Patron Import Converter - Getting Started

While you can manually enter patron information or import a comma-separated values (CSV) file in the Back Office at the site level, the standard method for updating patron records in an XML (extensible markup language) file format at the district level. This capability offers the following advantages:

- You can upload data for one school or an entire district in one file.
- You can add, update, transfer, or delete patrons.
- You can compare the previous and current CSV files, and then convert and upload only the updated records.
- You can automate the conversion and upload processes.

You can use the Patron Import Converter to change patron data that's in a comma-delimited format to an XML file that conforms to the Destiny patron file format. The Converter can quickly process very large CSV files, and it is meant to be a general replacement for the Transform utility and the Patron Conversion utility.

NOTE The Converter does not accept files with tab-delimited or fixed-length fields. It does accept comma-delimited input files, or CSV files.

This document provides an overview of the Patron Import Converter (PIC) process, and guides you through configuring the Converter, converting your CSV data, and uploading the XML file. The information and instructions will assist you whether you are doing a one-time manual conversion and upload of data into Destiny, configuring the PIC for use with an OS scheduled batch file update (if your CSV data is not local to the Destiny application server), or configuring Destiny's Scheduled Patron Update feature.

Once you've configured the Converter, you can automatically convert and upload your patron data on a regular basis, using Destiny's Scheduled Patron Update feature or by using the OS schedule batch files.

Process Overview (for Updating Manually)

With the Patron Import Converter, you'll convert data from a CSV format into the Destiny XML format and upload it using the following procedures. For those working with the Scheduled Patron Update within Destiny or the OS scheduled batch files, you will continue to use the appropriate scheduling process:

- Export patron data from your student management system (SMS) to a CSV text file.
- 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.
  To edit the properties file, you can either open the Destiny Patron Import Converter application that has a graphical user interface, or use a text editor.
- Convert the patron data from the CSV format to the XML format that Destiny can interpret using the Patron Import Converter.
- Upload the resulting XML file into Destiny in the Back Office of Destiny.

CSV file requirements

Destiny cannot directly extract the patron data from an SMS. You must export the data to a CSV file and convert it to XML format before Destiny can process the information. You may need to contact your SMS vendor for information about exporting patron data to a CSV data file.

There is substantial flexibility in the format of the CSV file. The only requirements for the CSV file are:

- The CSV file must contain at least one field that maps to a Destiny patron record field.
- The CSV file must contain either a District ID, or a barcode number and Site Short Name, to uniquely identify each patron.
- The CSV file must contain a Last Name field.

About character encoding

For the CSV file, the Converter uses Windows Latin (Cp1252) encoding by default. If it

encounters any illegal characters, the conversion process fails.

You can use any of the following character encodings for the CSV file:

- ASCII
- UTF8
- Cp1252
- UTF-16
- ISO8859_1

Other encodings are available. However, they have not been tested with the application. You can view the full set of encoding values supported at:
http://-download.oracle.com/javase/1.5.0/docs/guide/intl/encoding.doc.html.

NOTE If the CSV file is not in Windows Latin (for example, a SQL Management Studio query result saved as CSV), the converter may not be able to preview the data. In this case, copy and paste a line into Notepad and use this for previewing the data.

The Converter always creates an XML file with UTF8 encoding.

Available fields

Click here for details

The following table defines the fields that are available for updating. Besides the required fields mentioned above, you can include any or all of these fields in your CSV file.

Destiny Field	Configuration File Field	Max length	Comments
Site Short Name	field_siteShortName	10	*Required.* Must match site short name, or an Alias, entered in district configuration. If not in the CSV file for a single school, enter it in the properties file.
Barcode	field_barcode	14	*Required* when adding a patron. Must be unique in a school. Value of “autoassign” causes Destiny to assign the next available barcode number to the patron. Cannot use “autoassign” when matching by Barcode and Site short name.
District ID	field_districtID	40	*Optional* if not used for matching or required on Edit District page. Must be unique in the district.
Last Name	field_lastName	42	Required.
First Name	field_firstName	42	
Middle Name	field_middleName	42	
Nickname	field_nickname	30	
Patron Type	field_patronType	30	If not specified, default is used. If unknown, Patron Type is created. S and F not accepted for Student and Faculty.

Destiny Field	Configuration File Field	Max length	Comments
Access Level	field_accessLevel	30	If not specified, default is used.
Asset Group	field_assetGroup	30	Only applies to Destiny Asset Manager sites. If not specified, default is used.
Status	field_status	1	A, I, or R (Active, Inactive, or Restricted) Defaults to A if blank or invalid.
Gender	field_gender	1	M, F, U Defaults to U (Unspecified) if blank or invalid.
Homeroom	field_homeroom	90	
Grade Level	field_gradeLevel	30	
Card Expires	field_cardExpires	10	Format as required on Edit District page
Acceptable Use Policy on File	field_isAcceptableUsePolicyOnFile	5	True, T, False, F, Yes, Y, No, or N
Is Teacher	field_isTeacher	5	True, T, False, F, Yes, Y, No, or N
User Defined 1	field_userDefined1	30	
User Defined 2	field_userDefined2	30	
User Defined 3	field_userDefined3	30	
User Defined 4	field_userDefined4	30	
User Defined 5	field_userDefined5	30	
Graduation Year	field_graduationYear	4	YYYY
Birth Date	field_birthdate	10	Format as required on Edit District page.
User Name	field_username	50	*Optional.* Unique within the district. Required if updating the Password field.
Password	field_password	50	
Email 1	field_emailPrimary	128	
Email 2	field_emailSecondary	128	
Address 1 Line 1	field_addressPrimaryLine1	42	
Address 1 Line 2	field_addressPrimaryLine2	42	
Address 1 City	field_addressPrimaryCity	42	
Address 1 State	field_addressPrimaryState	42	
Address 1 Postal Code	field_addressPrimaryZipCode	20	
Address 1 Phone 1	field_addressPrimaryPhoneNumberPrimary	30	
Address 1 Phone 2	field_ addressPrimaryPhoneNum-berSecondary	30	
Address 2 Line 1	field_addressSecondaryLine1	42	

Destiny Field	Configuration File Field	Max length	Comments
Address 2 Line 2	field_addressSecondaryLine2	42	
Address 2 City	field_addressSecondaryCity	42	
Address 2 State	field_addressSecondaryState	42	
Address 2 Postal Code	field_addressSecondaryZipCode	20	
Address 2 Phone 1	field_ addressSe-condaryPhoneNumberPrimary	30	
Address 2 Phone 2	field_ addressSe-condaryPhoneNumberSecondary	30	

Accessing the Converter

If you are using the PIC for configuring Destiny's Scheduled Update feature, you can load the PIC directly from the Schedule Patron Update job page. If you are working to configure the Converter for use with the Troubleshooting the scheduled Patron XML Update process, you must complete the following steps on the Destiny application server.

If you are working to do a one-time manual conversion and upload, create a working folder on your local computer.

1. Navigate to, or obtain, the files:
   - Click the link for the Patron Import Converter on the Schedule Patron Update job page, and choose to run or download and run based on your browser configuration.
   - If you are working on the server, browse to your installation's `\FSC-Destiny\fsc\bin` folder. It should contain the following files:
     ```
     PatronImportConverter.exe
     PatronImportConverter.l4j.ini.sample
     patronimport.properties.sample
     ```
   - If working on your local computer, download the files from Destiny and unzip them into your working folder.
2. If you're working at your local computer, copy your CSV file to the working folder.

This next section guides you through customizing the Patron Import Converter (editing the properties file) to suit your CSV file and your preferences.

If you prefer to edit the properties file directly, skip to Configuring the Converter in a text editor.

Configuring the Converter in the application

With the Patron Import Converter application, you can open, edit, and save your properties file. Once you've selected the necessary matching/skipping/deleting parameters and saved the configuration, you can also run the CSV-to-XML conversion from the application.

The easiest way to configure the properties file is to open the CSV file. If you do not have the CSV file but do know its contents and format, you can configure the properties file without opening the CSV file.

Configuring the properties consists of the following steps:

- Mapping the fields in the CSV file to those in the Destiny patron record.
- Specifying the format for dates, changing incoming CSV field values to those that Destiny uses, and applying special processing (concatenating and formatting) where needed.

- Defining the way the XML Upload matches patrons and under what conditions the Upload is to update or delete patrons.

Getting started

1. Open the application using either of the following methods:

   - In your working folder or FSC-Destiny\fsc\bin folder, double-click `PatronImportConverter.exe`.
   - From a command line, enter the following command without specifying any parameters: `PatronImportConverter`

   The Converter first searches your computer's registry for a Java Virtual Machine (JVM). If it cannot find one, a browser window opens, allowing you to download the free JVM from Oracle. The website provides complete instructions.

   After successfully downloading the JVM, open the application as instructed above.

2. If you have the CSV file, click **Browse** to select it.

   - The first several records appear on the *Data Preview* tab:
   - If you're not opening a CSV file, enter the *Number of fields on each row*.
   - If you have opened a CSV file, the application enters the *Number of fields on each row* automatically and grays out (disables) the field.
   - To re-enable the field and clear the Preview pane, click **Clear** in the lower corner.

3. If your CSV file contains any header rows, enter the row number of the first patron record next to *Data begins on row*.

> **NOTE** If you're configuring the Converter for use in an automated upload, leave the value at `1`.

Mapping the fields

1. Open the **Field Mapping** tab.

2. For each field that exists in your CSV file, locate the corresponding *Destiny Field*, and then click in its adjacent *CSV Input Field*. From the list that appears, select the corresponding field number. Review the bottom pane to help map the CSV field to the correct Destiny field.

Then verify your selection in the *Sample Output* column. You can map to any or all fields in the patron record.

You can map the same CSV field to multiple Destiny fields.

Empty fields

You can also define the Converter's action when it encounters an empty CSV field that you mapped to a Destiny field:

- To clear the existing value in a Destiny field that's missing an incoming CSV value, select the checkbox in the *Include Empty Tag* column. This inserts an empty field (for example, `<FirstName></FirstName>`) in the XML file that clears the field in Destiny.

  In the above bullet (example), the Upload will clear the Homeroom and Grade Level fields for records without incoming values in them.

- To preserve the existing value in a Destiny field that's missing an incoming CSV value, deselect the checkbox in the *Include Empty Tag* column. This inserts no element in the XML file, thus preserving the existing value in Destiny. In the example above, the Upload will preserve all the existing values except Homeroom and Grade Level for records without incoming values in them.
- You can also have the Converter insert the contents of another field when it encounters an empty field. See Mapping empty fields.

Formatting the fields

If you need to specify the format for dates, change the incoming CSV field values to those that Destiny uses, or apply special processing to any fields, use the options in the *Format* column.

Formatting Date fields

For a field containing a date, click in its *Format* column and select DATE from the list that appears. The *Choose Date Format* dialog opens.

1. Review the Sample Date String from the CSV file and, from the *Month*, *day*, and *year order* list, select the order that it has.
2. In the *Date Format* column, locate and select the format that best matches that in the CSV file.
3. Click **OK** to close the dialog.
4. Back on the *Field Mapping* tab, verify your choice by selecting any of the records in the bottom pane of the *Field Mapping* tab and reviewing its date in the Sample Output column.

If you need to reopen the *Choose Date Format* dialog, double-click the field name in the *Destiny Field* column on the *Field Mapping* tab.

Mapping field values

For a field whose values require mapping, select MAP in its *Format* column. The *Field Mapping Values* dialog opens.

1. Click **Add Mapping**.
2. In the table row that appears, enter the value from the CSV file in the *When the CSV input data equals* column on the left.
3. Enter the value to which you want it changed in the *Change it to* column on the right. Continue adding all possible values. Click **OK** when you are done with this field.
4. *Defining a default value is optional.*To define a default value for this field—a value to use if the CSV data does not match any other defined mapping values—enter *default* in the left column, and your default value in the right column.
5. Verify your mapping by selecting any of the records in the bottom pane of the *Field Mapping* tab and reviewing its mapped value in the *Sample Output* column.

EXAMPLE Your CSV file contains the numbers 7 through 12, in a Grade Level field and you've mapped this field to the Destiny's Grade Level field.

You can also map this field to Access Level: You'd MAP 7 and 8 to "MS student", and 9 through 12 to "HS student". The Upload then assigns the "MS student" Access Level to all 7th and 8th graders and "HS student" to all 9th to 12th graders.

In case any incoming records don't contain a Grade Level, you'd also define a "default" value of "MS student". Note that the Access Levels "MS student" and "HS student" must already exist in Destiny.

If you need to reopen the Field Mapping Values dialog, double-click the field name in the *Destiny Field* column on the *Field Mapping* tab.

Entering custom mapping and formatting

For a field whose values require special formatting, concatenation, or added text, select CUSTOM in the Format column. The Edit Custom Format dialog opens.

1. Enter your custom formatting string.

IMPORTANT Do not add quotation marks or CDATA statements. The Converter inserts the necessary CDATA statements into the XML file.

2. Click **OK**.
3. If your CSV file is open, verify your formatting by selecting any of the records in the bottom pane of the *Field Mapping* tab and reviewing its formatted value in the Sample Output column.

> **EXAMPLE** Your CSV file contains a field for the Homeroom number and a field for the Homeroom advisor. You'd like to enter both in Destiny's Homeroom field.
>
> You'd specify the field numbers, surrounding them with braces: {6}{13} and insert text (a "literal string") to improve readability: " - Room "

To update a field to the same value for all incoming patron records, select CUSTOM in the Format column. In the Edit Custom Format dialog that opens, enter a literal string and click **OK**.

If you need to reopen the *Edit Custom Format* dialog, double-click the field name in the *Destiny Field* column on the *Field Mapping* tab.

For more information on concatenation, substrings, and date or number formatting, see Configuring the Converter in a text editor.

Matching patrons

Before importing them, Destiny checks to see if the incoming patrons already exist in your database.

Destiny can determine whether an incoming patron is a match to an existing patron by comparing either the District ID or the Site short name and the patron's barcode number.

> **NOTE** We strongly recommend that you use *District ID* (the unique number for the patron in the district). This enables the conversion/upload process to transfer patron records from one school to another.

1. Open the **Patron Matching** tab.
2. Under *Match patrons using their*, select the unique identifier your school or district uses.

Skipping, deleting, and associating patrons

By default, the Converter adds information to the XML file that instructs Destiny to add or update each incoming patron record.

In XML terms, it adds the attribute, action="addmodify", to each Patron element. You can, however, configure the Converter to skip or delete patron records, or associate them with an additional school, based in criteria you define.

Once you define the conditions, the Converter can identify the records and perform the correct action. A Patron XML file can have any combination of add/modify, delete, and multipleassociation actions.

To skip patrons

1. On the *Patron Matching* tab, open the **Skip Rules** tab.
2. At the bottom, click **Add**.
3. Click under *Destiny Field* and select the field from the list.
4. Click under *Contains this value* and enter the value that will make the Converter skip the record.
5. Repeat steps 2 - 4 for any additional conditions.
6. Decide whether a record must meet *ANY* or *ALL* conditions to be skipped and select that option under *Skip patrons when*.

As the Converter processes each CSV record, it tests the data. If the data matches the conditions you define, the Converter skips the record.

NOTE To skip a header record, configure the table to skip the record when encountering the header values.

EXAMPLE In the dialog above, the Converter will match patron records by comparing the District ID in the existing and the incoming record. The Converter will skip any patron whose Status is *Inactive*.

To delete patrons

1. On the *Patron Matching* tab, open the **Delete Rules** tab.
2. At the bottom, click **Add**.
3. Click under *Destiny Field* and select the field from the list.
4. Click under *Contains this value* and enter the value that will make the Converter instruct Destiny to delete the record.
5. Repeat steps 2 - 4 for any additional conditions.
6. Decide whether a record must meet ANY or ALL conditions to be deleted and select that option.
7. To update a patron record before deleting it, select the **Update before delete** checkbox. (The Destiny database retains archived information for all patrons.)

As the Converter processes each CSV record, it tests the data. If the data matches the conditions you define, the Converter instructs Destiny to delete the patron record. In XML terms, it adds the attribute, action="delete", to each Patron element.

If you selected the check box, the Converter adds a Patron element with the attribute, action="delete" immediately after a Patron element with the attribute, action=" addmodify", for each qualifying patron.

EXAMPLE In the dialog, above, the Converter will match patron records by comparing the District ID in the existing and the incoming record. The Converter will instruct Destiny to delete any patron whose Grade Level is 12 or whose Graduation Year is 2010.

Associating

While most patrons have a single site association—the school where they teach or attend classes—the Converter can associate patrons with additional sites. You may want to do this for students who attend classes, or faculty who teach classes, at multiple schools.

When you upload a particular patron record for the first time, a patron's site association is determined by the Site Short Name in the patron record. On subsequent uploads, if the Site Short Name is different, Destiny updates the Site Short Name, in effect, transferring the patron to that school. Once you set up your Multiple Association Rules, however, the Converter can instruct Destiny to add a site association to the patron record instead of replacing it. Destiny designates the first site association as the Primary site and any additional associations as Secondary sites.

To associate patrons

1. On the *Patron Matching* tab, open the **Multiple Association Rules** tab.
2. At the bottom, click **Add**.
3. Click under Destiny Field and select the field from the list.
4. Click under Contains this value and enter the value that will make the Converter instruct Destiny to add—not update—a site association.
5. Repeat steps 2 - 4 for any additional conditions.
6. Decide whether a record must meet ANY or ALL conditions to be associated with multiple sites and select that option.

EXAMPLE In the dialog above, the Converter will match patron records by comparing the District ID in the existing and the incoming record.

The Converter will instruct Destiny to add a site association to any patron whose Access Level is Teacher or whose Patron Type is Faculty. In XML terms, it adds the attribute, action="multipleassociation", to each Patron element.

Finishing up

Now that you've configured the Converter, click **Save** to save the configuration in the properties file. You'll need to provide a filename to save the settings:

- If you're working on your local computer, provide a name of your choosing, such as `studentupdate.properties`.
- If you're working with an existing configuration used by Destiny's Scheduled Patron Update feature, save the changed you've made with the existing `patronimport.propertiesfile` name
- If you're working on the server, configuring for automated processing, make sure that you specify the correct file name as defined in the PATRON_PROP parameter (line 57) of the `process_patrons.bat` file.

If you are ready to run the Converter, skip to Running the Converter.

Configuring the Converter in a text editor

If you don't need to load all of the files into the user interface, you can work with the PIC configuration in a text editor window. It is easier to make simple changes this way.

If you are working with a configuration file used with the OS scheduled batch file process, you can open the appropriate `.properties` file from the disk with Notepad.

If you are working with Destiny's Scheduled Patron Update feature, edit the update job, and click the Edit button beside the "Change properties file" heading. When done, click Save at the edit window and click Save again on the job definition page.

IMPORTANT Field names are case-sensitive.

Site short name

This block does not normally need attention; you can leave it commented out.

Use this section when you're importing a single site and there is no site information associated with each row in the incoming CSV file. When the Converter encounters such a CSV file, you can define a default site short name, and the Converter will assign this default site to all incoming records.

```
#-------------------------------------------------------------
# Default short site name
# Uncomment the next line and adjust the value if CSV file
# does not contain site short name information.
#-------------------------------------------------------------
#SiteShortNameDefault=
```

Matching

This section specifies whether the patron is referenced by the District ID field or by a combination of the patron's Barcode and the Site short name of the site of the patron.

Set this value to true if the patrons should be matched using the District ID.

IMPORTANT Each line of the CSV file is required to have a District ID if this field is set to true or must have a Barcode and Site short name if the field is set to false.

```
#-------------------------------------------------------------
# Matching mode # true=match by District ID
# false=match by barcode/short site name
```

```
#-------------------------------------------------------------
MatchUsingDistrictID=true
```

Rows to skip

Some CSV files may contain one or more header rows that precede the data.

The InputRowsToSkip entry configures the number of rows of the CSV file that the Converter skips when converting the data.

If you're doing a manual one-time upload, you can set the value appropriately based on whether your file has a header (set to 1) or it does not (set to 0).

If you're configuring the Converter for use in an automated upload, leave the value at 0.

NOTE To quickly perform a test conversion on a very large input file, you can set the value to a larger number, causing the Converter to skip most of the rows of the file. For example, with a CSV input file containing 100,000 rows, by setting InputRowsToSkip to 99000, the Converter will only process the last 1,000 rows of the file.

```
#-------------------------------------------------------------
# The number of CSV rows to skip before the data begins
#-------------------------------------------------------------
InputRowsToSkip=0
```

Rules for skipping records

When processing a CSV file, it may be necessary to ignore records based on criteria specified in this section. As the Converter processes each CSV record, it tests the record data. If the data matches the conditions, the Converter discards the record. The Converter writes no data to the XML file for skipped rows.

You can apply any number of conditions for each record read.

- When the SkipMode value is `or`, the Converter skips the record if any of the conditions are true.
- When the value is `and`, the Converter skips the record if all of the conditions are true for the record.

In the example below, the Converter skips any record whose field_districtID value is empty or the fourth CSV field equals Inactive.

NOTE In most cases, the Converter tests the CSV fields (for example, {1},Inactive) instead of the data that it will insert into the XML file (for example, field_status,Inactive).

There may be cases where it is necessary to test the contents of the XML data instead.

EXAMPLE field_districtID may be constructed by combining parts of CSV fields 1 and 2.

The resulting field_districtID value would be tested in this case, as it may not be possible to test fields 1 and 2 separately.

```
#-------------------------------------------------------------
# Define the rules for when a record should be skipped.
#-------------------------------------------------------------
SkipMode=or
SkipWhen_1=field_districtID,
SkipWhen_2={4},Inactive
```

Rules for deleting records

When processing a CSV file, it may be necessary to instruct the Upload to delete records based on criteria specified in this section. As the Converter processes each CSV record, it tests the record data. If the data matches the conditions, the

Converter instructs the Upload to delete the patron (action=delete).

You can apply any number of conditions for each record read.

- When the DeleteMode value is `or`, the action attribute is set to delete if any of the conditions are true.
- When the value is `and`, all of the conditions must be true for the patron to be marked for deletion.

In the example below, the Converter marks any record for deletion whose ninth CSV field value equals Inactive and whose `field_graduationYear` value equals 2010.

If you'd like to update a patron record before deleting it, set UpdateBeforeDelete to true. This causes the Converter to add a Patron element with the attribute, action="delete", immediately after a Patron element with the attribute, action="addmodify", for each qualifying patron.

```
#-------------------------------------------------------------
# Define the rules for when a record should be deleted.
#-------------------------------------------------------------
UpdateBeforeDelete=false
DeleteMode=and
DeleteWhen_1={9},Inactive
DeleteWhen_2=field_graduationYear,2010
```

Rules for multiple associations

When processing a CSV file, it may be necessary to instruct the Upload to add a site association instead of updating the existing site association. As the Converter processes each CSV record, it tests the record data. If the data matches the conditions, the Converter instructs the Upload to add a site association (action=multipleassociation). This action is a modified addmodify action as the other fields in the record are updated.

You can apply any number of conditions for each record read.

- When the AssociationMode value is `or`, the action attribute is set to multiple association if any of the conditions are true.
- When the value is `and`, all of the conditions must be true for the patron to be associated with more than one site.

In the example below, the Converter adds an association to any record whose field_accessLevel value equals Teacher or whose field_patronType value equals Faculty.

```
#-------------------------------------------------------------
# Define the rules for when a site association is added.
#-------------------------------------------------------------
AssociationMode=or
AssociationWhen_1=field_accessLevel,Teacher
AssociationWhen_2=field_patronType,Faculty
```

Field mapping

This section defines the contents of Patron XML data. The table at the beginning of this document lists the XML fields that you can define. You do not need to define all of the elements.

If a field is not specified, the Converter does not include it in the XML output file.

If a field is specified without assigning the contents of the field, the Converter inserts an empty XML element for that field.

IMPORTANT The first field is numbered 1, not 0, and that field names are case-sensitive.

Keep and Omit

The Keep and Omit switches correspond to the I*nclude Empty Tag* checkbox in the application.

- To clear the existing value in a Destiny field that's missing an incoming value, add keep to the fieldmapping statement, such as `field_firstName={5,keep}`.
  This inserts an empty element in the XML file (such as `<FirstName></FirstName>`) that clears the field value in Destiny.
- To preserve the existing value in a field that's missing an incoming value, add omit to the field-mapping statement, such as field_firstName={5,omit}. This inserts no element in the XML file, thus preserving the existing value in Destiny.
- To preserve the existing value in a field, you could also omit the field altogether in the configuration file; then the Converter would not include that field in the XML output file.

In the following example, eight XML fields are defined.

The first five fields map to specific CSV field numbers. The omit switch causes the Converter to omit any empty field from the XML file, thus preserving the existing value.

The content of the sixth field, field_userDefined1, consists of the string literal, Fall 2011, which the Converter will insert in each Patron element created.

The Converter inserts an empty element in each Patron element, and the Upload clears any existing data in that field in the Patron record in Destiny.

```
#-------------------------------------------------------------
# CSV fields to Destiny field mapping -- Associates XML fields that Destiny uses with
incoming CSV fields. For special
# processing (concatenated and formatted fields), please refer to the configuration guide.
#-------------------------------------------------------------
field_siteShortName={1,omit}
field_barcode={2,omit}
field_districtID={3,omit}
field_lastName={4,omit}
field_firstName={5,omit}
field_userDefined1=Fall 2011
field_userDefined3={9,map_userDefined3,omit}
```

Mapping empty fields

To have the Converter insert the contents of another field when it encounters an empty field, use the ISEMPTY command on a separate line. Be sure to specify all parts of the command:

- The ISEMPTY command
- The field to evaluate
- The field to insert if the evaluated field is blank
- The field to insert if the evaluated field is not blank

EXAMPLE In this example, the Converter will populate field 7 with the contents of field 6 if field 7 is blank. Otherwise, it will populate field 7 with the contents of field 7:

```
ISEMPTY({7},{6},{7})
```

IMPORTANT Be sure to format the command as shown in the example, including all brackets, commas, and parentheses.

Custom mapping and formatting

You can map any incoming data for the Destiny XML fields to a list of values defined in the properties file. Common examples are mapping codes (123) to strings (“Einstein Academy”) for such fields as Site Short Name, Gender, Patron Status, and Access Levels.

EXAMPLE Using the example below, the Converter will change "123" to "Einstein", "456" to "Newton", and any other value to "Unknown".

```
#-------------------------------------------------------------
# Define the specialized mapping values for the input data
# in the CSV file.
#-------------------------------------------------------------
map_userDefined3_123=Einstein
map_userDefined3_456=Newton
map_userDefined3_default=Unknown
```

Number of fields and data types

The Converter needs to know how many fields are in each row of the CSV file. It also needs to know if any of the input fields must be mapped to a different data type for formatting.

The field_type_total field is required and must match the number of fields in each data row of the CSV file. Do not count the number of fields in the header row when specifying this value.

If numeric or date formatting is required, the format of the incoming CSV data for the field is required for proper conversion.

If a field-type entry is not present for a CSV field, it is treated as a string field and used as-is.

EXAMPLE In this example there are 38 fields in each row of the CSV file, and the eighth CSV field is to be treated as a date in the form of yyyyMMdd (e.g., 20070620).

```
#-------------------------------------------------------------
# CSV input fields types -- Defines the type and format of
# the incoming data in a given CSV field. Please refer to
# the configuration guide for valid values.
#-------------------------------------------------------------
field_type_total=38
field_type_8=date,yyyyMMdd
```

Advanced configuration

To accommodate most field data, the Converter allows flexible formatting of the input and output data fields. It is possible to "string together" (concatenate) part or all of multiple fields, convert numerical or date/time information from one format into another, and define regular expressions (regex) and conditional templates.

EXAMPLES Let's begin with some examples.
Given this CSV row:
"001","1","2","20120115, "8153448700", "342/123.12"

Results in the fields being set to these values:

```
{1} = 001
{2} = 1
{3} = 2
{4} = 20120115
{5} = 8153448700
{6} = 342/123.12
```

Custom formatting

By applying custom formatting, you can produce values such as the following:

Mapping in properties file	Resultant output
field_test={2}	1

Mapping in properties file	Resultant output
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
field test=({5,substring (1,3)}) {5,substring (4,3)}-{5,substring (7,4)}	(815) 344-8700

Date/time formatting

If you use date or number formatters, you must also make entries in the section that defines the field types and formats used in the input CSV file. In this example, the entries would look like this:

```
field_type_total=6
field_type_3=number,0;0
field_type_4=date,yyyyMMdd
```

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

- Date: http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html
- Number: http://java.sun.com/j2se/1.5.0/docs/api/java/text/DecimalFormat.html

The Converter maps CSV fields to other values using the mapping table entries in the properties file. If a data field contains a literal string as well as data from the CSV file (for example, a formatted telephone number)—and the CSV data is empty—the resulting XML element will also be empty.

An empty XML element causes Destiny to clear the corresponding field.

A missing XML element causes Destiny to ignore the field.

Regular expressions and conditional templates

While the Converter supports regular expressions, this is an advanced feature. A complete discussion of the use of regular expressions is beyond the scope of this document.

In addition, for very complex or nested expressions or those referenced by multiple fields, the Converter provides 5 variable fields: field_variable1 through field_variable5. You can create expressions in these fields, and then other fields can reference them.

In the application, they are at the bottom of the *Destiny Field* column on the *Field Mapping* tab.

EXAMPLE `field_username={field_variable1}`

The expression that assigns a value to the Patron XML field can be a regular expression introduced by the keyword "regex".

The regex function takes two quoted parameters:

- The first parameter is a regular expression.
- The second parameter is the template that applies the matches from the regular expression.

Matching groups are referenced using the $ character, and $0 refers to the contents of the complete match.

NOTE The data is surrounded by quotes in these examples so that spaces in the data can be shown. The actual output data does not contain surrounding quotation marks.

EXAMPLE 1 Parse district ID number

Template:
`field_userDefined1={1,regex("([0-9]{3})([0-9]{6})","{$1}-{$2}")}`
instructs the Converter to change this incoming data to this output data:

CSV data	XML data
"333221234"	"333-221234"

The template parameter has the ability to apply different templates based on the string length of the entire match ($0). This feature addresses issues like telephone numbers that can be empty, contain 7 digits, or contain 10 digits. The Converter scans the template for an entry for the string length of $0.

If found, the Converter replaces any # character (left to right) with a character from the match string.

EXAMPLE 2 Parse phone number, drop area code if area code is not specified:
Template:

`field_addressPrimaryPhoneNumberPrimary={1,regex("\d+","0=;7=###-####;10=(###) ###-####")}`
instructs the Converter to change this incoming data to this output data:

CSV data	XML data
"3448700"	"344-8700"
"8153448700"	"(815) 344-8700"

EXAMPLE 3 Parse phone number, default area code if area code is not specified:
Template:

`field_addressPrimaryPhoneNumberPrimary={1,regex("\d+","0=;7=(815) ###-####;10=(###) ###-####")}`
instructs the Converter to change this incoming data to this output data:

CSV data	XML data
""	""
"3448700"	"(815) 344-8700"
"8153448700"	"(815) 344-8700"

PROPERTIES FILE EXAMPLE

Here is an example of a properties file that uses data mapping.

```
#-------------------------------------------------------------
# PatronImportConverter configuration file (machine generated)
# Do not place comments anywhere else than between the comment
# block markers that follow.
#-------------------------------------------------------------
#COMMENTS_BEGIN
#COMMENTS_END

#-------------------------------------------------------------
# Default short site name
# Uncomment the next line and adjust the value if CSV file # does not contain
site short name information.
#-------------------------------------------------------------
```

```
SiteShortNameDefault=XYZAcademy

#-------------------------------------------------------------
# Matching mode # true=match by District ID
# false=match by barcode/short site name
#-------------------------------------------------------------
MatchUsingDistrictID=true
#-------------------------------------------------------------
# The number of CSV rows to skip before the data begins
#-------------------------------------------------------------
InputRowsToSkip=0

#-------------------------------------------------------------
# Define the rules for when a record should be skipped.
#-------------------------------------------------------------
SkipMode=or
SkipWhen_1=field_districtID,
SkipWhen_2={1},Unassigned
SkipWhen_3={2},Inactive
#-------------------------------------------------------------
# Define the rules for when a record should be deleted.
#-------------------------------------------------------------
UpdateBeforeDelete=false
DeleteMode=and
DeleteWhen_1=field_status,Inactive
#-------------------------------------------------------------
# CSV fields to Destiny field mapping -- Associates XML fields
# that Destiny uses with incoming CSV fields. For special
# processing (concatenated and formatted fields), please
# refer to the configuration guide.
#-------------------------------------------------------------
field_siteShortName={14,omit}
field_districtID={1,omit}
field_lastName={2,omit}
field_firstName={3,omit}
field_patronType={1,map_patronType}
field_accessLevel={1,map_accessLevel}
field_status={1,map_status}
field_gender={5,map_gender}
field_homeroom={15,keep}
field_gradeLevel={6,keep}
field_birthdate={4,date,MM/dd/yyyy}
field_addressPrimaryLine1={8,omit}
field_addressPrimaryCity={9,omit}
field_addressPrimaryState={10,omit}
field_addressPrimaryZipCode={11,omit}
field_addressPrimaryPhoneNumberPrimary=({12,substring(1,3)}) {12,substring(
4,3)}-{12,substring(7,4)}
#-------------------------------------------------------------
# Define the specialized mapping values for the input data
# in the CSV file.
#-------------------------------------------------------------
map_accessLevel_default=Patron
map_gender_default=Unknown
map_gender_1=M
map_gender_2=F|
map_patronType_default=Patron
map_status_default=Active
#-------------------------------------------------------------
# CSV input fields types -- Defines the type and format of # the incoming data
in a given CSV field. Please refer to # the configuration guide for valid values.
#-------------------------------------------------------------
```

```
field_type_total=15
field_type_4=date,yyyyMMdd
```

Running the Converter

Once you have edited the properties file, you can test your configuration by running the Converter to generate the XML file. When satisfied, you can then manually upload the XML data, upload the configuration file into Destiny's Scheduled Update job, or save the configuration file to disk for use with the OS scheduled batch file.

In the application, click **Run**. In the *XML Output File Name* dialog that opens, select the location for, and enter the name of, the output file.

EXAMPLES

```
PatronImportConverter input.csv output.xml patronimport.properties
PatronImportConverter input.csv output.xml patronimport.properties ASCII
```

NOTE If the Converter cannot find a Java Virtual Machine (JVM), a browser window opens, allowing you to download the free JVM from Oracle. The website provides complete instructions. After successfully downloading the JVM, re-enter the command, above.

The Converter writes any exceptions to the log file, named `<filename>-log.txt`. At the end of the log file is a summary count of the number of rows skipped and the number of errors.

The fully qualified path to the XML file and the exception log file appears in the *Confirmation* dialog:

In addition, a browser window opens, displaying the first 10 patron records. This allows you to verify that your output XML file is as expected. If it's not, you'll need to adjust your configuration. After reviewing the preview, close the browser window.

If your output file is satisfactory, close the Converter by choosing **File > Exit** from the menu.

Possible errors

We recommend that you examine the log file, located in the same folder as the XML file, for any conversion errors that were encountered as the file was processed. If you are using the OS scheduled batch files, you can use the ErrorLevel codes in your batch file.

Number of fields

> Row <x> skipped because there are too few fields.
> Expected 41 fields but read 16.

> Row <x> skipped because there are too many fields.
> Expected 39 fields but read 40.

These errors indicate that the properties file definition, field_type_total= ##, does not match the number of fields in the CSV file.

> Row <x> has an invalid index: dataIndex=16 dataFields.length=15

This error indicates that the "Define the rules for…" section contains a value for a Field 16 that does not exist in the CSV file and that the CSV file contains only 15 fields.

Invalid formatting

> Row <x> index is larger than the data field {12,substring(7,4)} for configuration field {12,substring(1,3)}-{12,substring(4,3)}-{12,substring(7,4)}

This error indicates that the "CSV fields to Destiny field mapping" section is not configured correctly for the type of incoming data (in this case, the phone number). It may also be that the incoming data does not contain the correct information for the "CSV fields to Destiny field mapping" section.

> Row <x> has an incorrect number of braces for {7,substring(1,3)-{7,substring(4,3)}-{7,substring(7,4)}

This error indicates that the properties file is missing a matching brace.

In this example, it should be {7,substring(1,3)}- {7,substring(4,3)}-{7,substring(7,4)}

> Row <x> - No starting substring index in {7,substring( )} for configuration field {7,substring( )}-{7,substring(4,3)}-{7,substring(7,4)}

This error indicates that the properties file has an empty start index for the substring.

> Row <x> - Start index must be greater than '0' {8,substring(0,0)} for configuration field {8,substring(0,0)} {8,substring (4,3)}-{8,substring(7,4)}

This error indicates that the properties file has a start index for the substring with a value of "0".

> Row <x> - No CSV data field specified in {l2,substring(1,3)} for configuration field ({l2,substring(1,3)}){12,substring (4,3)}-{12,substring(7,4)}

{l2, substring (1,3)} contains an "L", not a "1". Same error when there's an "O" instead of a "0" (zero).

> Row <x> skipped because of a parsing error: Unterminated quoted string.

This error indicates that the CSV file contains data fields that do not have matching quotes to encapsulate the string. For example, P O Box 2931"

Out of memory errors

> Error writing an XML output file
>
> Error during CSV to XML processing

These errors may indicate the Converter ran out of memory.

By default, the Converter uses the default amount of memory set up for the JVM.

If you encounter a memory error, perform the following steps:

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

Uploading the XML file to Destiny

The last step in the process is to upload the XML file to Destiny. If you are manually converting and uploading a file, follow the "In the Back Office" section below. If you are working with Destiny's Set up scheduled patron update, after uploading the changed properties file, click the **Run** button. If you're using OS scheduling of the batch files, run the batch file.

In the Back Office

1. Log into Destiny as a district patrons manager who has "Manage Patrons for the District" permission.
2. If the Welcome page opens when you log in, open the district view of Destiny by clicking **District** adjacent to the **Logout** button.
3. Open the **Upload Changes** tab of *Update Patrons* in the *Back Office*.
4. If the upload may delete patrons who have current checkouts or fines, select the checkbox to allow the deletion. Otherwise, Destiny retains their patron records.
5. Click **Browse** to locate and select your XML file.
6. Click **Update**, and then **Yes** to confirm.

Through a command line

1. Locate the following files in the FSC-Destiny\fsc\bin folder:
   `updatepatrons.properties.sample`
   `UpdatePatrons.exe`
2. Rename
   `updatepatrons.properties.sample` to `updatepatrons.properties`.
3. Open `updatepatrons.properties` with Notepad.
4. Enter the path to and the name of the XML file.
   Make sure to enter forward slashes, not backslashes.
5. Enter the Destiny username and password for a district patrons manager—one with the permission, "Manage Patrons for the District".
6. Enter your preferences for deleting patrons with current checkouts or fines and for the job summary by entering "true" or "false" as the value for the last two statements.
7. Save and close the file.
8. Open a command prompt. To upload the file, enter the command, the name of the properties file, and the name of the XML file produced by the Converter:
   updatepatrons updatepatrons.properties patrons.xml

If you are a member of a consortium, you'll need to add your Context Name:
`updatepatrons updatepatrons.properties patrons.xml contextname`

Viewing the Update Patrons job summary

After uploading the XML file, Follett recommends that you review the job summary to see if Destiny was successful in updating the patron records.

If you manually uploaded the file in the *Back Office* or clicked the **Run** button on the *Schedule* tab, the Job Manager is already open.

If you uploaded the file through a command line:

1. Log into Destiny with the username and password defined in the updatepatrons.properties file.
2. If the Welcome page opens when you log in, open the district view of Destiny by clicking **District** adjacent to the **Logout** button.
3. Open the **Back Office** tab, and then the Job Manager on the side menu.
4. Monitor the status of the Update Patrons job by clicking **Refresh List**.
5. When the Status is *Completed*, click **View** to open the summary of the upload.

For recommendations on eliminating errors in the patron XML upload process, search for the messagein the Support Knowledgebase. You can access the Knowledgebase in the Customer Portal of www.follettsoftware.com.