Follett Destiny®

Class Import Converter

Follett

Table of Contents

Getting Started with Class Import Converter

If you prefer to view these instructions in a single PDF file, click here.

In Destiny® Textbook Manager™, adding class information to the textbook information helps you discover textbook shortages and surpluses at each school. It can also improve the efficiency and accuracy of textbook acquisitions and transfers.

Class information consists of:

- The classes being held
- The teacher for each class
- The students who are enrolled in each class

While you can manually enter class information in the Back Office, Destiny Textbook Manager also accepts class schedule data in an XML (extensible markup language) file format. You can upload information about the classes and their sections, and, optionally, assign teachers and patrons to them. You can upload data for one school or the entire district. This information is typically available from your Student Information System (SIS).

The Class Import Converter helps you copy this information into Destiny. It accepts class schedule data in a CSV format and generates an XML file that conforms to the Destiny upload format. The Converter quickly processes very large CSV files and is a replacement for the Transform utility when creating class schedule upload files.

Note: Another name for "comma-delimited data" is "comma separated values" or "CSV." For brevity, this document refers to "comma-delimited input files" as "CSV files."

This document provides an overview of the Class Import Converter process, and guides you through:

- Configuring the Converter
- Converting your CSV data
- Uploading the XML file

Process Overview

Use the Class Import Converter to convert the data from a CSV format into the Destiny XML format, and upload by completing the following steps:

1. Export class schedule data from your SIS to a CSV text file. Make this text file available to Destiny by copying it to a working folder on your Destiny application server.
2. Customize the Converter by editing its properties file. Map the fields in your CSV file to those in Destiny; and set up default values for those fields not in your CSV file.
3. Using the Class Import Converter, convert the class schedule data from the CSV format to the XML format that Destiny can interpret and upload. Run the Class Import Converter process either manually through a command prompt or automatically as a Windows Scheduled Task.
4. Upload the resulting XML file into Destiny either manually – from a command prompt or in the Back Office of Destiny – or automatically as a Windows Scheduled Task.

Before You Begin

A successful XML upload requires that certain data currently exist in the Destiny database. Before you upload, Follett recommends verifying that your database contains the data required by the upload.

If you are uploading more than one type of XML file, make sure to upload them in the correct order:

A. First, import or upload your patron records.

- If your Class Schedules XML assigns teachers, students or both to class sections, the teacher and student records must already exist in the Destiny database. To convert and upload successfully, the CSV file must contain either their District IDs or their Barcode Numbers. The numbers must match those in the patron records in Destiny. Uploading the Class Schedules XML does not add student or teacher records to the database.
- To assign a teacher to a section, the teacher's patron record must be associated with the site where the section is taught. If the site association does not exist at the time of the Class Schedules XML upload, Destiny displays a link in the job summary allowing you to edit the teacher's patron record and add the required site association.
- Each teacher's patron record must have the "Currently Teaching" checkbox selected prior to uploading the Class Schedules XML. You can ensure that your teachers are recognized as teachers in Destiny by including the "IsTeaching" field in your Patrons XML upload.
- Students must be associated with the same site as their classes. The Class Schedules upload cannot transfer patrons between sites.

B. Next, upload your Class Schedules XML.

Note: This document provides information for this step only.

- The file should include information for current classes and sections. If you want to forecast future textbook needs, you can also include information for future classes and sections.

Important:

- By default, the upload deletes pre-existing Section information, including sections with current checkouts. Any checkouts to a deleted section are converted into checkouts to patrons.
  This lets you delete last year's section information while retaining the classes and their textbook associations. If you need to update or add only a few Sections, you can choose to retain Sections not in the XML file.
- All existing Classes are always retained.
- All existing default textbook associations are retained. Default textbooks are those assigned to Classes, not sections, by the district textbooks manager.

C. Finally, upload your Textbook Associations XML. This upload assigns your textbook records to your classes and sections. Your textbook title records must already exist in the Destiny database.

If you are updating existing class schedules, and have existing textbook associations:

- Any class-textbook associations (default textbooks) are retained.
- Any section-textbook associations for deleted sections are deleted.
- Any section-textbook associations for updated sections are retained.

For example, if you are updating section information in the middle of a term, the textbooks already associated with the section remain. If the upload contains no information for a section, and you accept the default option to remove sections not in the update file – thus deleting them – the upload also deletes any textbook associations for that section. It does not, however, delete any textbook records.

Transaction Data

If there are textbook checkouts associated with any deleted sections, the checkout transactions remain, but are not associated with a class section.

Tips for Implementing the Class Schedule XML Upload Process

- Make sure that all patron data (teacher and student) in Destiny matches the data in your SIS.
- Upload the Class Schedules XML to Destiny, allowing the process to add new classes and update existing classes. It will not delete existing classes or remove the textbooks associated with them.
- Associate the textbook titles with the classes and/or sections.
- Automate the process of uploading updated class schedules on a regular (weekly) basis.

CSV File Requirements

Since Destiny cannot directly extract class data from an SIS, you must export it to a CSV file and convert it to XML before Destiny can process the information. You may need to contact your SIS vendor for information about exporting class schedule data to a CSV data file.

It is strongly recommended that all CSV data fields be surrounded by quotation characters to ensure the data with embedded commas can be processed.

For example, class information for a class called "Dollars, Cents, and Sensibilities" will not import correctly without the data being surrounded by quotes.

Change this formatting: 1,Dollars, Cents, and Sensibilities,3,101

To this formatting: "1", "Dollars, Cents, and Sensibilities", "3", "101"

Example CSV file

While there is some flexibility in the format of the CSV file, the following table outlines the optimal structure of the CSV file:

Field	Description	Required	Comments and rules
1	Site ID	Yes	Make sure the Site ID values are mapped correctly to your Site Short Names, either in the properties file of the Converter or in the XSL stylesheet of the Transform.
2	Class name	Yes	**Example:** Algebra I
3	Catalog designation	Yes	The unique identifier for a Class. **Example:** MAT 101
4	Section ID	Yes	In combination with the loan period (Starts and Ends dates, fields 10 and 11), the unique identifier for a Section of a Class at a site. **Important:** If two or more sections have the same Section ID but have different loan periods, Destiny considers them different sections. This allows you to upload information for future school terms while preserving the information for current terms.
5	Teacher number: Barcode or District ID	No	If not included, any existing assigned teacher is removed unless they have checkouts of the associated textbooks. If included, make sure you've included the IsTeaching element with a value of "true" in the Patron XML file. Make sure the properties file of the Converter or the XSL stylesheet of Transform is set correctly for the type of number (Barcode or District ID) in the CSV file. Patron records must already exist in Destiny.
6	Period that the class section meets	Yes	School day period. Must be a number between 1 and 99, inclusive.

Field	Description	Required	Comments and rules
7	Student number: Barcode or District ID	No	If not included, any existing assigned students are removed unless they have checkouts of the associated textbooks. Make sure the properties file of the Converter or the XSL stylesheet of the Transform is set correctly for the type of number (Barcode or District ID) in the CSV file. Patron records must already exist in Destiny.
8	Department	No	If not included – or included but not mapped in the properties file or stylesheet – set to the default department in the properties file or stylesheet. If there is no default, the department is set to "Undefined" in Destiny where you can manually change it. If an incoming department does not exist in Destiny, the department is added.
9	Meets	No	The days of the week that the section meets. Acceptable values are SU, M, TU, W, TH, F, SA. Multiple days must be separated by commas. Must exist in the XML file. If not included in the CSV file, the default values from the properties file of the Converter or the XSL stylesheet of the Transform are used.
10	Starts	Both required if entered	If not included, Destiny uses the start and end dates you specify when you upload the file. For the Converter, the default format is mm/dd/yyyy. You can configure a different format in the properties file. For the Transform, dates must be in the mm/dd/yyyy format.
11	Ends		
12	Loan period description	No	If you are creating loan perids by including new Starts/Ends date ranges, above, you can supply a name for each one "Added During Upload XXXX." Not used for matching.

Important: The rows in the CSV file must be sorted in the following order:

CSV File Sorting Order	
First by	Site ID
Then by	Class name (ClassName)
Then by	Catalog designation (CatalogDesignation or Class ID)
Finally by	Class section (Section ID)

Creating a Working Folder

For ease of use, create a working folder on your Destiny application server containing all the necessary files:

1. From your installation's FSC-Destiny\fsc\bin folder, copy the following files to the working folder:
   - `ClassImportConverter.exe`
   - `ClassImportConverter.l4j.ini.sample`
   - `ClassImport.properties.sample`

Note: If you are not at the server and you can log in as a Destiny Administrator, you can download the files from Destiny.

2. Rename `classimport.properties.sample` to `classimport.properties`.
3. Copy your CSV file to the working folder.

Configuring the Converter

The Class Import Converter is controlled by the settings in its properties file. By default, the name of this file is `classimport.properties`. From your working folder, open `classimport.properties` in any text editor. The file lets you configure the Converter for your input data and consists of several sections:

Teacher and Student Identification

```
#-----------------------------------------------------
# Indicate whether the barcode (true) or DistrictID (false)
# should be the identifier inserted into the output XML
#-----------------------------------------------------

useBarcodeAsTeacherId=false
useBarcodeAsStudentId=false
```

You can identify the teacher of a class section either by their District ID or by their Barcode Number. If your CSV file contains the teacher's Barcode Number, set useBarcodeAsTeacherId to true. If the file contains the teacher's District ID, leave it at false. You can also identify students enrolled in a class section by either their District ID or their Barcode Number. If your CSV file contains the student's Barcode Number, set useBarcodeAsStudentId to true. If the file contains the student's District ID, leave it at false.

Site ID to Short Site Name Mapping

```
#-----------------------------------------------------------------
# SiteID to Destiny short site name mapping -- Add entries
# for each site to be mapped.
# Example: siteid_555=MySite
#-----------------------------------------------------------------

siteid_default=Undefined
```

When the CSV data is converted, the Site ID field in the CSV data must be converted to a valid Site Short Name value. This section defines the relationship between the Site ID and Site Short Name.

When a CSV row is processed, the Converter looks up the Site ID in the properties file. If there isn't an entry for a Site ID, the Converter uses the value of siteid_default.

In the example, if a row contains a Site ID value of 555, the Destiny Site Short Name is set to MySite.

Notes: Do not map more than one Site ID (school code) to a siteShortName.

Because the Converter wraps the Site Short Name in an XML CDATA entity, there is no need for you to add the **<![CDATA[]]** wrapper to ensure properly-formed XML.

Department ID to Department Name Mapping

```
#-----------------------------------------------------------------
# Department ID to department name mapping -- Add entries
# for each department id to be mapped to a department name.
# Example: dept_1=English
#-----------------------------------------------------------------
dept_default=
```

The incoming CSV data may or may not have department information. The entries in this section of the properties file allow the mapping of department identifiers to ones that may be in use in Destiny. Department names are not required in Destiny, and an empty value for dept_default is acceptable.

When the CSV data is imported, the department field in the CSV data can be converted to a Destiny Department Name value. As a CSV row is processed, the Converter looks up the department in the properties file. If there isn't an entry for a department, the value of dept_default is used. If there is no value for dept_default, the class appears under Other in the Department list in Destiny.

In the example, if a CSV row contains a department value of 1, the Department Name is set to English.

Note: Because the Converter wraps the Department Name in an XML CDATA entity, there is no need for you to add the **<![CDATA[]]** wrapper to ensure properly-formed XML.

Default Value for Days of Week that a Session Meets

```
#-----------------------------------------------------------------
#Default value for "meets" when one is not provided
#-----------------------------------------------------------------
meets_days_of_week_default=M,Tu,W,Th,F
```

Each class section in the Class Schedule XML requires an element that indicates the days of the week that the section meets. If the CSV file does not contain this information, the Converter uses the default value defined here. For a successful upload of the XML file, the default value ***cannot*** be empty.

Advanced Configuration in Properties File

The Converter allows flexible formatting of the input and output data fields to accommodate most field data. It is possible to "string together" (concatenate) multiple fields, and convert numerical or date/time information from one format to another. This flexibility introduces a little complexity.

Field Mapping

This section assigns CSV fields to internal fields that will be used to create the XML output file.

This section does not require any changes unless you have special rules in place.

```
#-----------------------------------------------------------------
# CSV fields to Destiny field mapping -- Associates XML fields
# that Destiny uses with incoming CSV fields.
#-----------------------------------------------------------------
field_siteID={1}
field_className={2}
field_classDesignation={3}
field_section={4}
field_teacherNumber={5}
field_period={6}
field_studentNumber={7}
field_department={8}
field_meets={9}
field_starts={10}
field_ends={11}

# LoanPeriodName is optional and rare
#field_loanPeriodName={12}
```

Data Type Definitions

The final section defines the data type of the CSV input fields. Unless an output field uses a date or number formatter, it is unnecessary to make any changes in this section.

```
#-----------------------------------------------------------------
# CSV input fields type mapping -- Defines the type and
```

```
# format of the data in a given CSV field.
#-----------------------------------------------------------------
field_type_1=string
field_type_2=string
field_type_3=string
field_type_4=string
field_type_5=string
field_type_6=string
field_type_7=string
field_type_8=string
field_type_9=string
field_type_10=string
field_type_11=string
# LoanPeriodName is optional and rare
#field_type_12=string
```

Examples:

A

Given this CSV row: "001","1","2","20120115"

Results in the fields being set to these values:

{1} = 001

{2} = 1

{3} = 2

{4} = 20120115

After configuring the mapping, concatenation, and formatting, the output values would change:

Mapping in properties file	Resultant output
`field_test={2}`	`1`
`field_test={2}{1}`	`1001`
`field_test=IBE{2}-{1}`	`IBE1-001`
`field_test={3,number,00000000}`	`00000002`
`field_test=P{3,number,00000000}`	`P00000002`
`field_test={4,date,MM/dd/yyyy}`	`01/15/2012`

If you use the date or number formatters, you must make entries in the section that defines the field types and formats used in the input CSV file.

In the example above, the entries would look like this:

```
field_type_1=string
field_type_2=string
field_type_3=number,0;0
field_type_4=date,yyyyMMdd
```

B

For another example, some districts construct the teacherNumber by appending the Site ID:

```
field_teacherNumber={5}{1}
```

C

An example of taking the student number, right-justifying it, and adding a P at the beginning to create a Destiny patron barcode. 1 becomes P0000001:

```
field_studentNumber=P{7,number,0000000}
field_type_7=number,0;0
```

Additional information about the formatting characters that can be used for date or number formatters can be found at the following locations:

Date: http://download.oracle.com/javase/1.5.0/docs/api/java/text/SimpleDateFormat.html

Number: http://download.oracle.com/javase/1.5.0/docs/api/java/text/DecimalFormat.html

Running the Converter

Once you have copied `ClassImportConverter.exe`, `ClassImportConverter.l4j.ini.sample` and `classimport.properties.sample` to your working folder, renamed `classimport.properties.sample` to `classimport.properties`, and edited the properties file as needed, you can run the Class Import Converter.

1. Open a command prompt, and change to your working directory.
2. Enter the following command with the parameters in the correct order:

```
ClassImportConverter inputFile outputFile propertyFile fileEncoding
```

EXAMPLE
C:\FSC-Destiny\fsc\bin> classimportconverter newclassinfo.csv newclassinfo.xml classimport.properties UTF-16

Parameter name	Required	Description
inputFile	Yes	CSV input file
outputFile	Yes	XML input file
propertyFile	Yes	Properties file, default provided is classimport.properties
fileEncoding	No	Specifies the character encoding used to read the input file. For some Microsoft SQL Server csv export files, UTF-16 may be required. Defaults to windows-1252 (Windows Latin-1)

3. Note the name and location of the log file, named `<outputFileName>.xml.log`, that the command window displays.
4. Review the log file for any exceptions.

Once you have successfully created an XML output file, you can continue to run the Class Import Converter manually, or automatically as a Windows Scheduled Task.

Possible Errors

No JVM Installed

If you run the Converter and there is no Java™ Virtual Machine available, the Converter launches a browser window that lets you download and install the required JVM™.

Data Ordering Issues were Detected

This message indicates that the incoming CSV file is not sorted properly.
Before running the Converter, you must ensure that the first four fields are sorted correctly.
The Converter can sort the CSV file for you:

1. At a command prompt, enter the following command, using the correct file names for your input and output files:
   `ClassImportConverter unsorted.csv sorted.csv /s`
2. After sorting, run the Converter using the newly sorted CSV file.

Remember, this sort option is a convenience feature and should not replace the requirement that the input CSV file be ordered by the first four fields. There is no assurance that you will have the required memory to sort very large CSV files.

Out of Memory Error

By default, the Converter uses the default amount of memory set up for the JVM. If you encounter this error when sorting your CSV file, perform the following steps:

1. Rename `ClassImportConverter.l4j.ini.sample` to `ClassImportconverter.l4j.ini`. This enables the Converter to use more RAM than the default amount allocated for the JVM.
2. If necessary, edit `ClassImportConverter.l4j.ini` to modify the maximum memory to use. Change the 1024 in the statement, shown below, in the .ini file. -Xmx1024m (tells the JVM to use 1GB of RAM)
3. Run the Converter again.

Depending on the amount of memory available, it should be possible to sort all but the largest input files.

Automating the Conversion

To automatically convert your class data on a regular basis, you can use the Windows Scheduled Tasks and a batch file.

Initially, you'll need to configure your batch file. The Converter can return error codes as ErrorLevel from the command line utility. You can test for these values in a batch file, and then the batch file can take appropriate action based on the value returned.

Error Codes

ERR_NONE = 0
ERR_NO_INPUT_FILE = 1
ERR_FILE_NOT_FOUND = 2
ERR_IO_READ = 3
ERR_IO_WRITE = 4
ERR_SORTING = 6
ERR_XML_PARSE = 7
ERR_INVALID_CHARACTERS_IN_XML = 8

Batch File Example

As an example, the following batch file incorporates the error codes to detect problems with the conversion and skips the upload of a faulty XML file.

```
-----------------------------------------------------------------

rem Execute the conversion utility
ClassImportConverter input.csv output.xml classimport.properties
rem Check the status of the conversion and continue if no errors
IF ERRORLEVEL 8 GOTO InvalidXMLCharacters
IF ERRORLEVEL 7 GOTO ParseError
IF ERRORLEVEL 6 GOTO SortError
IF ERRORLEVEL 4 GOTO WriteError
IF ERRORLEVEL 3 GOTO ReadError
IF ERRORLEVEL 2 GOTO FileNotFoundError
IF ERRORLEVEL 1 GOTO NoInputFileError

rem Otherwise -- success
rem continue processing and then exit
goto END

:InvalidXMLCharacters
echo Invalid characters were found in the output XML file
goto END

:ParseError
echo Unable to parse the output XML file
goto END

:SortError
echo A processing error occurred during the file sort phase
goto END

:WriteError
echo Unable to write output file to disk
goto END
```

```
:ReadError
echo Unable to read file from disk goto END
:FileNotFoundError
echo CSV input file was not found
goto END
:NoInputFileError
echo No CSV file was specified to convert
goto END
:END
```

Uploading XML File to Textbook Manager

You can upload the final XML file to Destiny in either of two ways:

In the Back Office

1. Log in to Destiny as a district textbooks manager or a site administrator.
2. Open the **Upload Changes** tab of **Update Classes** in the **Back Office**.
3. Select **Class schedules** from the list.
4. Select to either remove or retain existing Sections that are not in the XML file.
5. Select start and end dates for the Sections whose dates are missing from the XML file.
6. Click **Browse** to find and select your XML file.
7. Click **Update**.
8. When the Job Manager opens, review the job summary for any errors or exceptions.

Through a Command Line

1. Log in to the Destiny server.
2. Find the following files in the `FSC-Destiny\fsc\bin` folder:
   - updateclasses.properties.sample
   - updateclasses.properties.sample
3. If you have not yet done so, rename `updateclasses.properties.sample` to `updateclasses.properties`.
4. Open `updateclasses.properties`.
5. Enter the Destiny username and password for a user with the permission to upload class schedules.

   The following users can upload class schedules:
   - Destiny Administrator
   - District textbooks manager
   - A site user granted the permission, *Update class information*
6. Save and close the file.
7. Open a command prompt.
8. Change to your working folder.
9. Enter the command in the correct order:

```
UpdateClasses configurationFile inputFileName contextName removeSections startDate
endDate
```

Example: C:\FSC-Destiny\fsc\bin> updateclasses updateclasses.properties classinfo.xml district_123 no 8/15/2017 6/15/2018

10. When the upload completes, open the Job Manager, and review the job summary for any exceptions.

To automate the class schedules upload on a periodic basis, create a Windows Scheduled Task.