

XML Document Handling

- [XML over HTTP](#)
- [Use an XML Parser](#)
- [HTML in XML](#)

This documents offers tips on handling XML documents.

XML over HTTP

Transferology uses the following de facto standards for sending XML documents over HTTP (or HTTPS):

- Transferology sends the standard HTTP request headers **Content-Type** and **Content-Length**
- The Content-Type is always **"text/xml"**
- The Content-Length will be the actual document size
- By default, Transferology sends the XML document within the HTTP body
- A school can change their Transferology settings so Transferology sends the XML as an HTML form field (i.e., HTTP request parameter and value)
- Transferology uses the **POST** method to send the XML

A data capture of the HTTP POST from Transferology will appear similar to this:

```
POST /uselect-int/document/xml HTTP/1.1
Content-Type: text/xml
Content-Length: 3087
Host: www.someschool.edu:80
Connection: keep-alive

<RequestDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
http://www.transfer.org/xsd/RequestDA_slh.xsd">
  <Envelope>
    <RequestType>CAS</RequestType>
    <TransactionType>Request Degree Audit</TransactionType>
    <DocumentNumber>2009110315255772</DocumentNumber>
    <PartyId>http://test.transfer.org/uselect/service/xml</PartyId>
  </Envelope>
  <RequestDetail>
    <Control>
      <ControlBatchId>2009110315255772</ControlBatchId>
      <ControlRequestServerName>darslx</ControlRequestServerName>
      <ControlRecord>UH</ControlRecord>
      <ControlRequestType>A</ControlRequestType>
    </Control>
    <RequestData>
      <Person>
        <PersonID>
          <PersonIdCodeQualifier>CAS</PersonIdCodeQualifier>
          <PersonIdCode>super@MU</PersonIdCode>
        </PersonID>
        <PersonName>
          <NameType>CAS</NameType>
          <NameFirst>CAS</NameFirst>
          <NameLast>Student</NameLast>
        </PersonName>
        <StudentType>
          <StudentTypeStatus>CAS Student</StudentTypeStatus>
        </StudentType>
      </Person>
      <PersonControl>
        <BatchId>2009110315255772</BatchId>
        <ControlRecord></ControlRecord>
      </PersonControl>
      <DegreeProgram>
        <Institution>
          <Entity>
            <EntityIDCodeQualifier>73</EntityIDCodeQualifier>
            <EntityIDCode>007104</EntityIDCode>
            <EntityIDCodeGroup/>
          </Entity>
```

```

        </Institution>
        <AcademicProgram>
            <ProgramType>Program</ProgramType>
            <ProgramCode>AS59BB117</ProgramCode>
            <ProgramCatalogYear>999999</ProgramCatalogYear>
        </AcademicProgram>
    </DegreeProgram>
    <CourseInstitution>
        <Institution>
            <Entity>
                <EntityIDCodeQualifier>73</EntityIDCodeQualifier>
                <EntityIDCode>003018</EntityIDCode>
                <EntityIDCodeGroup/>
            </Entity>
        </Institution>
        <Course>
            <CourseSubjectAbbreviation>BIOL</CourseSubjectAbbreviation>
            <CourseNumber>101</CourseNumber>
            <CourseTitle>Biology</CourseTitle>
            <CourseSession>200901</CourseSession>
            <CourseCreditValue>03.00</CourseCreditValue>
            <CourseAcademicGrade>A</CourseAcademicGrade>
            <CoursePseudoCode>N</CoursePseudoCode>
            <CourseFlagCodeType>
                <CourseFlagCode/>
            </CourseFlagCodeType>
        </Course>
    </CourseInstitution>
</RequestData>
</RequestDetail>
</RequestDocument>

```

Transferology expects external interfaces to also adhere to these de facto standards.

Use an XML Parser

Transferology sends well-formed XML documents. However, well-formed documents can have an XML version declaration, extra spaces, non-required attributes and elements, empty elements, and name spaces that make it difficult to parse with a home grown parser or, even worse, straight text/string comparisons.



NOTE:

Avoid the temptation to write your own XML parser. Instead, use an existing XML parser.

HTML in XML

Within the context of Transferology, it is normal for an XML document to contain HTML. For the HTML to be correctly transferred between Transferology and an external system, both systems must use and accept the two standard methods of escaping markup within text per XML 1.0 specification, [Section 2.4 Character Data and Markup](#). To summarize, the HTML content must either be wrapped within an XML `<![CDATA[...]]>` section or escape any characters that are one of the five predefined XML entities. The following table identifies the five predefined XML entities with their string escaped representations:

Entity	String Escape
&	&
<	<
>	>
'	'
"	"