name="columns" type="columns"/>
```

```
    </xs:sequence>
  </xs:complexType>



  <xs:complexType name="dictionaryItems">
    <xs:sequence>
      <xs:element name="dictionaryItem" type="dictionaryItem"/>
    </xs:sequence>
  </xs:complexType>



  <xs:complexType name="columns">
    <xs:sequence>
      <xs:element name="column" type="column"/>
    </xs:sequence>
  </xs:complexType>


<!-- Metadata regarding the individual Dictionary Items which describe the
     columns in the Report Source -->

  <xs:complexType name="dictionaryItem">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="dictionaryItemName" type="xs:string"/>
      <xs:element name="dictionaryAttributes" type="dictionaryAttributes"/>
      <xs:element name="column" type="xs:string"/>
      <xs:element name="expression" type="xs:string"/>
      <xs:element name="fieldType" type="fieldTypeEnum"/>
      <xs:element name="scalingFactor" type="xs:integer"/>
      <xs:element name="updateBeforeExtract" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
```

```
<!-- The DI Attributes that are stored -->

  <xs:complexType name="dictionaryAttributes">
    <xs:sequence>
      <xs:element name="attributeNumber" type="xs:integer"/>
      <xs:element name="value" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>



  <xs:complexType name="sourceFilters">
    <xs:sequence>
      <xs:element name="sourceFilter" type="sourceFilter"/>
    </xs:sequence>
  </xs:complexType>


<!-- Filters out invalid values from the Report Source's incoming data -->

  <xs:complexType name="sourceFilter">
    <xs:sequence>
      <xs:element name="dictionaryItemName" type="xs:string" minOccurs="0" maxOccurs="1"/>
      <xs:element name="value" type="xs:string"/>
      <xs:element name="expression" type="expressionEnum"/>
    </xs:sequence>
  </xs:complexType>


<!-- Metadata about the configuration of the Column in the Report Server -->

  <xs:complexType name="column">
    <xs:sequence>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
```

```
      <xs:element name="length" type="xs:integer"/>
      <xs:element name="scale" type="xs:integer"/>
      <xs:element name="precision" type="xs:integer"/>
      <xs:element name="sortOrder" type="xs:integer"/>
      <xs:element name="indexed" type="xs:boolean"/>
      <xs:element name="inChildTable" type="xs:boolean"/>
      <xs:element name="reportDataType" type="reportDataTypeEnum"/>
    </xs:sequence>
  </xs:complexType>


<!-- These are available options for Multivalue Types.
     - A Scalar value is not multivalue, meaning that the field only contains
       a single value per record.

     - Dependent Multivalues will only display at most the same number of
       values as the Controlling Multivalue field for that record.

     - Independent Multivalues contain multiple values, but the values are
       not directly related to the Controlling Multivalue field's value (an
       example is a Comment field)

     - There can be only one Controlling Multivalue field per Report Source.
       This field specifies the maximum number of items per record.
-->
<xs:simpleType name="fieldTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="SCALAR"/>
    <xs:enumeration value="DEPENDENT_MV"/>
    <xs:enumeration value="INDEPENDENT_MV"/>
    <xs:enumeration value="CONTROLLING_MV"/>
  </xs:restriction>
</xs:simpleType>

<!-- The different types of expressions available for Filters -->
<xs:simpleType name="expressionEnum">
```

```
  <xs:restriction base="xs:string">
    <xs:enumeration value="EQUAL"/>
    <xs:enumeration value="NOT_EQUAL"/>
    <xs:enumeration value="LESS_THAN"/>
    <xs:enumeration value="GREATER_THAN"/>
    <xs:enumeration value="LESS_THAN_OR_EQUAL"/>
    <xs:enumeration value="GREATER_THAN_OR_EQUAL"/>
  </xs:restriction>
</xs:simpleType>

<!-- Indicates the type of MV Database this import/export originated from -->
<xs:simpleType name="databaseTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="UNIVERSE"/>
    <xs:enumeration value="UNIDATA"/>
  </xs:restriction>
</xs:simpleType>

<!-- The relationship between multiple filters in the same filter group
     or across multiple filter groups -->
<xs:simpleType name="filterRelationshipEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="AND"/>
    <xs:enumeration value="OR"/>
  </xs:restriction>
</xs:simpleType>

<!-- The type of data that is contained within the Column.
     - Decimals are used for any number that contains numerics past the decimal point
     - Integers are used for any number (decimal values will be truncated)
     - String values are text or numeric values (will be sorted lexiconigraphically)
     - Date values are self explanatory
 -->
<xs:simpleType name="reportDataTypeEnum">
  <xs:restriction base="xs:string">
    <xs:enumeration value="DECIMAL"/>
```

```
    <xs:enumeration value="STRING"/>
    <xs:enumeration value="INTEGER"/>
    <xs:enumeration value="DATE"/>
  </xs:restriction>
</xs:simpleType>

<!-- Used for SQL based Relational database Sources -->

  <xs:complexType name="relationalReportSource">
    <xs:sequence>
      <xs:element name="connection" type="connection"/>
      <xs:element name="sourceTable" type="sourceTable"/>
      <xs:element name="sourceColumns" type="sourceColumns"/>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="description" type="xs:string"/>
      <xs:element name="columns" type="columns"/>
    </xs:sequence>
  </xs:complexType>



<!-- Metadata for the connection.
     NOTE: The connection MUST already exist on the Report system to import
     using the webservice! -->

  <xs:complexType name="connection">
    <xs:sequence>
      <xs:element name="driverClass" type="xs:string"/>
      <xs:element name="label" type="xs:string"/>
      <xs:element name="path" type="xs:string"/>
      <xs:element name="connectionInPool" type="xs:integer"/>
    </xs:sequence>
  </xs:complexType>



<!-- The table from whence the Source data will be retrieved from -->
```

```
  <xs:complexType name="sourceTable">
    <xs:sequence>
      <!-- The SQL element is used only if you wish to specify your own SQL
           statement to generate the Report Source.  Once created using this
           you cannot delete or add columns.

           NOTE: This is mutually exclusive with the other tags. -->
      <xs:element name="sql" type="xs:string" minOccurs="0" maxOccurs="1"/>

      <!-- These tags describe a specific table/view/etc. to retrieve the
           Report Source data from.

           NOTE: These tags are mutually exclusive with the sql tag. -->
      <xs:element name="catalog" type="xs:string"/>
      <xs:element name="schema" type="xs:string"/>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="type" type="databaseTypeEnum"/>
    </xs:sequence>
  </xs:complexType>



  <xs:complexType name="sourceColumns">
    <xs:sequence>
      <xs:element name="sourceColumn" type="sourceColumn"/>
    </xs:sequence>
  </xs:complexType>


<!-- A column in the Source table that will be used as a Column in the
     Report Source -->

  <xs:complexType name="sourceColumn">
    <xs:sequence>
      <!-- The label of the Column to use in the Report Source -->
      <xs:element name="destinationColumn" type="xs:string"/>
```

```
        <!-- The name of the column in the Source Table to use -->
        <xs:element name="sourceName" type="xs:string"/>
        <xs:element name="scale" type="xs:integer"/>
        <xs:element name="precision" type="xs:integer"/>

        <!-- This is the Type as specified by the SQL database -->
        <xs:element name="type" type="xs:integer"/>
        <xs:element name="filters" type="relationalSourceFilter"/>
      </xs:sequence>
    </xs:complexType>


  <!-- Data Filters applied to incoming data -->

    <xs:complexType name="relationalSourceFilter">
      <xs:sequence>
        <xs:element name="relationship" type="filterRelationshipEnum"/>
        <xs:element name="filter" type="sourceFilter"/>
      </xs:sequence>
    </xs:complexType>


</xs:schema>
```

Report Library API Schema Reference

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">


  <!--  Request : Creates a new Report Library  -->
  <xs:element name="createLibrary">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="label" type="xs:string"/>
        <xs:element name="description" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <!--  Request : Lists the reports contained within one Report Library  -->
  <xs:element name="listReportsInLibrary">
    <xs:complexType>
      <xs:sequence>
        <!-- This is the internally used library id.  It can be found using the
             listReportLibraries webservice.


             NOTE: When the libraryId is left empty, the Reports in all Report
             Libraries are returned -->
        <xs:element name="libraryId" type="xs:integer" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <!--  Request : Lists all Report Libraries  -->
  <xs:element name="listReportLibraries"/>


  <!--  Response : One or more libraries  -->
  <xs:element name="reportLibraries">
    <xs:complexType>
```

```
    <xs:sequence>
      <xs:element name="library" type="library"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>


  <xs:complexType name="library">
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="id" type="xs:integer"/>
      <xs:element name="reports" type="reports"/>
    </xs:sequence>
  </xs:complexType>


<!--  Response : Contains information about a particular report  -->
  <xs:complexType name="report">
    <xs:sequence>
      <!-- The internally used id for this Report -->
      <xs:element name="id" type="xs:integer"/>
      <xs:element name="name" type="xs:string"/>
      <!-- The internally used id for this Report's Source -->
      <xs:element name="sourceId" type="xs:integer"/>
      <xs:element name="sourceName" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>


<!--  Response : Contains Reports  -->

  <xs:complexType name="reports">
    <xs:sequence>
      <xs:element name="report" type="report"/>
    </xs:sequence>
  </xs:complexType>
```

```
  <!--  Response : Returns a failure message with an error  -->
  <xs:element name="failureReponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="failureReason" type="failureReasonEnumType"/>
        <!-- The errorMessage is often helpful for identifying exactly why/where
             the failure occurred (most likely due to invalid xml)  -->
        <xs:element name="errorMessage" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

  <!--  Enum Type : Failure Reasons  -->
  <xs:simpleType name="failureReasonEnumType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Unknown"/>
    </xs:restriction>
  </xs:simpleType>

</xs:schema>
```

Delete Report API Schema Reference

```
<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">


  <!--  Request : Deletes an individual report based on the reports ID  -->
  <xs:element name="deleteMitsReport">
    <xs:complexType>
      <xs:sequence>
        <!-- The internal id of the Report you wish to delete (can be found
             using the listReportsInLibrary service) -->
        <xs:element name="reportId" type="xs:integer"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <!--  Response : Returns a success message  -->
  <xs:element name="successResponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="responseTime" type="xs:dateTime" minOccurs="0"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>


  <!--  Response : Returns a failure message with an error  -->
  <xs:element name="failureReponse">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="failureReason" type="failureReasonEnumType"/>
        <!-- The errorMessage is often helpful for identifying exactly why/where
             the failure occurred (most likely due to invalid xml)  -->
        <xs:element name="errorMessage" type="xs:string"/>
      </xs:sequence>
    </xs:complexType>
```

```
  </xs:element>


  <!--  Enum Type : Failure Reasons  -->
  <xs:simpleType name="failureReasonEnumType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="Unknown"/>
    </xs:restriction>
  </xs:simpleType>


</xs:schema>
```