# Introduction

The NWEA Rostering Kit enables partners to utilize a script based approach for rostering students and teachers, providing a more automated experience than manual rostering and a more flexible experience than Clever. The rostering scripts can be used in conjunction with automation to extract data from a student information system (SIS) into a roster file to fully automate the rostering process keeping your roster of students, instructors and programs accurate and current.

The NWEA Rostering Kit includes Python™ scripts that can be used to initiate various types of roster imports as well as retrieve errors. The rostering kit provides helpful examples so you can customize the scripts to automate the import of your districts rosters. Follow this guide for step-by-step instructions on how to use the sample Python scripts to import roster files into the NWEA Comprehensive Assessment Platform.

**PLEASE NOTE:** You will need familiarity with Python, scripting, script scheduling, and creation of NWEA roster files to utilize this solution. Support from NWEA is limited and does not provide assistance with editing or customizing scripts.

# Process Outline

1. **[Initial Setup](#_Initial_Setup)**
   1. *Downloading and installing Python*
   2. *Installing the Requests library*
   3. *Configuring the rostering.conf file*
2. **[Choosing a Script](#_Choosing_a_Script)**
   1. *Types of uploads*
   2. *Scripts library*
3. **[Verifying and Uploading Your File](#_Verifying_and_Uploading)**
   1. *Checking your roster file*
   2. *Running the script*
4. [**Understanding Feedback**](#_Understanding_Feedback)
   1. *Submitting rosters*
   2. *Cancelling roster import*
   3. *Retrieving import status*
   4. *Retrieving import errors*
   5. *Python errors*

# Contents of the Rostering Kit

* Documentation
  + Setup Guide (this document)
* Configuration file (rostering.conf)
  + This file contains settings used by the scripts
* Import scripts
* Error retrieval script
* Status script
* Cancel script

# Pre-requisites

All rosters imported using the NWEA Rostering Kit method will use configuration settings for match-on-ID by default. Please read carefully the information that follows as it details the functionality, qualification, and risks involved when using match-on-ID with NWEA's assessment system.

The Import Profile module is a key component of NWEA's assessment system on which both MAP Growth and MAP Skills are delivered as it provides a process for bulk creation or modification of student and user profiles. This is crucial for providing access to the most accurate test data possible with the least amount of problem-solving and rework. When importing profiles there are two different configuration options available which the assessment system can use to detect duplicate or potential duplicate profiles.

* Candidate Matching *(Default for manual imports)*

Under this default option, matches for the incoming profile are presented and require the Data Administrator to reconcile the profile either by matching to an existing profile, or by affirmatively selecting to create a new profile. A complex matching algorithm based on a number of criteria (e.g. last name, date of birth, gender, phonetic last name, phonetic first name) is used to identify candidate and duplicate matches.

* Match-on-ID *(Default for script based imports and Clever integrations)*

When this option is configured, the Import Profiles process ignores the candidate matching criteria and looks only to find a matching student ID or instructor ID. If a profile with matching ID is found, the system will update that profile with the incoming data. If no matching profile is found, a new profile will be created.

**Important!**When electing to use the script based rostering you and your organization accept these changes:

* Student and instructor IDs must NOT be recycled within your source student information system. Student and instructor IDs are unique and must remain with the student/instructor throughout their enrollment in the district. This includes when a student leaves, and subsequently returns to the district.
* The accuracy of your data is crucial as the data administrator will NOT have an opportunity to verify and confirm any incoming student information. Any inaccuracy will result in an existing student's profile being overwritten with an incoming student's data and the test history of the existing student being assumed by the incoming student.
* The *Potential Duplicate Profiles*operational report will no longer be available to you. Therefore, it is important that records with duplicate or null student and instructor IDs be resolved prior to NWEA configuring this option for your organization.

Qualifications

* Student IDs are unique and remain with the student throughout the student's enrollment in the district, including when a student leaves and subsequently returns to the district.
* Student IDs are not recycled after a student leaves the district.

Preparing to Configure Match-on-ID **(Action Required)**

If you have previously rostered students and/or instructors in NWEA's assessment system:

* Run the *Potential Duplicate Profiles* operational report and use it to resolve all duplicate profile issues.
* Run the *Profiles with Shared IDs* operational report and use it to resolve all shared ID and missing ID issues.

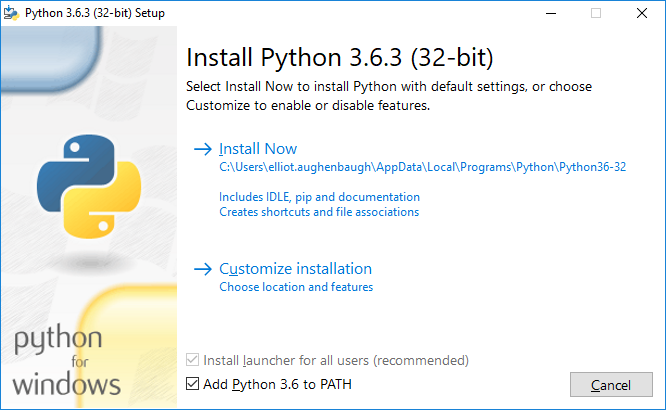
# Step-by-Step

## Initial Setup

***Note:*** *You need to complete Initial Setup only once before your first rostering effort. If you have already completed initial setup, please skip to step 2,* ***Choosing a Script****.*

* 1. *Downloading and installing Python*
     1. Python must be installed on the same computer that you will use to run the scripts and scheduler
     2. Download the latest version of the Python installer appropriate for your operating system from [python.org/downloads](https://www.python.org/downloads/)
     3. Run the installer file
     4. On the first page of the installer window:
        1. For Windows® installs, first check the “Add Python 3.x to PATH” checkbox
           1. **Note:** Forgetting this option will cause “python” to not be recognized as a command at the Command prompt—the entire file path to Python must be executed instead
        2. Next choose the “Customize installation” option

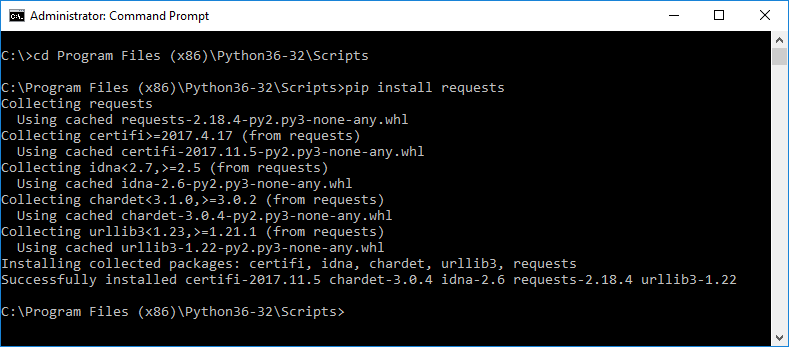
Click the “Add Python 3.x to PATH” checkbox before continuing with “Customize installation”

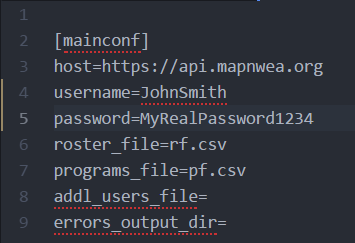


* + 1. On the next page, “Optional Features,” leave the default checkboxes checked:
       1. ☑ Documentation
       2. ☑ pip
       3. ☑ tcl/tk and IDLE
       4. ☑ Python test suite
       5. ☑ py launcher ☑ for all users
    2. On the final page, “Advanced Options,” deselect the first checkbox but do not change the others:
       1. ☑ Install for all users
          1. **Note:** Leaving this option unchecked can install Python in a location with too long of a path for Windows, depending on your file structure
       2. ☑ Associate files with Python (requires the py launcher)
       3. ☑ Create shortcuts for installed applications
       4. ☑ Add Python to environment variables
       5. ☑ Precompile standard library
       6. ☐ Download debugging symbols
       7. ☐ Download debug binaries (requires VS 2015 or later)
    3. Install
  1. *Installing the Requests library*

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Explanation** | **Windows Example** | **Mac/Linux Example** |
| i | Open a Command line interface | Search for “Command Prompt” | Search for “Terminal” |
| Right-click Command Prompt and select “Run as Administrator” | Open Terminal |
| ii | Using the command line, navigate to the folder containing the Python installation you just completed | cd C:\Program Files (x86)\Python36-32 | cd /Applications/Python36-32 |
| dir | ls |
| iii | Navigate into the Scripts folder | cd Scripts | cd Scripts |
| iv | Use Pip to install the Python requests library | pip install requests | sudo pip install requests |

**Note:** Paths provided in step ii are examples only—adjustments may be required depending on where Python was installed



* 1. *Configuring the rostering.conf file*
     1. **Note:** You need to edit a text file, which you can do with the default text editor included on most computers. However, we find the code is easier to read with the Atom editor available from atom.io. Notepad++ is another option for Windows.
     2. Open the Rostering Kit and then, using your text editor, open the rostering.conf file and complete the [mainconf] section of the file. Many of the required fields will be pre-filled. The file names of the roster, programs, and additional users must match the pre-filled names. However, if your files are not in the same folder as the scripts, you should update the pathname.
        1. host=https://api.mapnwea.org
        2. username=[insert your MAP system data administrator user name here]
        3. password=[insert your MAP system data administrator password here]
        4. roster\_file=rf.csv
        5. programs\_file=pf.csv
        6. addl\_users\_file=others.csv
        7. errors\_output\_dir=[optional—if blank, defaults to directory containing roster.conf file. Put a file path here to change the destination directory of any error files created by the process]

## Choosing a Script

1. *Types of uploads*

You have several different options to customize your import. You should understand what type of import you’re trying to do in order to select the correct script.

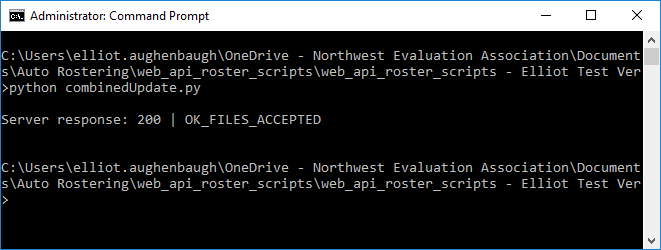
1. User-type uploads
2. Students Only: imports only students
3. Instructors Only: imports only instructors
4. Standard: imports both students and instructors
5. Additional Users: adds users and or updates user roles for the MAP system
6. Roster or program
7. Roster upload: uploads a roster file for rostering students and teachers
8. Program upload: adds or updates program names (English Language Learners, talented and gifted, etc.) and associates them with student IDs
9. Update or replace
10. Update: adds students not already rostered and updates information for students who are rostered
11. Overwrite: overwrites (replaces) all currently rostered students/teachers with your upload
12. *Scripts library*

Once you have determined what type of upload you are trying to complete, choose a script that will complete your upload correctly. Keep reading for explanations of the different scripts available in the rostering kit.

1. Rostering scripts
2. standardOverwrite.py
   * + - 1. Submits a standard roster file (students and instructors) that replaces all students and instructors rostered for the current term
3. standardOverwriteWithPrograms.py
4. Submits a standard roster file (rf.csv) and a programs file (pf.csv) that replace all students, instructors, and programs rostered for the current term
5. standardUpdate.py
6. Submits a standard roster file (students and instructors) that updates any currently rostered students and instructors in the file with updated information and adds students and instructors not already rostered
7. standardUpdateWithPrograms.py
8. Submits a standard roster file (rf.csv) and a programs file (pf.csv) that updates any currently rostered students, instructors, and programs in the file with updated information and adds students, instructors, and programs not already rostered
9. instructorsOnly.py
10. Submits an instructors-only roster file that updates any currently rostered instructors in the file with updated information and adds any instructors not already rostered
11. studentsOnly.py
12. Submits a students-only roster file that updates any currently rostered students in the file with updated information and adds any students not already rostered
13. additionalUsers.py
14. Submits an additional users file (addl\_users.csv) that adds additional users or assigns roles to current users
15. Status/reporting scripts
16. importStatus.py
    1. Retrieves the status of the most recently submitted roster via MAP automated import services
17. importErrors.py
    1. Retrieves any errors found when processing the most recently submitted roster via MAP automated import services, including file-format, group, and individual errors
18. cancelImport.py
    1. Cancels the currently running import, if any

## Verifying and Uploading Your File

1. *Checking your roster file*
   * 1. Use a roster file template to avoid group errors
2. *Verify path, filename, and file type*
   * 1. Make sure that your roster file (rf.csv) is saved in the same folder as the scripts and rostering.conf file or that the conf file contains the correct path to the roster file
     2. Make sure the roster file is saved as a CSV (comma separated values) file
     3. Make sure the roster filename is rf.csv
3. *Running the script*
   * 1. Open the Command Prompt or Terminal
     2. Navigate to the folder containing the scripts, roster file, and rostering.conf file
     3. Type the command to execute your chosen script in Python (without quotes and substituting the name of the script you chose): “python YourSelectedScript.py”
     4. You’ll receive a response; continue to the next section to learn what it means



## Understanding Feedback

When run, the scripts will give feedback to indicate whether the import was successful or not, and if not, why. This section of the guide is dedicated to translating those codes.

1. *Submitting rosters*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| 200 | OK\_FILES\_ACCEPTED | File successfully imported | No fix necessary |
| 400 | ERR\_INVALID\_FORMAT\_TYPE | Import type is not valid | Restore changes made to the rostering.conf file in the [import types] section or to the import script. Import types must be complete, others, students, or instructors. |
| 400 | ERR\_INVALID\_UPDATE\_TYPE | Update\_type is not valid | Restore changes made to the rostering.conf file in the [import types] section or to the import script. Update type must be replace (overwrite) or update. |
| 400 | ERR\_NO\_CURRENT\_TERM | The roster will be imported for the current term, but the current term does not exist in MAP. This typically happens at the beginning of a new academic year. | In MAP, go to Modify Preferences, Manage Terms. Add an academic year and set the term and test window dates. |
| 409 | ERR\_IMPORT\_IN\_PROGRESS | There is currently an import (automated or manual) already in progress | Wait a short while and try again |
| 500 | ERR\_UNABLE\_TO\_CREATE\_IMPORT | An unexpected error occurred: e.g., could not save the csv file to the NWEA file system, database not available, bad configuration | Contact NWEA® Partner Support for troubleshooting |
| 400 | ERR\_UNABLE\_TO\_START\_IMPORT | An unexpected error occurred: e.g., could not save csv file to the NWEA file system, database not available, bad configuration | Contact NWEA Partner Support for troubleshooting |
| 400 | ERR\_INVALID\_FILE\_NAME | Upload file’s filename does not match the required filename | Roster file must be named rf.csv, program file must be named pf.csv, and additional users file must be named others.csv |
| 400 | ERR\_ATTACHMENT | Something unexpected happened when saving the attached roster files | Try again |
| 400 | ERR\_INVALID\_FILE\_FORMAT | Rostering or program file is using an unsupported file type | Resave the rostering or program file into a supported file format (.csv or .zip) |
| 400 | ERR\_CSV\_TYPE\_MISMATCH | The upload file provided does not match the type of import script selected (i.e., instructorsOnly script was run with others.csv) | Provide the expected type of upload file for the script selected |
| 400 | ERR\_NO\_CRF\_PROVIDED | Rostering file not found | Verify that the rostering file is present in the scripts folder |

1. *Cancelling roster import*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| 200 | OK\_CANCELLED | Latest import successfully cancelled | No fix necessary |
| 500 | ERR\_IMPORT\_CANNOT\_BE\_CANCELLED | The import cannot be cancelled if the import status is accepted, the file format validation is queued, or is posting | No fix; import cannot be cancelled |
| 500 | ERR\_CANCELLING\_LAST\_AUTO\_IMPORT | Cancellation failed for an unknown reason | Try again; if the issue continues, contact Partner Support for assistance |

1. *Retrieving import status*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| 500 | ERR\_RETRIEVE\_LAST\_IMPORT | The status of the last import cannot be retrieved | No fix possible; check on the status of your last import in the MAP system |
| 500 | ERR\_LAST\_EVENT\_NOT\_AUTOMATED | The last import was not entered using this process and cannot be reported on using this script | No fix possible; check on the status of your last import in MAP system |

1. *Retrieving import errors*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| 200 | OK\_NO\_ERRORS | No errors occurred during the last import | When no errors are found, an empty errors.csv file will be returned |
| 200 | OK\_IMPORT\_NOT\_COMPLETE | Previous import not yet finished uploading; errors currently unknown | Wait for import to finish before requesting errors again |
| 500 | ERR\_UNABLE\_TO\_BUILD\_ERRORS\_FILE | The error file was unable to be created | No fix possible |
| 200 | OK\_NO\_ERROR\_FILE | Import did not complete successfully, but no errors are available (import may have been cancelled before errors were generated) | No fix possible |
| 500 | ERR\_NO\_AUTOMATED\_IMPORT | The last import was not entered using this process and cannot be reported on using this script | No fix possible |
| 500 | ERR\_RETRIEVE\_LAST\_AUTOMATED\_IMPORT | A database error occurred when retrieving import record | Try again; if the error continues, contact Partner Support for assistance |

1. *Python errors*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| Error: FileNotFoundError: [Errno 2] No such file or directory: ‘[rostering.conf roster file name]’ | A file matching the name or path specified in the rostering.conf file was not found | Verify that your roster\_filename in the rostering.conf file is rf.csv and that there is a roster file named rf.csv in the folder with your scripts |
| TypeError: expected str, bytes or os.PathLike object, but NoneType | Path not found | There may be more than one roster\_file-like entry in the rostering.conf file; verify its contents before trying again |

1. *HTTP errors*

|  |  |  |
| --- | --- | --- |
| **Text** | **Translation** | **Suggested Fix** |
| 401 | Bad Credentials  (this error may include lots of HTML code) | Username and password were not correct | Verify username and password in the rostering.conf file |
| 401 | Empty Username  (this error may include lots of HTML code) | No username provided | Verify that you have provided a username in the rostering.conf file |

## Scheduling the Imports

Once you have configured and tested the roster scripts, you can schedule them to automatically run on a recurring basis. You can use various scheduling solutions (e.g., cron or ActiveBatch®) to automate running the import scripts. \*\***NWEA is not able to aid with these third-party solutions\*\*** We recommend consulting the documentation for your scheduling solution assistance.

# Support

If you have questions or need additional information please contact NWEA Partner Support at (877) 469-3287. Support from NWEA is limited and does not provide assistance with editing or customizing scripts.