

Set Up: Asynchronous Enrollment Integration

eCollege™
4900 South Monaco Street
Denver, CO 80237

Contents

eCollege	1
OVERVIEW	3
INITIAL SETUP WITH CLIENT SERVICES	3
CREATE THE DATA FILE	5
DELIMITED FILE	5
<i>File name</i>	6
<i>File formatting</i>	6
<i>Common delimited file errors</i>	7
<i>File size</i>	7
XML	8
<i>File name</i>	8
<i>Common errors</i>	9
<i>File Size</i>	9
SUBMIT THE DATA FILE	9
SUBMIT VIA SECURE FILE TRANSFER PROTOCOL (SFTP)	9
<i>Firewall</i>	9
<i>Configure the FTP client</i>	10
<i>Connect to the SFTP server</i>	10
SUBMIT VIA THE ADMINISTRATIVE PAGES	10
CHECK THE STATUS OF THE REQUEST	11
APPENDIX A: SAMPLE AUTOGENERATED EMAIL	12
APPENDIX B: FORMATTING A DELIMITED FILE—REQUIRED FIELDS	13
APPENDIX C: SAMPLE DELIMITED FILE	15
EXAMPLE	15
APPENDIX D: COMMON DELIMITED FILE ERRORS	16
APPENDIX E: COMMON CUMULATIVE FILE ERRORS	18
APPENDIX F: XML OVERVIEW	19
ELEMENTS	19
ROOT ELEMENT	19
NESTING ELEMENTS	19
ATTRIBUTES	20
SCHEMAS	20
VALID OR WELL-FORMED XML DOCUMENT?	20
IMS ELEMENTS IMPLEMENTED BY ECOLLEGE	20
APPENDIX G: SAMPLE XML DATA	21
APPENDIX H: COMMON XML ERRORS	23

Overview

You've selected to integrate your Student Information System (SIS) with the eCollege database using *asynchronous* integration. This option essentially allows your SIS to "talk" to the eCollege data system at a predetermined time or interval and share user creation and enrollment data. For example, users and enrollments could be updated via asynchronous integration every hour, every day, once a week, every Monday and Friday, etc. Once an asynchronous request is submitted, the request is processed within 24 hours.

There are two ways you can format your user/enrollment information:

- **Delimited file**; or
- **XML**

You can compile your user data in either format. Because it is extremely important that your file be formatted correctly—regardless of which structure you choose—this document will provide tips and tricks for proper, error-free formatting.

Once you've formatted or created your data file, there are two ways that you can send or upload the file to the eCollege system:

- **Secure File Transfer Protocol (SFTP)**; or
- **Administrative Pages upload**

You can use either of these methods to transfer your SIS information to the eCollege system. This document includes instructions for setting up a secure FTP site, as well as information on accessing and using the Administrative Pages. Use this document to select which transfer method is best for you.

Note: Batch XML files (i.e., asynchronous XML files) and delimited files—both of which can be submitted via Secure FTP or through the Administrative Pages—end up in the same file location. As a result, please notify your Client Services representative if you plan to submit both XML and delimited files via SFTP and/or the Admin Pages.

Before you format your user/enrollment information and select your transfer process, however, you'll need to get with your Client Services representative for some preliminary set up.

Initial setup with Client Services

Your Client Services representative will need to provide you with some information as you begin your integration. Similarly, your Client Services representative will need to collect information from you to ensure that the integration process for your institution is set up correctly from the eCollege side. The table below summarizes what information you can expect to receive from your Client Services representative and what information you'll need to provide to your Client Services representative.

API Setup Information	Description
Your Client Services representative will provide:	
Username and password	Regardless of your integration method, you will receive a Username and Password with which you can access your secure FTP site and/or call the API web services. This information is determined by the eCollege IT department and will be provided by your Client Services representative.

Secure FTP site license key and registration number	The eCollege SIS for API <i>requires</i> a secure FTP site. If you're already set up and using secure FTP, you can continue to use your existing solution. If you're not already using a secure FTP site, eCollege will provide a one-year-license to CuteFTP Pro <i>free of charge</i> . After one year, you will need to maintain your secure FTP server license. It does not matter which secure FTP software you use. If you decide to use the CuteFTP Pro software, you'll need a license key and registration number, which your Client Services representative will provide. <i>Remember that after the first year, you'll be responsible for your CuteFTP license if you choose to continue with CuteFTP.</i>
SFTP Folder	This is determined by the eCollege IT department and will be provided by your Client Services representative.

API Setup Information	Description
You will provide your Client Services representative with:	
File type	Will you be sending delimited files or XML data?
If delimited file, delimiter type?	Will your file be delimited by commas, tabs, or pipes ()?
Using Division Tools?	Yes or No. <i>This is only relevant for institutions using Division Tools.</i> The field is required, however, for column count and validation. If you're not using Division Tools, this value is "No."
Send Student Email?	<p>Yes or No. The API can be configured to send users autogenerated emails based on either account creation or enrollment in a course. Emails can be differentiated by enrollable node, role type (i.e. Student Email, Faculty Email) or action (i.e. Enrollment Confirmation, Drop Confirmation, Waitlist, etc.). These emails can also pass the following user-specific information:</p> <ul style="list-style-type: none"> • User Name • User Login ID • User Password • Course(s) Enrolled <p>Auto-emails must be configured during the initial set-up of the API, and can be updated by your CS representative. You can create your own email "templates" and send them to your CS representative, or you can use the eCollege template, available in Appendix A, as a starting point.</p>
Generate User Passwords?	Yes or No. The API can be configured to auto-generate random passwords for all new users if you leave the password field blank. The auto-generated password will be included in the instruction email sent to the user (if you've chosen to send student email alerts). However, if a password is supplied, it will be used.

Update User Properties?	<p>Yes or No. The API can be configured to update user properties, based on each user’s login ID. If this option is set to “yes,” any changes in name, password, email address, or any of the optional user property fields will result in updated information within the eCollege system when the API file is processed.</p> <p>If this property is set to “no” or “off,” any properties differing from what is in the eCollege database will not be updated when the API request is processed. Thus, if you expect user information to be updated through the eCollege system, and that information is not also updated in your SIS, it is possible that updated information in the eCollege system will be replaced by “old” information from your SIS during the integration process. If changes are made only rarely in the eCollege system, and your SIS is always up to date, then this risk is minimal. Similarly, if your SIS only sends incremental data (i.e., only records that have changed since the last integration), rather than snapshot or cumulative data, there is little risk.</p> <p><i>Note that if you are sending data in a delimited format, the only way a user can be identified is by the Login ID. As a result, Login ID cannot be updated when using a delimited format. The XML data format identifies users differently and thus allows you to update the Login ID for a user.</i></p>
If not updating user properties, source for student email address?	If you’ve chosen to send students email notifications (upon user creation or enrollment, for example) but you’re not updating user properties, there’s a chance that the email address in the eCollege system may become out-of-date. As a result, you need to decide the “default” system of record from which student email notifications should come—send notifications to the student email address in your SIS file or the email address in the eCollege database?
Email EP?	Yes or No. This option sends an auto-generated summary email to the submitter or to an email alias you specify when the file has been processed. One email is sent per API request.
If yes, EP Email Address?	This is the email alias you would like automatic updates sent to.

Create the data file

Once you’ve set up the behind-the-scenes or administrative part of the integration process, you’re ready to create or format your user and enrollment information. The eCollege SIS for API accepts either delimited files or XML. Both data types can be submitted via SFTP or uploaded through the Administrative Pages. Create or format your user information using either a delimited file or XML.

Delimited file

This stripped-down file format includes a fixed number of columns, each separated by a “delimiter” like a comma, a tab, or a pipe character. Every delimited file **must** follow a specific format exactly—it must include the correct number of columns in the correct order or it will not process.

[Appendix B](#) gives you all the necessary data fields in the correct order.

[Appendix C](#) shows you a sample delimited file.

[Appendix D](#) explains some of the more common delimited file errors.

Delimited files can be easily created using a spreadsheet like Microsoft Excel or a text editor like Notepad. The file you submit should have a .txt extension.

Note: *Cumulative* data files include a "history" of all user actions and events for a given period of time—typically a term or a quarter. For example, if a student was added to a course, then dropped, and then added to the same course, a cumulative file would include all three status changes in the submitted file. If you plan to submit cumulative files, you **must** use the **delimited file** option. XML data format (whether synchronous or asynchronous) cannot include cumulative records. Remember, too, that cumulative files include a high percentage of redundant and obsolete information and are the most prone to error due to improper ordering of information in the file. Cumulative files also take the longest time to process.

File name

The preferred file naming convention for delimited files is:

ClientString_DD_MM_YYYY.txt

For example, a client whose client string is "EP" would have a file name of "EP_26_03_2007.txt" for the twenty-sixth of March, 2007. You can look up your Client Sort String information using the **Enrollment ID Lookup utility** on the **Manage Users** tab in the eCollege Administrative Pages. While you can vary the file name to meet your needs, the file extension *must* be ".txt".

Note: If two files are sent on the same day and have the *exact* same file name, the second file will have an incremented value appended to the file name. For example, if file **a.txt** is submitted in the morning and another file, **a.txt** is submitted in the afternoon, the second file would have its name changed to **a(2).txt**. Note, too, that copies are not guaranteed to stay in the root directory longer than 60 days.

File formatting

Every delimited file **must** follow a specific format exactly—the eCollege system expects specific data in a specific order; if the data is mixed up or simply missing, the file cannot be read and, therefore, processed.

Please refer to [Appendix B](#) for the data fields that must be included in every file. Remember that even if you're not submitting data in a particular field (like middle name, for example), you still need to create a "placeholder" for that field in your file.

See [Appendix C](#) for a sample formatted file.

Common delimited file errors

[Appendix D](#) lists some of the most common errors in delimited files. Please refer to this appendix for help and/or troubleshooting before you submit your delimited file. Some important things to look out for initially, however, are:

- The file does not use an accepted delimiter—only commas, tabs, or pipe characters can be used to separate data.
- The file does not contain the proper number of columns. Even if you are not providing data in every field/column, you must “hold a space” for the column.

File size

Although there is currently no file size limitation, very large files should be sent via SFTP. *(Please note, there is a **1MB file size limit** for jobs submitted through the Admin Pages.)* Following are some things to consider as you create and format your file:

- Some institutions choose to send **cumulative** files through the API. Cumulative files include a "history" of **all** user actions and events for a given period of time, typically a term or a quarter. In order to insure that the final state of the user is correct, there are additional formatting rules for cumulative files that must be followed. As a result, cumulative files take the longest time to process, as they are also processing redundant information, and are the most prone to producing unexpected results. For example, a submitted job has Line 9 Johnny as Student in English 101; Line 10 Johnny Drops English 101; Line 11 Johnny as Student in English 101. This happened because there was a mistake and Johnny shouldn't have been dropped. If this is a cumulative file, the same information is sent day after day after day. Now the deadline to ADD a course has passed, but the deadline to DROP a course has not. If the file is split during processing due to size and the split occurs between Line 10 and 11, even if part A and part B process in that order, Part A will drop Johnny, but Part B will be unable to enroll Johnny back in the course because the ADD date has passed. See [Appendix E](#) for more information about sending cumulative files.
- Should you experience a network outage or lose your network connection while you're submitting your file, the transmitted data you might lose is much smaller if you're submitting small files.
- Occasionally the API may need to break up or “chunk” large files in order to process them. If a file happens to be broken in the middle of a group, there may be strange results. For example, a file might be broken in half. There is no guarantee, however, that the records from the first and second chunks of the file will process in sequential order (e.g., in a cumulative file Line 10 is Johnny as Student in English 101, Line 11 is Johnny as Withdraw in English 101. If the file is broken between lines 10 and 11, there is a good chance line 11 will process FIRST (part A and part B get processed simultaneously), and line 10 will process second, with the end result being that Johnny remains a Student in English 101. Smaller file size reduces the need to break up or chunk files.

XML

Similar to HTML, Extensible Markup Language (XML) provides a basic structure and rules for users to create their own custom markup tags and enables the definition, transmission, validation, and interpretation of data between applications and organizations, regardless of where the applications and organizations reside. The actual markup looks very much like HTML, although the grammar rules for an XML document are more strict. The eCollege API adheres to the IMS Enterprise v1.1 XML schema for user and enrollment information.

One benefit of using XML files is that you can create custom user properties. The delimited format requires that the file include a specific number of predefined data columns. As a result, it's only possible to create extended user fields that are unique or specific to your institution using the XML data format. Also, the XML option is the only integration that allows you to change or update a user's login information after the user has been created (this requires that you have chosen to update user properties through the API for SIS, however).

[Appendix F](#) will give you a brief, general overview of XML.

[Appendix G](#) shows you sample XML data, formatted according to the eCollege implementation of the IMS specification.

[Appendix H](#) lists some of the more common XML errors.

File name

The preferred file naming convention for delimited files is:

ClientString_DD_MM_YYYY.xml

For example, a client whose client string is "EP" would have a file name of "EP_26_03_2007.xml" for the twenty-sixth of March, 2007. You can look up your Client Sort String information using the **Enrollment ID Lookup utility** on the **Manage Users** tab in the eCollege Administrative Pages. While you can vary the file name to meet your needs, the file extension *must* be "xml".

Note: If two files are sent on the same day and have the *exact* same file name, the second file will have an incremented value appended to the file name. For example, if file **a.txt** is submitted in the morning and another file, **a.txt** is submitted in the afternoon, the second file would have its name changed to **a(2).txt**. Note, too, that copies are not guaranteed to stay in the root directory longer than 60 days.

File formatting

XML schemas provide a means for defining the structure, content, and semantics of XML documents. In the XML documents you will be creating for eCollege, you will be using the [IMS Enterprise v1.1](#) schema. See [Appendix F](#) for specific information on formatting your XML.

Common errors

[Appendix H](#) lists some general errors in XML data. The most common XML-failures occur as a result of either the XML not being valid or the XML does not match the IMS XML schema. Please refer to Appendix I for help and/or troubleshooting before you submit your request.

Some important things to look out for as far as actual data in your XML file are:

- Check for apostrophes in names, as they may cause problems with files
- All required fields must have a value

File Size

Small files consisting of less than 10 users and enrollments can be looked over for formatting, special characters, etc. rather easily should something go wrong, but huge XML files (containing 100's of users) are very difficult to troubleshoot since even though the XML format is very structured, the particular data elements to enroll this user in that course are scattered all over within the file. While each line in a delimited file can be treated as discrete, the XML file has to be processed as an integrated whole.

Submit the data file

There are two ways that you can send or upload your data to the eCollege system, regardless of whether the data is in a delimited file or formatted as XML:

- **Secure File Transfer Protocol (SFTP), or**
- **Administrative Pages**

Submit via Secure File Transfer Protocol (SFTP)

FTP is a standard protocol for transferring a file from one computer to another over the Internet. You can use an automated or manual Secure FTP client to contact the eCollege SFTP server and send a file to your directory on the eCollege system. The eCollege SIS API monitors that drop-off point and will then pick up the file to be processed. For most current users, submitting files through SFTP represents no change to the current integration process.

Note: You'll need a secure FTP site before you can submit via FTP

Firewall

eCollege uses non-standard ports for secure FTP server (SFTP). Prior to connecting, please check with your Network Administrators to ensure ports 13265 – 13267, 28,000 – 30,000 are available to IPs: 216.58.170.118 – 216.58.170.120

Configure the FTP client

To configure the FTP client, you must configure the SFTP client with the required parameters to connect to the eCollege SFTP server.

1. In the **Host** field, type externalftp.ecollege.com
2. Enter your username and password.
3. Select one of the following protocol types. In the **Port** field, change the default port number for the selected **Protocol Type** accordingly:

Protocol Type	Default Port	Change Port To
FTP with TLS/SSL	990 – Implicit	13267
FTP with SSL	Auth SSL – Explicit	13265
FTP with TLS/SSL	Auth TLS – Explicit	13265
SFTP using SSH2	Secure Shell	13266

4. From the **Data connection type** drop-down menu, select **Use PASV**.
Note: For the SFTP using SSH2 Protocol Type, use the parameter default values. Click **OK** to save your connection parameters.

Connect to the SFTP server

When your configuration is complete, connect to the eCollege SFTP server to ensure your installation was a success.

Submit via the Administrative Pages

Users with access to the eCollege Program Administrative Pages can upload a batch file (either a delimited file or an XML file) through the Program Admin Pages. When you upload directly to the Program Admin Pages, the file you submit is validated for formatting during the upload process and you'll be notified immediately if the file has formatting problems or cannot be processed.

To submit your file through the Program Admin Pages:

1. Log in to the Program Administration Pages.
2. Click the **Automated User Management** link in the left-side navigation. In the screen that displays, you can upload requests or view the status of previously submitted jobs.
3. Select **Upload user management file** and click **Next**.

Follow the instructions on-screen to complete the upload process, or refer to the online Help for more detailed information.

You can check the status of any submitted request—even those submitted via (SFTP)—from the Program Admin Pages. To check the status of your request, as well as view errors, see the section of this document called [Check the status of the request](#).

Check the status of the request

Once you've submitted your request, whether by SFTP or through the Program Admin Pages, you can check the status of the request, as well as view errors and validation results. You can use this reporting option to view the status of a specific request, or you can view the status of all requests within a certain timeframe.

To check the status of your request, as well as view errors:

1. Log in to the Program Administration Pages.
2. Click the **Automated User Management** link in the left-side navigation. In the screen that displays, you can upload requests or view the status of previously submitted jobs.
3. Select to view a single, specific user management request, or to view the status of all requests processed on a certain date; then click **Next**.

Follow the instructions on-screen to complete the process, or refer to the online Help for more detailed information.

You can view the status of a request for up to 60 days. After that, the record is no longer available in the Program Admin Pages. However, you can download or export a report to your local machine for your own records.

Note: Should you need to re-submit data, we suggest submitting an incremental file. Incremental data files include only changes that have occurred since the last time the file was run—in the case of an error, this would mean processing only records with errors. The system does not then need to receive and re-process every record.

Appendix A: Sample autogenerated email

eCollege can designate an auto-email to be sent to users as they are created or enrolled in their online courses. If you'd like to automatically send email notifications, please send an email template with the information you would like sent to your users to your Client Services representative. Feel free to use our standard auto-email as a template for your own:

Hello [Student First, Student Last],

This message is a notification of enrollment as a student in a course hosted on the [Your institution name] eLearning Platform. Login at <http://www.yourinstitution.net>.

Use the following credentials when asked for your username and password:

Username: [user name]

You have been enrolled in the following course(s):

QA111 | Test Course 111
QA112 | Test Course 112

- If you are a new student, your password is a combination of the year/month you were born (yyyymm)(i.e., If you were born in June of 1975, your password would be 197506).
- If you are a continuing student, your password was not changed, and will still be the last password you used when accessing your courses on www.yourinstitution.net.

When you log in, you will see your courses listed on your Personal Home Page. If you are new to the [Your institution name] eLearning Platform, you should view the Student Tutorial listed on your Home Page. Your instructor or facilitator will provide you with further information about participating in your eLearning courses.

If you have difficulties logging in, please contact the Help Desk at helpdesk@yourdomainname.net. If you do not know why you received this e-mail, please contact your campus Academics Department or the staff at your Center.

[Your institution name] eLearning Support Team

Appendix B: Formatting a delimited file—required fields

You'll need to make sure that the delimited file you upload is formatted so that the system can read it. The file must be a .txt file with data separated by commas, tabs, or pipe characters and must follow specific formatting guidelines *exactly*. The following table and information will help you format your file correctly.

Create a table with ALL of the following column headings (you should have **17 or 18 columns** in your table depending on whether you're using Division Tools). Even if you don't plan to enter any information in a particular column, you still need to create a "place holder" for that column (except for the last column—School ID). Fields that are required for User Creation are identified in the table.

Note: All requests for a particular user enrolled in a particular course **must** be grouped together in your delimited file, with the most current request for that user appearing LAST in the file. The last request in a file is the one that will be processed, regardless of the date any other changes were made (i.e., if a user was added to a course, then dropped, then re-added, but the "drop" request appears LAST in the file, the user will be dropped from the course).

Field	Required for user creation	Comments
First Name	X	
Middle Name		
Last Name	X	
Login ID	X	The Login ID must appear exactly as it is in the eCollege system. For example, if the student's login ID includes hyphens and hyphens are required in registration, hyphens must be included in the text file. The login ID is the primary identifier for the user in the eCollege system.
Email address	X	The API will check for a valid email format (username@someplace.com , .edu, .net, etc.). Each user can have one email address only.
Password	X	The API can auto-generate passwords for all new users if you leave the password field blank. The auto-generated password will be in the login instruction email sent to the user if this feature was enabled during the initial configuration of your API. If not configured, the API will not generate a password, and as a result a new user line that does not contain a password will fail.
Daytime phone number		
Evening phone number		
Street Address		
City		
State/Province		
Postal Code		
Country		

Enrollable Node Client Sort String	X	You can look up your Enrollable Node Client Sort String information using the Enrollment ID Lookup utility on the Manage Users tab in the eCollege Administrative Pages. This value is used by eCollege to place the student into the correct node / enrollable area in your online campus. Your client sort string will look something like this: "schoolcode.programcode.". This sort string MUST have a trailing period. Note that the Enrollable Node Client Sort String must be included in order to change the Enrollable Node Status.
EP Course Call Number	X	You must provide a unique identifier for each course in order to process course enrollments. Each instance of a cross-listed course must have its own unique identifier. If the user's enrollable node status needs to be updated, this value must be 0. When more than one call number identifies a course slot in the eCollege system, and the same user is being enrolled using more than one of those course call numbers, the API will not treat those two different course call numbers as cumulative records. Both requests will be processed and the last one processed "wins."
Role ID	X	Role IDs are used to determine what permissions a user will have in the selected node and/or course. Role IDs can be found in the Enrollment ID Lookup utility on the Manage Users tab in the eCollege Administrative Pages. To drop a user , you must explicitly include a line in the file with a Role id of 5. Refer to the Enrollment ID Lookup utility for all Role id numbers.
Instate	X	The Instate value identifies student residency for tuition purposes. This is a True/False value, so the number you enter in this field will be 1 or 0, respectively (i.e., 1 = Instate; 0 = Out-of-state).
School ID	(For Division Tool users only)	This final column—School ID—is only relevant for institutions using Division Tools. The appropriate numbers/values will be provided by your Client Services representative. If you do not use Division Tools, this field must not be passed in the delimited file. Passing this field, even without passing actual information, will result in file failure if Division Tools have not been implemented.

Note: Remember to delete the column headings from your file before you submit it. If you forget, the system will try and read the column information and the exchange will fail.

Appendix C: Sample delimited file

Following is an example of a comma-delimited text file format. Each value is separated by a comma, and if a non-required value is not present, then two or more delimiters act as placeholders. In the example below, the file does not contain day phone, evening phone, street address, city, state, postal code, or country, however the fields for each of these values is still held by a delimiter.

Valid delimiter values are:

- T – tab delimited
- , (a comma) – comma delimited
- | – pipe delimited

Note also that you will need to include a hard return at the end of each line of data. If you are coding the hard return, you will need to include an End Of Line String variable at the end of each record and a carriage return (“\r”), followed by a line feed (“\n”).

Because the eCollege system runs on a window server, always create the hard return in Windows Style.

For example, you might enter:

```
eolstr = “\r\n”
```

where:

- eolstr is the variable,
- “\r” is the carriage return, and
- “\n” is the line feed.

Example

```
Astrit,,Mehmeti,LOGIN766,astrit@sch.edu,71992,,,,,,,,sch.prog.BIS.,BIS22S766,1,0  
Astrit,,Mehmeti,LOGIN766,astrit@sch.edu,71992,,,,,,,,sch.prog.BIS.,BIS22766,1,0  
Nick,,Powers,LOGIN775,npowers@sch.edu,31993,,,,,,,,sch.prog.MGMT.,MGMT44S775,1,0  
Nick,,Powers,LOGIN775,npowers@sch.edu,31993,,,,,,,,sch.prog.BUSN.,BUSN319S775,1,0  
Nick,,Powers,LOGIN775,npowers@sch.edu,31993,,,,,,,,sch.prog.MATH.,MATH221775,1,0  
Michael,,Vasilou,LOGIN746,mvasilou@sch.edu,31989,,,,,,,,sch.prog.ACCT.,ACCT346S746,1,0  
Michael,,Vasilou,LOGIN746,mvasilou@sch.edu,31989,,,,,,,,sch.prog.BUSN.,BUSN42S746,1,0  
Michael,,Vasilou,LOGIN746,mvasilou@sch.edu,31989,,,,,,,,sch.prog.ACCT.,ACCT21746,1,0  
Josie,,Anagbogu,LOGIN822,janagbog@sch.edu,71992,,,,,,,,sch.prog.CIS.,CIS45S822,1,0  
Josie,,Anagbogu,LOGIN822,janagbog@sch.edu,71992,,,,,,,,sch.prog.CIS.,CIS438S822,1,0  
Gail,,Kroepel,LOGIN28,gkroepel@sch.edu,61981,,,,,,,,sch.prog.CIS.,CIS34928,1,0  
Gail,,Kroepel,LOGIN28,gkroepel@sch.edu,61981,,,,,,,,sch.prog.CIS.,CIS4528,1,0  
Veronica,,Noble,LOGIN3376,vnoble@sch.edu,32,,,,,,,,sch.prog.CIS.,CIS2113376,1,0  
Douglas,,Skinner,LOGIN3341,dskinner@sch.edu,721,,,,,,,,sch.prog.MATH.,MATH13341,1,0  
Paul,,Bierbauer,LOGIN189,pbierbau@sch.edu,9198,,,,,,,,sch.prog.ECT.,ECT261S189,1,0  
Paul,,Bierbauer,LOGIN189,pbierbau@sch.edu,9198,,,,,,,,sch.prog.COMP.,COMP128S189,1,0  
Paul,,Papaioannou,LOGIN34,ppapaio@sch.edu,11968,,,,,,,,sch.prog.EET.,EET22S34,1,0  
Paul,,Papaioannou,LOGIN34,ppapaio@sch.edu,11968,,,,,,,,sch.prog.EET.,EET42S34,1,0  
Paul,,Papaioannou,LOGIN34,ppapaio@sch.edu,11968,,,,,,,,sch.prog.EET.,EET44S34,1,0  
Timothy,,Hart,LOGIN795,thart@sch.edu,11996,,,,,,,,sch.prog.HUMN.,HUMN42S795,1,0
```

Appendix D: Common delimited file errors

Following are some of the more common errors encountered when submitting a delimited file. We suggest you look over this list before submitting your file.

Error	Description
Delimiter errors	
Incorrect delimiter	The delimiter is not a comma, a tab, or a pipe ().
Filename errors	
Special Characters in filename	Your file name should be formatted as: ClientString_DD_MM_YYYY.txt The file name cannot contain special characters (like “ ‘ , “&,” @”, etc.).
Invalid file extension	Your delimited file must have a .txt extension.
Formatting errors	
Extra column	Every delimited file must have 17 or 18 columns <i>only</i> , depending whether you’re using Division Tools.
File size errors	
Large cumulative file	Large files should process, but be aware that they are more prone to errors and system-time out. If eCollege has to break up a large file to upload it, data may break and cause strange results.
File is larger than upload limit	Files larger than 3 MB may time-out.
Data errors	
Missing required field	Make sure all required fields contain data—especially for user creation.
Missing Password w/o password generation ON	If your API is not set up to automatically generate user passwords, you must submit a password for each user.
Invalid email address	The API will check for a valid email format (username@someplace.com , .edu, .net, etc.). Similarly, the email field cannot be left blank.
Username includes a restricted character	The Login ID cannot contain any of the following characters: %][+<>:'=:/\
Password with restricted character	The password cannot contain any of the following characters: %][+<>:'=:/\
Clientstring missing period	Your Client sort string should include a trailing period at the end of the sort string: schoolcode.programcode. However, if the period is missing, it will be added during processing and will not affect the file’s ability to process successfully.
Role ID not numeric	Role IDs can be found in the Enrollment ID Lookup utility on the Manage Users tab in the eCollege Administrative Pages and must be a numeric value.
Multiple email addresses	Each user is limited to a single email address.
Clientstring sent is unenrollable	Only enrollable nodes will allow enrollments. A list of all enrollable sort strings can be found at http://eputilities.live.ecollege.com .
Attempt to drop user from last enrollable node	This message means that if the node enrollment is dropped, the user will no longer be able to be associated with your institution. The API cannot drop a user from their last node.
Username not unique for explicit user creation	The user has already been created in the system. Username/Login ID must be unique.
Drop user that does not exist	You’ll see this message if a user is not currently enrolled in a course and a DROP (ID: 5) action is passed for the user.
Drop user with drop date in the past	The Drop date for the Term must be in the future. Drop dates can be altered under Term Management in the Admin Pages.

Prof course enrollment with Student node enroll	A user with a Student Role ID at the node enrollment level cannot be enrolled as a professor at a course level.
Course enrollment with add date in the past	The Add date for the Term must be in the future. Add dates can be altered under Term Management in the Admin Pages.
Enroll in course where max enrollment has been reached	The Course Maximum value will not allow users to be enrolled once the maximum has been reached. Course Max can be altered in the Edit Courses section of the Admin Pages.
Waitlist student where max waitlist has been reached	You'll see this error is you try to enroll a user as a "Waitlist Student" in a course which has reached its Waitlist max. The Waitlist Max can be altered in the Edit Courses section of the Admin Pages.
Attempt to add/update an Admin role type	The API will not enroll an Administrator role type.
Drop user from node with course enrollments below	A user can only be dropped from a node when a) they are not enrolled in any courses within that node, and b) if they are enrolled in at least one other enrollable node. This message means that if the node enrollment is dropped, then the user will no longer be able to be associated with the client. The API cannot drop a user from their last node.

One other thing to consider:

- Make sure you delete the column headings from your file before saving and submitting the file.
- Make sure you include a hard return at the end of each line of data. See [Appendix C](#).

Note for Excel Users: Excel's comma-delimited export format does not conform to the API specs and should therefore not be used. If you want to use a spreadsheet to create your file, you should 1) make sure the upper-leftmost cell (cell A1 in Excel) contains only the letter T, and 2) export using the Tab-delimited format. Files created in this manner are compatible with the API, so long as the Tab-delimited type is configured when the API is set-up.

Appendix E: Common cumulative file errors

Some institutions choose to send cumulative files through the API. Cumulative files are files that include historical user information—they include a "history" of **all** user actions and events for a given period of time, typically a term or a quarter. For example, if a student was added to a course, then dropped, and then added to the same course, a cumulative file would include all three status changes in the submitted file.

Cumulative files usually include a high percentage of redundant and obsolete information and are the most prone to error due to improper ordering of information in the file. Cumulative files also take the longest time to process. If you need to send cumulative files, please make sure you read the common errors listed below before submitting your file.

- Since cumulative files contain **all** add/drop actions, the delimited file **must** be grouped by user and course **with the latest action being the last in the grouping**. It is common problem with cumulative files for users to be repeatedly added/dropped because their records are randomly placed in cumulative files. This also results in multiple emails from the eCollege system to the user telling them they've been added or dropped. Users gain or lose access to courses they shouldn't.
- All required fields need to exist in the cumulative file for every record. An error will occur if the file does not contain all the required information or if the data differs for a user.
- Check for **duplicate accounts**. The LoginID is the primary identifier for the user in the eCollege system. If a single user has two different LoginIDs, two separate user accounts will be created.
- Part of the API process may involve splitting up large files—most often cumulative—into "chunks" that represent similar and specific pieces of work, like grouping and processing all Create User requests together, all Update User requests together, all Enrollments together, etc. This is one reason it is important to keep all information for a specific user together with the last, or most current, action last.
- Finally, very large cumulative files are **extremely difficult to troubleshoot** should something go wrong, simply because of the sheer amount of data.

We recommend submitting **incremental** data files, rather than cumulative files, if possible. Incremental files include only changes that have occurred since the last time the file was run. For example, at the end of every XXX minutes/hours/days/weeks you send us all the changes since the last time the information was sent.

Appendix F: XML overview

Like HTML, XML is a markup language based on the concept of applying tags to content and then interpreting and displaying the information “described” by the tags. Unlike HTML, though, XML allows you to define your own markup language with its own set of tags rather than using the pre-defined tags associated with HTML. XML is about making information self-describing using meta-data.

Having a basic understanding of the building blocks of an HTML document will help you to identify the elements you will use to create XML for student enrollment and registration exchange with eCollege.

The following sections will discuss some XML basics, such as elements, the root element, nesting elements, attributes, and schemas. The eCollege SIS for API uses the [IMS Enterprise v1.1](#) schema standard.

Elements

An XML element is the most basic component of your document. Elements can contain elements within elements, attributes, and values. Like HTML tags, XML uses open and close tags.

Basic rules to remember about elements:

- You can only have one root element in an XML document (in this case `<enterprise>`).
- Every element must be closed. This can take the form of a pair of tags, one open and one closed `<enterprise>...</enterprise>`, or a single self-closing tag `<enterprise/>`.
- Element names are case sensitive. `<enterprise/>` is not the same element as `<Enterprise/>`.
- All element names are all lowercase (an IMS schema rule). This also applies to all eCollege extensions.
- Element names cannot contain spaces. For example, `phone number` becomes `<phonenumber>`.

Root Element

The first element defined in an XML document is the root element. All of the content within an XML document is contained within the root element. The purpose of the root element is to specify what type of XML file you are creating. For example, if we create an XML document for a cooking recipe, we might define our root element as `<recipe>`. XML documents begin and end with a root element. You can only have one root element per document.

In an HTML file, the root element would be `<html>`. But because the elements in HTML are predefined, the root element will always be `<html>`; however, in XML you can name the root element anything. Remember, the XML document you will create for eCollege must follow the IMS Enterprise v1.1 schema. The IMS schema defines the root element as `<enterprise>`. (See the section on schemas in this appendix.)

Nesting elements

Elements within another element are considered “nested.” When an element is nested within another element, it is encapsulated by the root (outer) element. Nesting elements is a way of keeping order in an XML document. XML elements must be closed in the

order that they are opened. The outer element is referred to as the parent and the inner elements are referred as children, such as:

```
<parent>
  <child>
  </child>
</parent>
```

Attributes

Attributes provide a way to specify additional information about elements. Attributes consist of a name and value (values are always in quotes), and appear within an element. For example, if you had an element like `<phonenumber>`, you might use an attribute `type` to specify that the number is a home, work, fax, or cell phone number.

Basic rules to remember about attributes:

- All attributes values are contained in double quotation marks. Where `<table width=640>...</table>` is acceptable in HTML, in XML it must be written as `<table width="640">...</table>`
- All attributes must have a value. For example, in HTML `<checkbox checked/>` is valid; in XML it needs to be `<checkbox checked="true"/>`
- Like element names, attribute names cannot contain spaces.

Schemas

When the architecture to an XML document is established, it is saved as its own document known as a schema. XML schemas provide a means for defining the structure, content and semantics of XML documents. In the XML documents you will be creating for eCollege, you will be using the [IMS Enterprise v1.1](#) schema.

Remember, a schema defines the structure, not the format, of the XML document. It consists of rules about which elements are allowed.

Valid or Well-Formed XML document?

XML documents that follow the general rules of XML elements are considered well-formed. XML documents that follow the general rules of XML documents and follow the specific rules defined in their schema are valid. An XML document must be well-formed before it can be valid. Only well-formed XML can display in a browser.

Remember, when an XML document is associated with a schema, such as the IMS schema, you must follow the rules defined in that schema. For example, the root element in the IMS schema is defined as `<enterprise>`. If you were to define the root element in your XML document by any other name, the document would not be considered valid XML.

IMS Elements Implemented by eCollege

Obviously, creating a schema and defining elements to an XML document can become quite a daunting task. For your convenience, we have already created an XML template for you using the element data structures defined in the rules of the IMS schema.

For a description of each of the element data structures you will use to build your XML document, please refer to the *eCollege Annotated Guide to the IMS Specification*. This document can be accessed in PDF format from the SDK or is available from your Client Services representative.

Appendix G: Sample XML data

Following is a sample XML request which conforms to the IMS schema as interpreted by eCollege:

```
<enterprise xmlns="http://schemas.ecollege.com/Common/2006/01/ims_epv1p1.xsd">
  <properties lang="">
    <datasource/>
    <datetime/>
    <extension/>
  </properties>
  <person recstatus="1">
    <sourcedid sourcedidtype="New">
      <source>SchoolOnline</source>
      <id>38641</id>
    </sourcedid>
    <userid pwencryptiontype="" authenticationtype="" password="peskykids"
useridtype="">scoobydoo</userid>
    <name>
      <fn>Scooby Dooby Doo</fn>
      <n>
        <family>Doo</family>
        <given>Scooby</given>
        <partname partnametype="Middlename" lang="">Dooby</partname>
      </n>
    </name>
    <demographics>
      <gender>Male</gender>
    </demographics>
    <email>user@ecollege.com</email>
    <url>www.mysteryinc.com</url>
    <tel teltype="1">tel_1</tel>
    <adr>
      <street>123 Scooby Snack Avenue</street>
      <locality>Scoobyville</locality>
      <region>CO</region>
      <pcode>88888</pcode>
      <country>USA</country>
    </adr>
  </person>
  <group recstatus="1">
    <comments lang=""/>
    <sourcedid sourcedidtype="New">
      <source>ECLG</source>
      <id>SCHOOL.-DENVER.CEAS.</id>
    </sourcedid>
    <grouptype>
      <typevalue level="">Enrollable Node</typevalue>
    </grouptype>
    <description>
      <short/>
    </description>
  </group>
  <membership>
    <sourcedid sourcedidtype="New">
      <source>ECLG</source>
    </sourcedid>
  </membership>
</enterprise>
```

```
<id>SCHOOL.-DENVER.CEAS.</id>
</sourcedid>
<member>
  <sourcedid sourcedidtype="New">
    <source>SchoolOnline</source>
    <id>38641</id>
  </sourcedid>
  <idtype/>
  <role recstatus="1" roletype="01">
    <subrole>2</subrole>
    <status/>
  </role>
</member>
</membership>
</enterprise>
```

Appendix H: Common XML errors

The best way to quickly check and see if you have well-formed XML is to open up the XML in a web browser. It will be broken if there is bad XML or if the format is off. Other errors to check for include:

XML-specific errors	
SourcedID not unique but Login ID is unique	If the Update User Properties option is NOT enabled during setup, this will cause the request to fail.
Unclosed XML tag	All elements, or tags, must be closed, either in a single tag<name/> when the element contains no other elements or text, or with an opening and closing element tag <name>User</name>.
Unsupported character encoding	

Some other general things to consider are:

- An XML document must have one element that contains all of the others. This is called the Root.
- XML is case sensitive. <name/> is not the same as <Name/>, and neither of these are <NAME/>.
- Elements must be closed, either in a single tag<name/> when the element contains no other elements or text, or with an opening and closing element tag <name>Dean James</name>.
- Similarly, every element must be closed in the order that the elements were opened, so <address><street>...</street></address> is valid, but <address><street>...</address></street> is not.
- All attributes must have a value: <checkbox checked="true"/> not <checkbox checked/>
- All attribute values must be enclosed in double quotes <address type="home"/> not <address type=home/>
- Like element names, attribute names cannot contain spaces.