External Degree Audit Interface Specification

This page contains information about Transferology and its support of the External Degree Audit (EDAG) Interface.

EDAG High Level Requirements

The custom built EDAG server must satisfy these high level requirements:

- Be able to receive degree audit requests from Transferology as described below in the **Transferology and EDAG Document Exchange** section. Upon receiving the degree audit request, it must reply to Transferology with the standard accept or reject response (e.g. rejected due to validation errors).
- Create an individualized degree audit report for the student's audit request.
- Establish an http/https connection with Transferology and send the student's degree audit report to Transferology. Upon receiving the degree audit report, Transferology will reply with the standard accept or reject response, indicating whether the report was accepted or rejected.

Please see the EDAG FAQ and Transferology Lab FAQs, Audits section for common EDAG questions.

What is EDAG?

Transferology is degree audit-*agnostic* and works with any degree audit system that offers an interface that implements one of the supported PESC version 1.0 Degree Audit request and reply specifications. This interface is characterized by the exchange of XML documents over HTTP (or HTTPS) and is commonly referred to as an External Degree Audit, or EDAG.

Prior EDAG testing and PROD go live, please email the URL to your external degree audit server to: DonD@collegesource.com

Supported PESC Degree Audit request and reply schemas

Transferology supports the following PESC Degree Audit request and reply schemas.

New EDAGs must used the 1.5 schemas, the legacy schemas are provided for legacy interface compatibility ONLY!!!

PESC Degree Audit Request Schema

RequestDA 1.5 - http://www.transfer.org/xsd/RequestDA/RequestDA-1.5.xsd

PESC Degree Audit Reply Schema

ReplyDA 1.5 - http://www.transfer.org/xsd/ReplyDA/ReplyDA-1.5.xsd

See XML Document Handling for tips.

Transferology and EDAG Document Exchange

Transferology's EDAG document exchange process is comprised of the steps shown below:

Step #1 - Transferology builds XML document

Transferology builds the Degree Audit request XML document based on the format defined by the receiving (i.e., target) school's Program Request Code setting in Transferology. See the PESC Degree Audit request schema for the specific elements and attributes within the XML document. Transferology provides a return URL address in the <PartyId> element. Element data values wrapped with double quotes identify static content. Do not ignore the URL specified in the <PartyId> element. Transferology reserves the right to change the return URL, when needed. Your school's EDAG server should be able to return program requests to a different <PartyId> in this situation. If your EDAG is unable to use a URL that is a value from the XML document, please consider using some form of a runtime/deploy property to prevent hard coding the return URL into your interface.

Step #2 - Transferology posts XML request document to EDAG Url

Transferology then attempts to open an HTTP (or HTTPS) connection to the defined EDAG URL. The EDAG URL must specify the correct protocol (HTTP or HTTPS) for Transferology to use. If the connection is successful, Transferology then posts the XML document. After the post, Transferology waits, on the open connection, for an acknowledgment that the EDAG has either accepted or rejected the program request.

EDAG Accept Acknowledgment

To accept the program request, on the open connection, the EDAG must send the following accept acknowledgment XML document on the existing open connection:

```
<response status="0"/>
```

EDAG Reject Acknowledgment

To reject the program request, on the open connection, the EDAG should send the following reject acknowledgment XML document on the existing open connection:

```
<response status="1">descriptive error message</<response>
```

> **NOTE:**
>
> Any non-zero status indicates the program request was rejected.

Transferology considers the request failed for any of the following situations:

- No data is received within 60 seconds
- The received data is not well-formed XML
- The received XML is not a <response> XML document
- The response's status attribute is non-zero

When a request fails, it is scheduled for a retry.

Transferology uses an increasing retry delay (i.e., time is doubled between each failed attempt: 5 minutes, 10 minutes, 20 minutes...), with a maximum delay of 12 hours between retries. Transferology retries the post until successfully sent.

Step #3 - EDAG posts XML reply document to Transferology

Once the EDAG generates the program content, the EDAG must post the program content within a Degree Audit reply XML document back to Transferology. The EDAG opens an HTTP (or HTTPS) connection to the return URL provided earlier by Transferology in the <PartyId> element of the request. The EDAG must use the protocol (HTTP or HTTPS) provided by Transferology in the <PartyId> element. The EDAG should follow the XML over HTTP conventions. If the reply document contains well-formed XML, then Transferology will send the following success acknowledgment XML document on the existing open connection:

```
<response status="0"></response>
```

Once Transferology sends the success acknowledgment, the audit is immediately available for viewing by the Transferology user that requested the program. If Transferology encountered an error while saving the document, a failure acknowledgment (non-zero status) will be sent on the existing open connection with an error message in the response element. An example failure acknowledgment is shown below:

```
<response status="1">Non existent or empty, userid or PersonIdCode element for jobId=2007091014514555</response>
```

Example **DegreeAuditRequest** Document

The following shows an example request document that contains one course taken by the student:

```
<RequestDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespaceSchemaLocation="
http://www.transfer.org/xsd/RequestDA_s1h.xsd">
    <Envelope>
        <RequestType>CAS</RequestType>
        <TransactionType>Request Degree Audit</TransactionType>
        <DocumentNumber>2009052712405402</DocumentNumber>
        <PartyId>https://www.transfer.org/uselect/service/xml</PartyId>
    </Envelope>
    <RequestDetail>
        <Control>
            <ControlBatchId>2009052712405402</ControlBatchId>
            <ControlRequestServerName>miamix</ControlRequestServerName>
            <ControlRecord>ASC</ControlRecord>
            <ControlRequestType>A</ControlRequestType>
        </Control>
        <RequestData>
            <Person>
                <PersonID>
                    <PersonIdCodeQualifier>CAS</PersonIdCodeQualifier>
                    <PersonIdCode>student123@MIAMI</PersonIdCode>
                </PersonID>
                <PersonName>
                    <NameType>CAS</NameType>
                    <NameFirst>CAS</NameFirst>
                    <NameLast>Student</NameLast>
                </PersonName>
                <StudentType>
                    <StudentTypeStatus>CAS Student</StudentTypeStatus>
                </StudentType>
            </Person>
            <PersonControl>
                <BatchId>2009052712405402</BatchId>
                <ControlRecord>ASC</ControlRecord>
            </PersonControl>
            <DegreeProgram>
                <Institution>
                    <Entity>
                        <EntityIDCodeQualifier>72</EntityIDCodeQualifier>
                        <EntityIDCode>1592</EntityIDCode>
                        <EntityIDCodeGroup/>
                    </Entity>
                </Institution>
                <AcademicProgram>
                    <ProgramType>Program</ProgramType>
                    <ProgramCode>825   44GEC</ProgramCode>
                    <ProgramCatalogYear>99983</ProgramCatalogYear>
                </AcademicProgram>
            </DegreeProgram>
            <CourseInstitution>
                <Institution>
                    <Entity>
                        <EntityIDCodeQualifier>72</EntityIDCodeQualifier>
                        <EntityIDCode>1643</EntityIDCode>
                        <EntityIDCodeGroup/>
                    </Entity>
                </Institution>
                <Course>
                    <CourseSubjectAbbreviation>ENG</CourseSubjectAbbreviation>
                    <CourseNumber>130</CourseNumber>
                    <CourseTitle>Intro to Technical Comm</CourseTitle>
                    <CourseSession>20083</CourseSession>
                    <CourseCreditValue>03.00</CourseCreditValue>
                    <CourseAcademicGrade>F</CourseAcademicGrade>
                    <CoursePseudoCode>N</CoursePseudoCode>
                </Course>
            </CourseInstitution>
        </RequestData>
    </RequestDetail>
</RequestDocument>
```

PartyId

PartyId is the URL the EDAG must connect to when sending the DegreeAuditReply XML document to Transferology.

DegreeProgram

The DegreeProgram element's children specify the audit's target school and program chosen by student.

Institution - School Id
AcademicProgram - Program

CourseInstitution

The children of this optional element specify the student's coursework.

Institution - School Id
Course - Course

DegreeAuditReply Document

The various *ReplyDA_s1* PESC schemas define numerous elements in the document; however, Transferology only extracts element content from the following elements:

- DegreeAuditResult.DegreeProgram.AcademicProgram.DegreeAuditReport.Report.**DegreeAuditBody**
- Envelope.**DocumentNumber**
- DegreeAuditResult.Person.PersonID.**PersonIdCode**

x.y.z syntax

The above elements are shown with x.y.z syntax, which is shorthand to remove the need for *"the z child element of y that is the child element of x"*.

DegreeAuditBody

The DegreeAuditBody element is the program content that will be displayed to the student.

Warning: HTML within program content

It is normal for this content to contain HTML. However, the following HTML elements must be avoided: <html> and <body>. Additionally, the <head> and it's child elements must be avoided with exception of the <style> element. To ensure the plans are correctly stored in Transferology and shown correctly by Transferology, keep the following points in mind:

- Use either an XML CDATA section or the standard escaping for XML predefined entities. See HTML in XML for further information.
- CSS styling must be inline via either style attributes for each HTML element or an inline style sheet element.

DocumentNumber

The DocumentNumber element is one of the two fields that uniquely identify the student's program request. Transferology will use the element content to locate the request in Transferology.

PersonIdCode

The PersonIdCode element is the other field that uniquely identifies the student's program request. Transferology will use the element content to locate the request in Transferology.

Example DegreeAuditReply Document

The following shows an example reply document.

```
<DegreeAuditReply>
   <Envelope>
      <DocumentNumber>2009052712405402</DocumentNumber>
   </Envelope>
   <DegreeAuditResult>
      <Person>
         <PersonID>
            <PersonIdCode>student123@MIAMI</PersonIdCode>
         </PersonID>
      </Person>
      <DegreeProgram>
         <AcademicProgram>
            <ProgramCode>ACCTBA</ProgramCode>
            <DegreeAuditReport>
               <Report>
                  <DegreeAuditBody><![CDATA[<div>Html of resulting plan...</div>]]>
                  </DegreeAuditBody>
               </Report>
            </DegreeAuditReport>
         </AcademicProgram>
      </DegreeProgram>
   </DegreeAuditResult>
</DegreeAuditReply>
```