

Update of The University of Missouri's Solution for Processing E-Transcripts

Presented October 23, 2011

By

Michael Jennings
University of Missouri - System

UM's Current Manual Transcript Processes

Outbound Processes:

- Print transcripts.
- Mail transcripts.

UM's Current Manual Transcript Processes

Inbound Processes:

- Receive transcripts from other institutions.
- Enter information from the received transcript into our student systems.

UM's Current Manual Transcript Processes

- Even though this is necessary, it is very time consuming and expensive.
- Here at the University, just like everywhere else, we are being asked to do more work with less resources.

E-Transcripts will be at least part of our solution to doing more with less.

Basically we are trying to electronically automate as much of our current manual business processes as possible.

Note: Not all institutions will need to do everything we are doing. You will need to look at the number of incoming and outgoing transcripts along with your business processes and make appropriate decisions for your institution.

E-Transcripts requires some form of Electronic Data Exchange (EDX)

- There are currently two protocols of EDX that Postsecondary Electronic Standards Council (PESC) supports. They are:
 - Electronic Data Interchange (EDI)
 - Extensible Markup Language (XML)

Electronic Data Interchange (**EDI**).

- EDI looks like a spreadsheet.
- All of the data is available in one easy place.
- You do have to know the position of the element you are looking at to know what it is.
- This protocol was created first and is very compact.

Extensible Markup Language (XML)

- XML looks like an html document similar to what you put out on your web site.
- It has beginning and ending tags around the data and is much more “eye-readable” than EDI.

Extensible Markup Language (XML)

Looks like:

```
<Source>
```

```
<Organization>
```

```
<ATP>006875</ATP>
```

```
<OrganizationName>University of Missouri -  
  Columbia</OrganizationName>
```

```
</Organization>
```

```
</Source>
```

The University of Missouri Chose XML

- Again, EDI and XML could have been used.
- We chose XML because:
 - Our institutions are starting to get involved with XML.
 - There is already some XML expertise available.
 - XML appears to be easier to read and write.
 - Maintenance to the programs to process XML is expected to be easier.

The University of Missouri Chose XML

- As expected the rest of this presentation is based on our XML Implementation.

Using XML Transcript for Outgoing Transcript Processing

- Generate and send XML Transcript files.
- Receive Acknowledgements files.
- Reconcile the Acknowledgements received with the Transcripts that were sent.

Using XML Transcript for Incoming Transcript Processing

- Receive and process Transcript XML files.
 - Note there are multiple ways you could process these. Our ultimate goal is to create an image to store and then completely automate the process to read the files and insert the data into the student system.
- Generate and send Acknowledgement XML files.

XML Files Definitions

- **College Transcript:**
 - It contains one transcript for one student.
- **Transcript Acknowledgement:**
 - It acknowledges a single transcript.
- **Batch XML Transcript:**
 - It this basically an outer wrapper to be able include multiple XML documents in one file.

College Transcript

- We are using College Transcript v 1.1.1
- It is documented in *Implementation Guide to the Postsecondary Electronic Standards Council XML Standard Format for the College Transcript Version 1.1.1 May 16, 2008*
- It is about 331 pages of detailed documentation.
- Currently there is a College Transcript v 1.3.0

Transcript Acknowledgement

- We are using Transcript Acknowledgement v 1.0.0.
- It is documented in *Implementation Guide to the Postsecondary Electronic Standards Council XML Standard Format for the Student Transcript Acknowledgment Version 1.0.0*
DRAFT April 22, 2007.
- It contains 63 pages of detailed documentation.

Batch XML Transcript

- The Batch XML Transcript is really a wrapper or an Envelope put around multiple Transcripts and Acknowledgements.
- There are two versions available. They are:
 - Academic Record Batch v 1.0.0
 - Academic Record Batch v 2.0.0
- Note: Version 2.0.0 is not compatible with Version 1.

Batch XML Transcript

- We are currently reading both versions of the Batch XML Transcript. I believe SPEEDE is currently sending version 1.0.0.
- We are currently only generating single XML Transcripts and Acknowledgments one document at a time. There are plans to do XML Transcripts in batches, but it is currently not implemented yet.

More on XML Files

- Just like all file structures, the data needs to be in a standard form.
- At the top of the file there is header information to help identify the version of the document.
- The remainder of the document contains the data surrounded by the appropriate XML tags.

More on XML Files

- There are several programs that can read the XML files to validate the XML document for form.
- These programs read the header information and then use that information to reference those files to validate the rest of the document.
- Like all programs they vary in what they can do and the price.

More on XML Files

- I use Liquid XML Studio Free Community Edition 7.1.6.1440.
- It does everything I currently need to do with XML and was suggested by some of my coworkers.
- It is also free.



Start Page USSR0300-COLUM-200...xml*

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- edited with Liquid XML Studio - Free Community Edition 7.1.6.1440 (http://www.liquid-technologies.com) >
3 <! by Steven Winters (University of MO - Columbia) -->
4 <ColTrn:CollegeTranscript xmlns:ColTrn="urn:org:psc:message:CollegeTranscript:v1.1.0" xmlns:AcRec="urn:org:pe
5 <TransmissionData>
6 <DocumentID>20091021T090902-MU-WINTERSS</DocumentID>
7 <CreatedDateTime>2009-10-21T09:08:58-05:00</CreatedDateTime>
8 <DocumentTypeCode>StudentRequest</DocumentTypeCode>
9 <TransmissionType>Original</TransmissionType>
10 <Source>
11 <Organization>
12 <ATP>006875</ATP>
13 <OrganizationName>University of Missouri - Columbia</OrganizationName>
14 </Organization>
15 </Source>
16 <Destination>
17 <Organization>
18 <ATP>006889</ATP>
19 <OrganizationName>University of Missouri - St Louis</OrganizationName>
20 </Organization>
21 </Destination>
22 </TransmissionData>
23 <Student>
24 <Person>
25 <SchoolAssignedPersonID>12345678</SchoolAssignedPersonID>
26 <SSN>123456789</SSN>
27 <Birth>
28 <BirthDate>1776-11-11</BirthDate>
29 </Birth>
30 <Name>
31 <FirstName>Michael</FirstName>
32 <MiddleName>L</MiddleName>
33 <LastName>Jennings</LastName>
34 </Name>
35 </Person>
36 <AcademicRecord>
37 </AcademicRecord>
```

Error List

Code	Description	File
------	-------------	------

General Programming Approach

- Steve Winters and Schuyler Wheeler have modified the existing transcript program to generate the XML files. The Transcript program also updates a set of tables so that reconciling the acknowledgements is easier.
- Mike Jennings is responsible for the rest.
- Obviously the work was not divided equally, but based on the programmers involved, their experience, and their responsibilities this was the best approach to try to meet our goal.

General Programming Approach

- The PESC specifications for the XML Transcripts are very flexible. They have tried to define data structures to try to meet everyone's needs.
- For example:
 - You can store courses by term or courses with no term information.
 - Academic Degrees, Academic Programs, and Academic Summaries can be stored in multiple places as well.

General Programming Approach

- UM Facts that drove our decisions:
 - The four campuses of the University of Missouri (MU, UMKC, Missouri S&T, and UMSL) are part of the University of Missouri System, but they each have their own business practices and policies. They effectively act as individual institutions, not as campuses from one larger institution.

General Programming Approach

- UM Facts that drove our decisions:
 - The campuses all are still closely tied together and all share a similar computing platform, computing environment, and most importantly share a common programming code base.

General Programming Approach

- It was decided to make loading the transcripts into a two step process.
 1. Read the XML files and load them into Staging Tables.
 2. Load the data from the Staging tables into the Student System tables.

General Programming Approach

1. Read the XML files and load them into Staging Tables.
 - The staging table do not completely mimic the PESC Transcript structure, but they are very close.
 - There are many places where the specification allows for zero to many occurrences of some of the optional fields. In most cases we created table structures for just one occurrence.

General Programming Approach

1. Read the XML files and load them into Staging Tables.
 - There are about 35 tables created to store the Transcript and acknowledgement data.
 - These tables will be used to load the transcript data into the student system and also be used to reconcile the Transcripts which are sent along with the Acknowledgements that are received.

General Programming Approach

1. Read the XML files and load them into Staging Tables.
 - There also about 15 tables and 36 views used to help support the integrity of the data.
 - Note: This is very general approach and could be used by anyone else that took this approach.

Describing PS_UM_XMLTR_SOURCE....

NAME	Null?	Type
UM_DOC_ID	NOT NULL	VARCHAR2(35)
UM_XML_TIMESTAMP	NOT NULL	VARCHAR2(25)
UM_DOC_TYPE_CD	NOT NULL	VARCHAR2(18)
UM_ORG_GRP_TYPE	NOT NULL	VARCHAR2(8)
UM_ORG_GRP_ID	NOT NULL	VARCHAR2(60)
UM_LOC_ORG_ID	NOT NULL	VARCHAR2(35)
UM_LOC_ORG_QUAL	NOT NULL	VARCHAR2(2)
UM_ORG_NAME	NOT NULL	VARCHAR2(60)
UM_ADDR_LINE1	NOT NULL	VARCHAR2(40)
UM_ADDR_LINE2	NOT NULL	VARCHAR2(40)
UM_ADDR_LINE3	NOT NULL	VARCHAR2(40)
UM_CITY	NOT NULL	VARCHAR2(30)
UM_STATE	NOT NULL	VARCHAR2(2)
UM_POSTAL_CD	NOT NULL	VARCHAR2(17)
UM_COUNTRY_CD	NOT NULL	VARCHAR2(2)
UM_ATTENTION	NOT NULL	VARCHAR2(45)
UM_CNTRY_PREFIX	NOT NULL	VARCHAR2(3)
UM_AREA_CD	NOT NULL	VARCHAR2(5)
UM_PHONE_NBR	NOT NULL	VARCHAR2(11)
UM_PHN_EXTENSION	NOT NULL	VARCHAR2(5)
UM_EMAIL_ADDR128	NOT NULL	VARCHAR2(128)
UM_URL		VARCHAR2(512)

```
select * from PS_UM_XMLTR_SOURCE
```

[1]: Statement processed in 0.00 sec; see Spool tab

UM_DOC_ID	UT123456
UM_XML_TIMESTAMP	2006-02-05T10:30:48-06:00
UM_DOC_TYPE_CD	Acknowledgment
UM_ORG_GRP_TYPE	ATP
UM_ORG_GRP_ID	006882
UM_LOC_ORG_ID	
UM_LOC_ORG_QUAL	
UM_ORG_NAME	THE UNIVERSITY OF TEXAS AT AUSTIN
UM_ADDR_LINE1	Office of Admissions
UM_ADDR_LINE2	
UM_ADDR_LINE3	
UM_CITY	Austin
UM_STATE	TX
UM_POSTAL_CD	78712
UM_COUNTRY_CD	
UM_ATTENTION	
UM_CNTRY_PREFIX	
UM_AREA_CD	512
UM_PHONE_NBR	00004757638
UM_PHN_EXTENSION	
UM_EMAIL_ADDR128	
UM_URL	

General Programming Approach

2. Load the data from the Staging tables into the Student System.
 - Since each campus has their own business rules and policies this program will have lots of “if-then-else” logic to support each campuses’ requirements.
 - Here is where all of the mapping takes place.
 - This code is not very useful to anyone but the University of Missouri.

Current Accomplishments

- Registration with the SPEEDE Server with the University of Texas at Austin is completed and tested.
- Initial coding of Generating the Transcript XML files is complete, but is in User Testing.
- Initial coding of Loading the staging tables from the XML transcripts and XML acknowledgements files complete, but is in User Testing.

Current Accomplishments

- Eleven Online Pages were created to help the user maintain the setup tables, crosswalk tables (used for mapping), and submit the batch processes. This is in user Testing
- Fourteen Online Pages were created to help the user cleanse the incoming data and finalize the mapping from PESC to the student system. This is in user Testing
- Three Online Pages were created to help the user Reconcile the XML Transcripts. This is in User Testing.

Current Accomplishments

- Initial coding to reconcile the XML Transcripts sent with the XML Acknowledgements received is complete, but it is in User Testing.
- Initial coding to Print an XML Transcript has been completed. This is a text based transcript that basically mimics the Quick 'n' Easy (QnE) Utility from SPEEDE. This was originally intended a utility program, but will probably be used to generate transcripts for our Document Imaging Software. It is in User Testing.
- Initial coding to populate the student system with the data that is stored in the XML Tables is complete and in User Testing.

Tasks Remaining

1. Finish our Initial Testing with the UM schools.

- The four campuses of the University of Missouri are initially creating trading partnerships among themselves and are working very hard to get this into production soon.

Tasks Remaining

2. Make some more enhancements.

- Implement the capability for Search/Match to correctly map the Student's Student Id Number.
- Implement the ability to only add the additional part of a transcript for students who take additional courses after they are already at our School.
- Programmatically load the transcript into our Document Imaging software.

Tasks Remaining

3. Seek More Trading Partner Agreements

- We have identified MCC and SLU as other early partners. We hope to start with MCC mainly because they also use PeopleSoft.
- We will start work with SLU as soon as MU, UMKC, Missouri S&T, UMSL, and MCC are sharing XML transcripts electronically.
- The University of Missouri is very interested in seeking Trading Partnerships with other schools as well.

Final Comments

- The University of Missouri is committed to using E-Transcripts.
- We want to encourage other schools around the state to use E-Transcripts.
- We want to try to help other schools get this project off the ground.

Final Comments

The University of Missouri will share:

- The Data Model of the Staging Tables.
- Actual SQL used to create our Staging Tables.
- Application Code if you are a PeopleSoft School.
- Be available as time permits to help answer questions and share our experiences.
- Work with you to become a Trading Partner.

Questions?

Michael L. Jennings Database Programmer Analyst - Expert
Enterprise Applications Services (EAS) Division of IT (DoIT)
University of Missouri – System Phone: 573-884-7288
615 Locust Street, E204-10 Fax: 573-884-5589
Columbia, MO 65211 jenningsm@missouri.edu