



# Using Board Package Tool to Update Adafruit Arduino Packages

Created by Tony DiCola



<https://learn.adafruit.com/using-board-package-tool-to-update-adafruit-arduino-packages>

Last updated on 2023-08-29 03:09:14 PM EDT

# Table of Contents

<a href="#">Overview</a>	3
<a href="#">Dependencies</a>	4
<a href="#">Package Update Workflow</a>	5
• <a href="#">TeeOnArdu Update Example</a>	

---

# Overview

```
package_adafruit_index...
{
  "packages": [
    {
      "maintainer": "Adafruit",
      "help": {
        "online": "https://forums.adafruit.com"
      },
      "websiteURL": "https://adafruit.com",
      "platforms": [
        {
          "category": "Adafruit",
          "name": "Adafruit AVR Boards",
          "url": "https://adafruit.github.io/arduino-board-index/boards/adafruit-1.1.0.tar.bz2",
          "checksum": "SHA-256:4e72d4267d9a8d86874edcd021dc661854a5136c0eed947a6fe10366bc51e373",
          "help": {
            "online": "https://forums.adafruit.com"
          },
          "version": "1.1.0",
          "architecture": "avr",
          "archiveFileName": "adafruit-1.1.0.tar.bz2",
          "boards": [
            {
              "name": "Adafruit Flora"
            },
            {
              "name": "Adafruit Gemma 8MHz"
            },
            {
              "name": "Adafruit Trinket 8MHz"
            },
            {
              "name": "Adafruit Trinket 16MHz"
            },
            {
              "name": "Pro Trinket 5V/16MHz (USB)"
            },
            {
              "name": "Pro Trinket 3V/12MHz (USB)"
            }
          ]
        }
      ]
    }
  ]
}
```

This is a small guide to document a set of scripts that automate publishing Arduino packages to [Adafruit's Arduino board package index \(\)](#). If you just want to use Adafruit's 3rd party boards in Arduino then you don't need to follow this guide--this is written more for people that maintain and update the Adafruit Arduino packages.

The [board package tool scripts \(\)](#) used in this guide are quite flexible and support a workflow for automating Adafruit's board packages. This guide explains their use and shows how they work for Adafruit. If you maintain your own 3rd party Arduino board packages the scripts might be useful to you too, but realize each 3rd party board maintainer has their own conventions and Adafruit's scripts might not work out of the box with your 3rd party boards. Check the --help output of the tool's commands to see descriptions of all the options and usage.

This is a guide we use to maintain our board support package, you're free to fork and adapt the tool but it is completely unsupported and we do not provide any support, advice, or help on use!

---

# Dependencies

This is a guide we use to maintain our board support package, you're free to fork and adapt the tool but it is completely unsupported and we do not provide any support, advice, or help on use!

The code for the board package tool lives in the [Adafruit arduino-board-index repository \(\)](#) and consists of `bpt.py` & `bpt_model.py`. The scripts are written in Python and require Python 3.4 or later.

The pip package manager is included with Python 3.4 or later. You need to install these packages using pip

- [click \(\)](#)
- [gitpython \(\)](#)

You can install these libraries by running the following command:

```
pip3 install --user click gitpython
```

Finally you will need to make sure you install the [git command line tools \(\)](#) (not a graphical interface tool like gitorious, TortoiseGit, etc.--you need the git command line tools for use with the gitpython library). When installing the command line tools make sure to check any installer option to add the git tools to your system path.

After installing all the dependencies you should test that you can run the `bpt.py` script. First [clone the repository with bpt.py \(\)](#) by running a command like:

```
git clone https://github.com/adafruit/arduino-board-index.git
```

Navigate inside the cloned repository directory and run the following command to run the script with the `--help` option (on Windows, you need to say `python` instead of `python3`):

```
python3 bpt.py --help
```

You should see a description like the following:

```
C:\Windows\system32\cmd.exe
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>python bpt.py --help
Usage: bpt.py [OPTIONS] COMMAND [ARGS]...

Adafruit Arduino Board Package Tool (bpt)

Swiss Army knife for managing Arduino board packages. Can check board
packages against a published board package index and notify when newer
versions are available, automatically build updated packages for the
index, and more.

Options:
  -d, --debug           Enable debug output.
  -c, --board-config PATH Specify a INI config file with list of board
                        packages to use. Default is a bpt.ini in the
                        current directory.
  -i, --board-index PATH Specify a board index JSON file. This is the
                        master index that publishes all the packages.
  --help               Show this message and exit.

Commands:
  check_updates  Check package index for out of date packages.
  test_server    Run a local webserver to test board index.
  update_index   Update board package in the published index.

C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>
```

If you see an error go back and make sure all of the dependencies above were installed and try again.

Once the dependencies are installed and you've verified the script runs you're ready to go through the workflow of updating an Arduino package. Continue on to the next page to learn how to update Adafruit's Arduino board packages.

---

## Package Update Workflow

**NOTE:** When updating packages that need file permissions for executables to be preserved (like the Adafruit SAMD package and bossac executables) you **MUST** use either OSX or Linux to update the board package. There is a bug/issue with Windows Python losing file permissions when generating the tar.gz board package file.

To update Adafruit's Arduino board packages first you'll want to be familiar with the [Arduino 3rd party board package index \(\)](#). For Adafruit's boards they are defined and published in the [package\\_adafruit\\_index.json \(\)](#) file of the [arduino-board-index \(\)](#) repository's gh-pages branch. Because this file is in the **gh-pages** branch it will use GitHub's pages publishing system and be available for anyone to access from the following URL:

```
https://adafruit.github.io/arduino-board-index/package_adafruit_index.json
```

This **.json** file contains the 'live' version of all the published Adafruit boards for Arduino's board manager. The board package tool script automates much of the process of publishing a new or updated package to this file.

In addition to the package index .json file there are archive files of all the board packages in the boards directory of the board index repository. This is not a hard requirement of the Arduino 3rd party board system but is a convention followed by Adafruit's board packages. Each board package release version will have a `.tar.bz2` archive inside this boards folder.

The list of boards which are published in the Adafruit Arduino board package index are defined in the `bpt.ini` file next to the `bpt.py` script. This INI style configuration file has a section for each board package, and inside the section are options to configure the source for the board package (i.e. what separate GitHub repository contains the board package) and how the package will be published to the index. For just updating existing board packages you don't need to worry about the contents of this file. Read the comments in the file if you're curious what different options are available and what they mean.

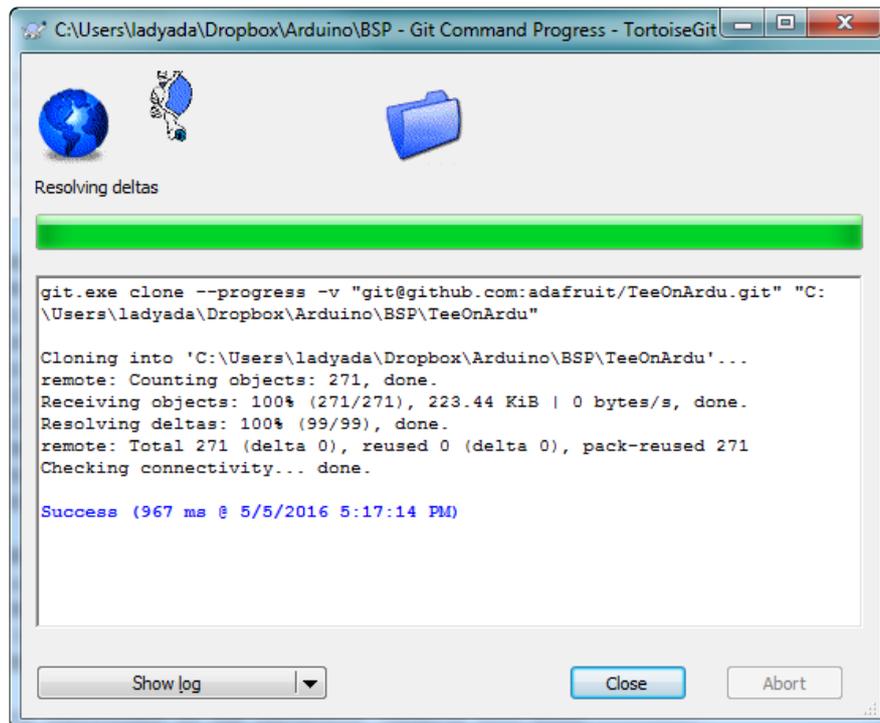
At a high level the workflow for updating an Adafruit board package follows these steps:

1. Update the board package inside its GitHub repository. You can make whatever changes necessary to the package and test them in isolation (like by dropping the package in the Arduino/hardware folder). Don't go on until your package is working exactly as expected.
2. Increase the version number in the package's `platform.txt` file. You'll see a line near the top of this file that looks like `version=1.0.8`. This version inside `platform.txt` is the 'master' version for the package and will be used when pushing to the board index. If the version number isn't higher than what's already in the package index it won't be able to be updated!
3. Once the package is updated in its GitHub repository use the `check-updates` command with the board package tool to verify that the package version in the board index is not up to date and should be updated.
4. Use the `update-index` command with board package tool to update the board index for the specified package. This will pull down the latest package code from its GitHub repository, create an appropriate archive file with the package code inside, and update the `package_adafruit_index.json` file with the new package release metadata.
5. Use the `test-server` command with board package tool to run a local webserver that allows you to test the Arduino IDE with the updated package index before making it 'live'.
6. After verifying the package index and updated package works you can commit the updated files in the Adafruit Arduino board index repository. Once these files are committed to the `gh-pages` branch they will be live for anyone to access!

# TeeOnArdu Update Example

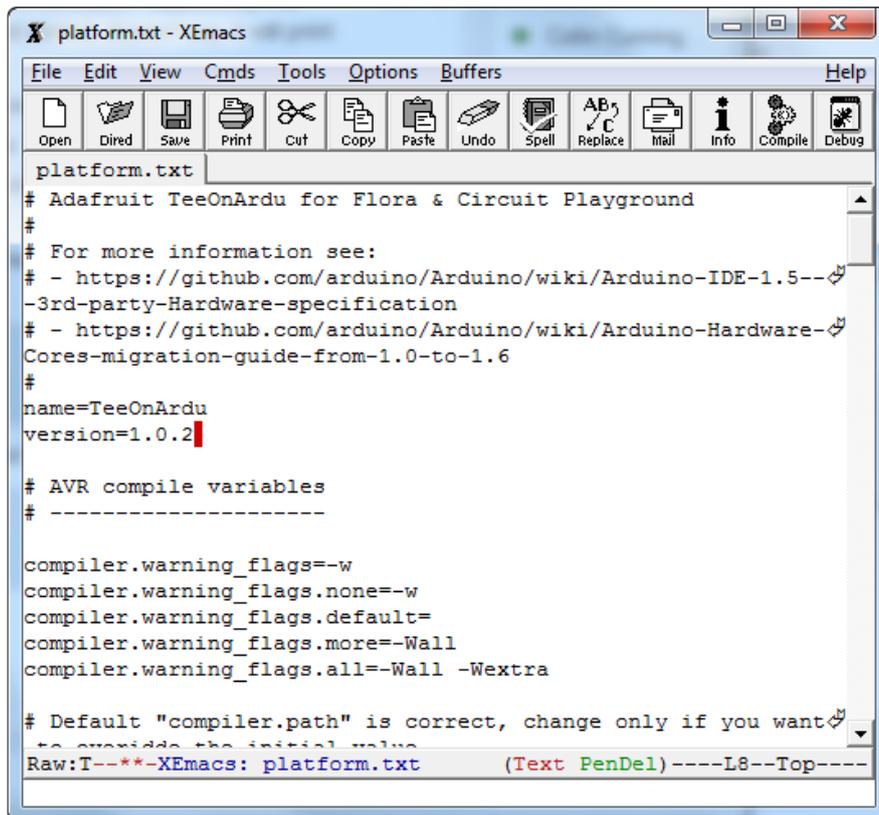
To demonstrate the update workflow we'll walk through updating the [Adafruit TeeOnArdu board package](#) ().

First clone the TeeOnArdu repository and make any necessary changes to the package data.

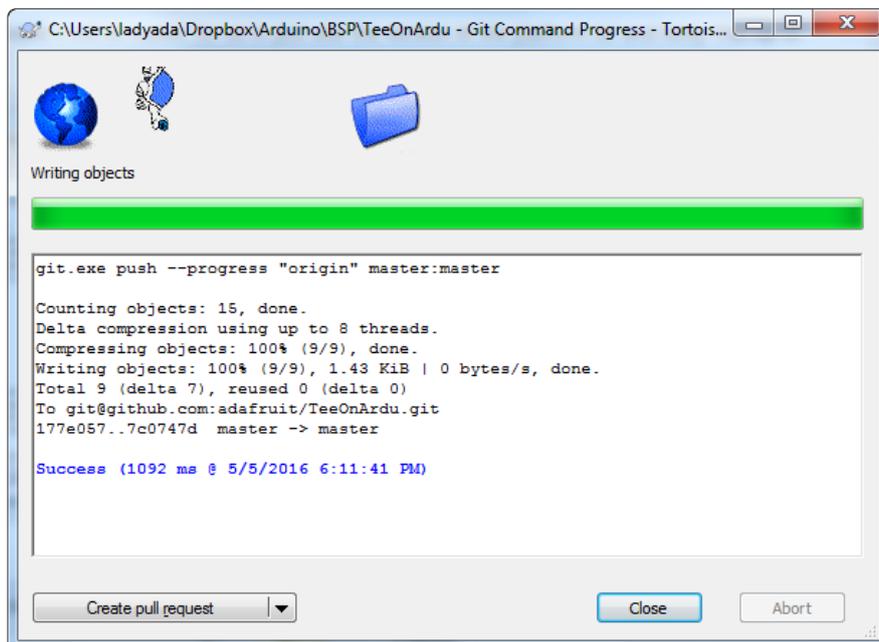


Test out these changes before continuing--you want to make sure the package repository has working code as it will be much more difficult to later debug from the package index.

Once the TeeOnArdu code is updated make sure to increase the **platform.txt** version! This version is the only source of truth for the current version of the package and must be increased for the board package tool to allow an update to the index (otherwise you'd be pushing new code with an old version and Arduino clients wouldn't see it!).



Now commit all the changes to the TeeOnArdu repository. This will update the source of the TeeOnArdu package that the board package tool will later retrieve.

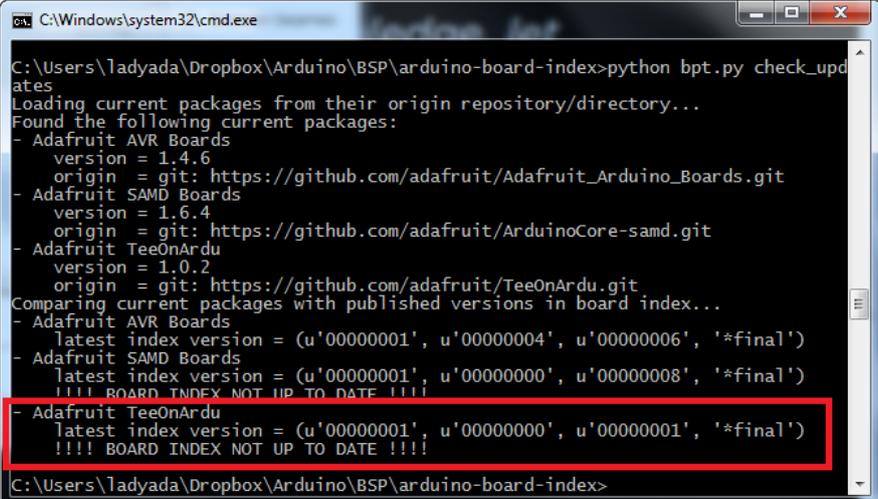


At this point you're ready to switch to the Adafruit [arduino-board-index \(\)](#) repository and use the board package tool scripts. If you haven't cloned this repository already do so now. If you already have the repository run a '**git pull**' command to make sure it's up to date.

Now run the `check-updates` command to have the tool scan the source package repositories and compare their platform.txt versions to the versions published in the board index. The tool will alert you of packages which are not up to date in the index. Run the tool as follows (use `python`, not `python3` on Windows in all the subsequent commands):

```
python3 bpt.py check-updates
```

You should see the TeeOnArdu package is noted as out of date with the `!!!` warning text:



```
C:\Windows\system32\cmd.exe
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>python bpt.py check_updates
Loading current packages from their origin repository/directory...
Found the following current packages:
- Adafruit AVR Boards
  version = 1.4.6
  origin = git: https://github.com/adafruit/Adafruit_Arduino_Boards.git
- Adafruit SAMD Boards
  version = 1.6.4
  origin = git: https://github.com/adafruit/ArduinoCore-samd.git
- Adafruit TeeOnArdu
  version = 1.0.2
  origin = git: https://github.com/adafruit/TeeOnArdu.git
Comparing current packages with published versions in board index...
- Adafruit AVR Boards
  latest index version = (u'00000001', u'00000004', u'00000006', '*final')
- Adafruit SAMD Boards
  latest index version = (u'00000001', u'00000000', u'00000008', '*final')
  !!! BOARD INDEX NOT UP TO DATE !!!
- Adafruit TeeOnArdu
  latest index version = (u'00000001', u'00000000', u'00000001', '*final')
  !!! BOARD INDEX NOT UP TO DATE !!!
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>
```

If you don't see the TeeOnArdu package as more recent in its origin vs. the index then something is wrong and the update to the TeeOnArdu package platform.txt did not work as you expected. Go back to the TeeOnArdu repository and make sure it is updated in its master branch and that the platform.txt version is greater than the version in the index as noted by the `check_update` command output.

Now you're ready to update the board index file with the new TeeOnArdu package.

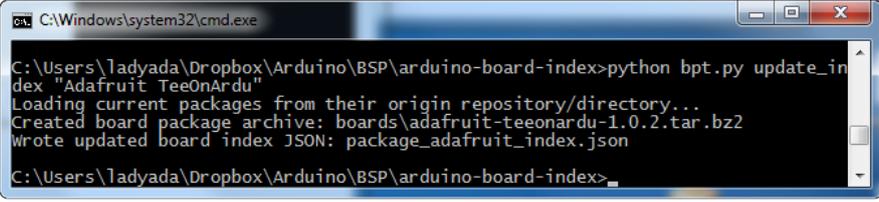
This update will only be done to your local git repository so don't worry about it breaking anything in the live index yet (just be careful not to commit the changed files until you're sure they work!).

To update the TeeOnArdu package run the following command:

```
python3 bpt.py update-index "Adafruit TeeOnArdu"
```

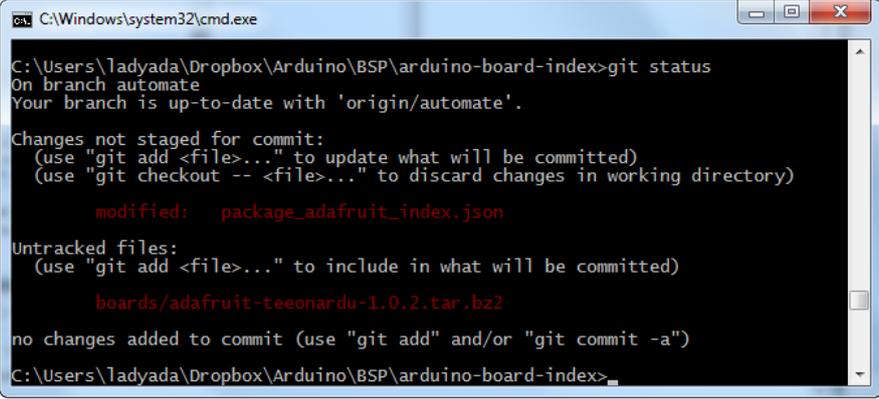
This command takes as a parameter the name of the package to update. This name is the value specified in the section title of the `bpt.ini` file, or you can see it printed in the `check_update` output.

You should see something like the following that confirms the board package archive was created and the index .json updated:



```
C:\Windows\system32\cmd.exe
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>python bpt.py update_index "Adafruit TeeOnArdu"
Loading current packages from their origin repository/directory...
Created board package archive: boards\adafruit-teeonardu-1.0.2.tar.bz2
Wrote updated board index JSON: package_adafruit_index.json
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>
```

If you run the 'git status' command you should see both these files as changed/added:



```
C:\Windows\system32\cmd.exe
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>git status
On branch automate
Your branch is up-to-date with 'origin/automate'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   package_adafruit_index.json

Untracked files:
  (use "git add <file>..." to include in what will be committed)

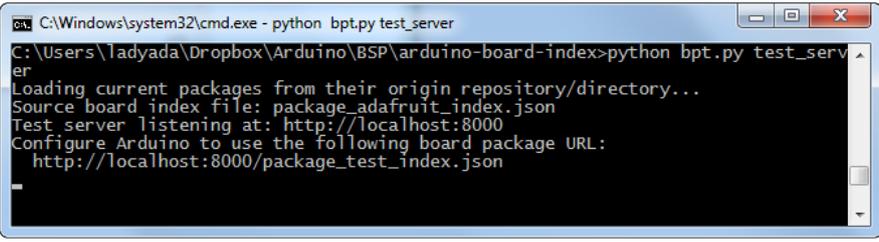
        boards/adafruit-teeonardu-1.0.2.tar.bz2

no changes added to commit (use "git add" and/or "git commit -a")
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>
```

Now you're ready to test out the updated package index using a local test web server. Run the following command to start this test server:

```
python3 bpt.py test-server
```

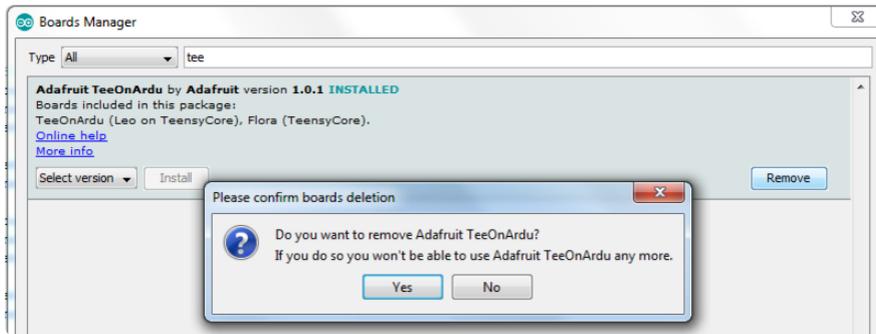
You should see the command start and print that a test server is running (note that you might need to confirm Windows firewall warning dialog first):



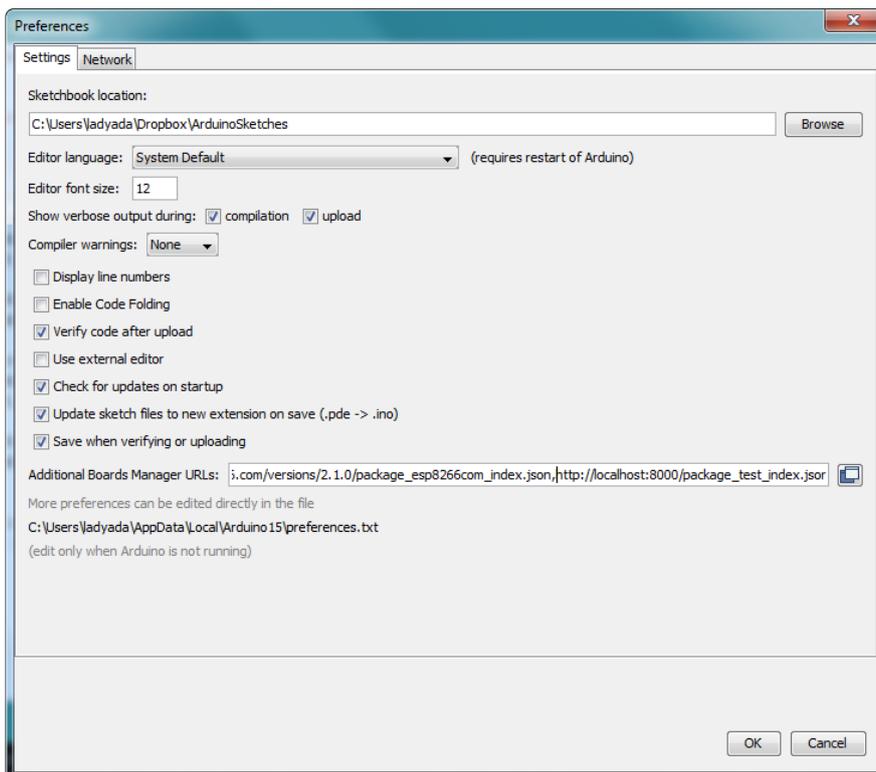
```
C:\Windows\system32\cmd.exe - python bpt.py test_server
C:\Users\ladyada\Dropbox\Arduino\BSP\arduino-board-index>python bpt.py test_server
Loading current packages from their origin repository/directory...
Source board index file: package_adafruit_index.json
Test server listening at: http://localhost:8000
Configure Arduino to use the following board package URL:
http://localhost:8000/package_test_index.json
```

At this point the test webserver is running and will serve the updated board index on the URL it mentions: [http://localhost:8000/package\\_test\\_index.json](http://localhost:8000/package_test_index.json) () If you configure the Arduino IDE to use this board platform URL you can test that the updated package and index are working as expected.

Open the Arduino IDE and in the board manager first remove any old versions of the TeeOnArdu package. This will help ensure you're starting from a clean slate.

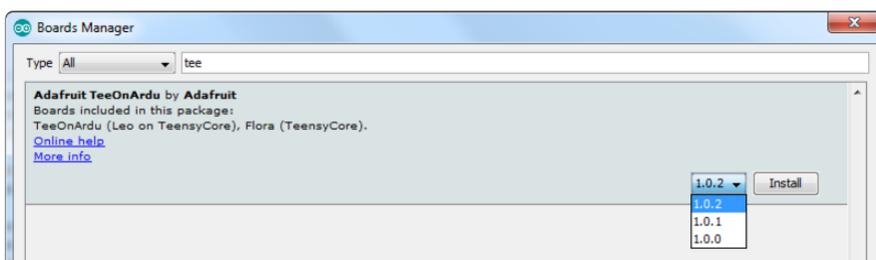


Now update the board package URL in the preference menu. Add the localhost test server URL:



Make sure to remove the live adafruit package index too (i.e. the <http://adafruit.github.io> )... URL!

Now open the board manager and allow it a few moments to update. Then search for the TeeOnArdu package and confirm you see the new version that you added (1.0.2 in this example):



Click Install and make sure the package successfully installs (this will test the archive was created and packaged correctly).

Now use the Arduino IDE with your updated package. Confirm that everything is working as expected and the new changes are available in the package. If you tested the package in isolation earlier there shouldn't be any surprises here, but it can't hurt to test again.

Once you see the updated package is working you can exit the Arduino IDE (and optionally remove the test server URL from the preferences menu and remove the updated TeeOnArdu package).

Press Ctrl-C in the command window running the test server to stop it from running.

The final step to finish the package update is to commit the updated Adafruit arduino-board-index repository files. As you saw earlier a new package .tar.bz2 file was added to the boards directory, and the package\_adafruit\_index.json file was updated. Use the '`git add`' command to add these files (or use whatever Git tool you're most comfortable using). Then use the '`git commit`' command to commit this change. Finally push the new commit to GitHub with the '`git push origin gh-pages`' command.

At this point the TeeOnArdu package update is complete! Once the files are committed to the gh-pages branch the Adafruit Arduino board package index file is updated and available for any Arduino client to see. That's all there is to using the board package tool to update an Adafruit Arduino board package!

As a last sanity check you should add the Adafruit board package URL back to the Arduino IDE and check the updated package is available and installs correctly. If you see any errors make sure both the new package `.tar.bz2` and updated `package_adafruit_index.json` were added to the repository in the previous step. If something is still seriously wrong with the updated package index you can always revert the `package_adafruit_index.json` file on GitHub and put it back to an older known-working version.