Solution home **Documentation** **System Administration**

Non-interactive Patron Import and Update

Modified on: Fri, 21 Oct, 2016 at 12:57 PM

The non-interactive patron import allows your IT Manager to set up a rolling user import and update facility between your school's administration system and Infiniti. This is a non-interactive, "set and forget" facility which, once set up, will run according to the schedule implemented by your IT Manager.

This process allows your school software to 'push' a file of records into Infiniti at scheduled intervals.

*An interactive user import, accessible under **Settings** ⚙ **> Users > Import**, is also available as an alternative method for importing patrons by manually importing a CSV file.*

Before getting started

Infiniti's '**Automatic' Patron Import and Update** solution provides your school with a means of automating the synchronisation of patron information from your Student Management or other appropriate solution. Much like Infiniti's Manual Patron Import, this automated process accepts CSV (Comma Separated Value) files that you, or your IT team, have generated from your internal systems.

This facility utilises a script created and scheduled by the school's suitably qualified IT Technician. The script is not intended for use by library managers or library staff and does not run in a browser. As an alternative, you can choose to use the **manual patron import** **(https://concord.freshdesk.com/solution/articles/1000097528)**.

Prior to commencing the setup of Infiniti's Automatic Patron Import and Update please ensure you are familiar with the following:

1. How to extract information, in CSV format, from your Student Management solution;
2. The use of your preferred scripting tool and language;
3. Basic knowledge of cross-platform non-interactive web agents (we recommend cURL or similar);
4. You school's Infiniti URL, this will be similar to the following: yourschool.concordinfiniti.com;
5. The username and password of an account in your school's Infiniti with system administration privileges.

The CSV File

The first step in automating the synchronisation of your patron information is to create the CSV file which will be used by your script to import / update your users. The non-interactive user import requires ***specific header column names*** in the CSV file in order to recognise user attributes.

The CSV File you upload to Infiniti through this script **MUST** contain the following fields:

username [varchar]: this is the unique identifier field used to identify each user, and the information your users enter as their username when logging into Infiniti
enabled [boolean]: Is the patron an active user of Infiniti. y, yes, *true, 1= True ; n,no,false,0 = False*
staffmember [boolean]: Is the patron a member of staff. y, yes, *true, 1= True ; n,no,false,0 = False*

The CSV File you upload to Infiniti through this script **MAY** contain the following fields (please refer to **manual patron import** **(https://concord.freshdesk.com/solution/articles/1000097528)** for a layperson's description of these fields):

givenname [string]: The patron's Given Name.
surname [string]: The patron's Surname.
middlename [string]: The patron's Middle Name.
preferredname [string]: The patron's Preferred Name (displayed in Circulation)
emailaddress [email]: The patron's primary email address.
graduationyear [integer]: The year (YYYY) the patron will graduate from your school (**mandatory for students**)
librarybarcode [varchar]: The barcode used to identify the patron **(mandatory** for borrowing items from the library.)
adminsystemid [varchar]: The identifier used in your student management solution for this patron (can be identical to username and / or librarybarcode).
gender [varchar]: The patron's gender. *M , F or blank*
password [varchar]: The default password to set for the patron.(optional and overridden by any SSO)
formclass [varchar]: The code that identifies the patron's Form Class (also known as Home Room or Class) in the current academic period.(ie. current year)
presentationgrouplabel [varchar]: The presentation group (by the group's unique label) that the patron may belong to. A patron can only belong to one presentation group at a time.

Below is an sample CSV. A PDF printable sample can be **found here** **(https://drive.google.com/file/d/1EOttWpeGut8tH1qtXCo0wjgemsQ9IgVn997kzzRmGcNlNgpnJRPdQeQWD1C7QhziCS93LqIQVOnq--0C/view?usp=sharing)**.

HTML

```
username,givenname,surname,middlename,preferredname,emailaddress,graduationyear,librarybarcode,adminsystemid,gender,password,formclass,enabled,staffmembe
jdoe1,Jane,Doe,Mary,Jane,jdoe1@yourschool.com,2016,123456,123456,F,changeme,12A,Y,N,style1
mhenry1,Mark,Steven,Henry,Henry,mhenry1@yourschool.com,2018,123457,123457,M,changeme,10C,Y,N,style2
```

The Non-interactive Script

Once your CSV is ready your import script is prepared by a suitably qualified IT technician and will *HTTP POST* a CSV file of users, extracted from your student administration system or similar source of user data, to this URL: ***https://[school.concordinfiniti.com]/api/import/users*.**

Infiniti will treat the first ***multi-part file*** of the *HTTP POST* as the CSV file - the file name and form attribute name of the post are ignored. The CSV file may be empty; for schools performing scheduled incremental imports.

The action is protected by *HTTP Basic Auth* over *SSL/TLS*. You will need to provide the username and password of a new or existing user with *system administration privileges* in Infiniti. Good practice would be to create a new user in Infiniti, e.g., *autoimport*, and grant this user *system administration privileges*. As SAML Single Sign-On requires a browser, the credentials for this user cannot be verified via SAML; one of the other credential sources will be required.

Your school's script can be implemented in any fashion chosen by your IT technicians. An example of how a script may post the CSV file to Infiniti is the **cURL (http://curl.haxx.se/)** application, a cross-platform non-interactive web agent:

HTML

```
curl -X POST -F "file=@users.csv" https://[user]:[password]@[yourschool.concordinfiniti.com]/api/import/users
```

The user CSV file will be processed in the same manner as the interactive import. You should **refer to data requirements and system settings (https://concord.freshdesk.com/solution/articles/1000097528-core-system-users-import-importing-new-users-and-updating-existing-users)** for manual interactive user import before running the script.

Results of Processing

A sample csv import file for existing users in Infiniti could look like:

HTML

```
username,password,emailaddress,formclass
jane,swordfish,jane@gmail.com,10A
john,password1,john@gmail.com,10B
```

When complete the script will produce a JSON result array:
'*success*' - Whether all user records imported without error
'*summary*' - A summary of the import action; successful or otherwise.
'*details*' - An array of each user row outcome. Only included if some rows failed to import. "row number" -> "row outcome"

A JSON result array can look like:

HTML

```
{"success":true, "summary":"100 users processed. 50 users created. 50 users updated. 0 users failed."}
```

or

HTML

```
{"success": false, "summary":"Unauthorised. Check credentials."}
```

or

HTML

```
{
  "success": false,
  "summary": "3 users processed. 1 user created. 1 user updated. 1 user failed.",
  "details": {
    "1": "User Updated",
    "2": "Field (emailAddress) is not a well-formed email address. User failed to import: Field (emailAddress) is not a well-formed email address. ",
    "3": "User Created"
  }
}
```