# Python Based Method for Automatically Downloading Texas Water Development Board Flood Risk Data

**Andrew Birt, Texas A&M Transportation Institute**

**a-birt@tti.tamu.edu**

## Introduction

The Texas Water Development Board (TWDB) offers flood risk data through its website. The data are hosted on a TWDB Microsoft SharePoint site. At the time of writing this document, the data are hosted at the following URL (the TWDB website can be used to find an alternate URL if they are moved).

<https://twdb.sharepoint.com/:f:/t/Flood_Planning_Submission_Collector/EgZpuuC_ebZAlea1Muu5TbgBGlT3U3NCZQnAQrt7BA1uAg?e=R6Joaq>

The problem with the Microsoft SharePoint site is that the size (e.g., GB) of some downloads (using the inbuilt SharePoint tools) is too big for the SharePoint system to handle. Specifically, if one attempts to download a whole folder of data, for example, all the Pluvial 1 in 100 flood depth data by selecting the folder (fig 1), then the resulting zip file is too large for SharePoint and as a result, the \*.zip file that is downloaded will contain download errors[[1]](#footnote-1). The only ‘simple’ solution to this is to download each raster file (\*.tif) manually which takes some time.

## Solution Using Python Code

The Python code below offers a potential solution to the problem of downloading these files for research purposes. Normally, python scripts that automatically download from URL’s are relatively easy to configure. However, the SharePoint host site presents some special issues – specifically authentication, and a complex encoding of URL’s. Based on our own experiences, we’ve prepared this document and a Python script to help others download the flood data. The Python script should be operable by anyone with a basic level of Python expertise. A user with moderate expertise should be able to modify the code if new versions of the flood data become available, or if the SharePoint host changes.

## Instructions

Open the Python script into your favorite IDE. There are extensive comments within the code that explain how it works, but in simple terms, the code contains:

* A function to set up folders on a hard drive of choice. These folders are arranged by “Flood Type then Flood Frequency”. The user of the python code specifies a location to store downloads.
* Three base URLs that are coded to download each “type” of flood data (Pluvial, Combined, and Fluvial. Note that Coastal is not included because the SharePoint site can download this successfully).
* A list of all tile names (\*tiff raster’s) that cover the data extent (Texas).

The Python code can be found here:

The program is run by specifying a flood type in the main calling function (e.g., Pluvial). The program creates storage for the downloads at the specified save path. It then loops through each flood frequency (e.g., 1in10, 1in20, 1in100…) and each tile name. It uses this information to constructs a specific URL (from the base URL) that points to a specific raster on the SharePoint host and downloads the relevant \*tiff file before continuing.

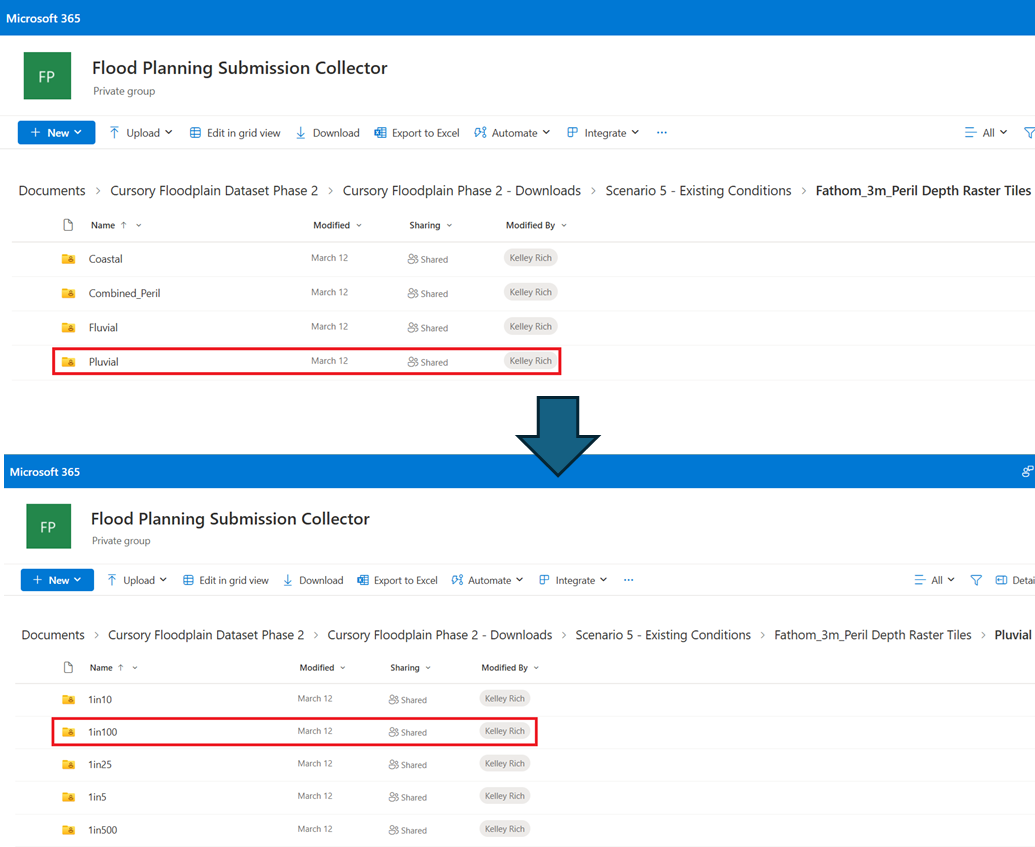
The only difficult part of running the code is that it needs an authentication token to be able to download the data from the SharePoint site. The authentication token can be obtained by visiting the site using a Chrome web browser and allowing it to do any necessary authentication which should happen behind the scenes. The authentication token can then be copied and pasted into the python program so that it makes valid requests. Typically, the token lasts for a day. The following steps show how to retrieve the authentication token using Google Chrome.

1. Navigate to the SharePoint site and double click on “Fathom\_3m\_Peril Depth Raster Tiles”. The link is easy to find by navigating the TWDB website (www.TWDB.org).

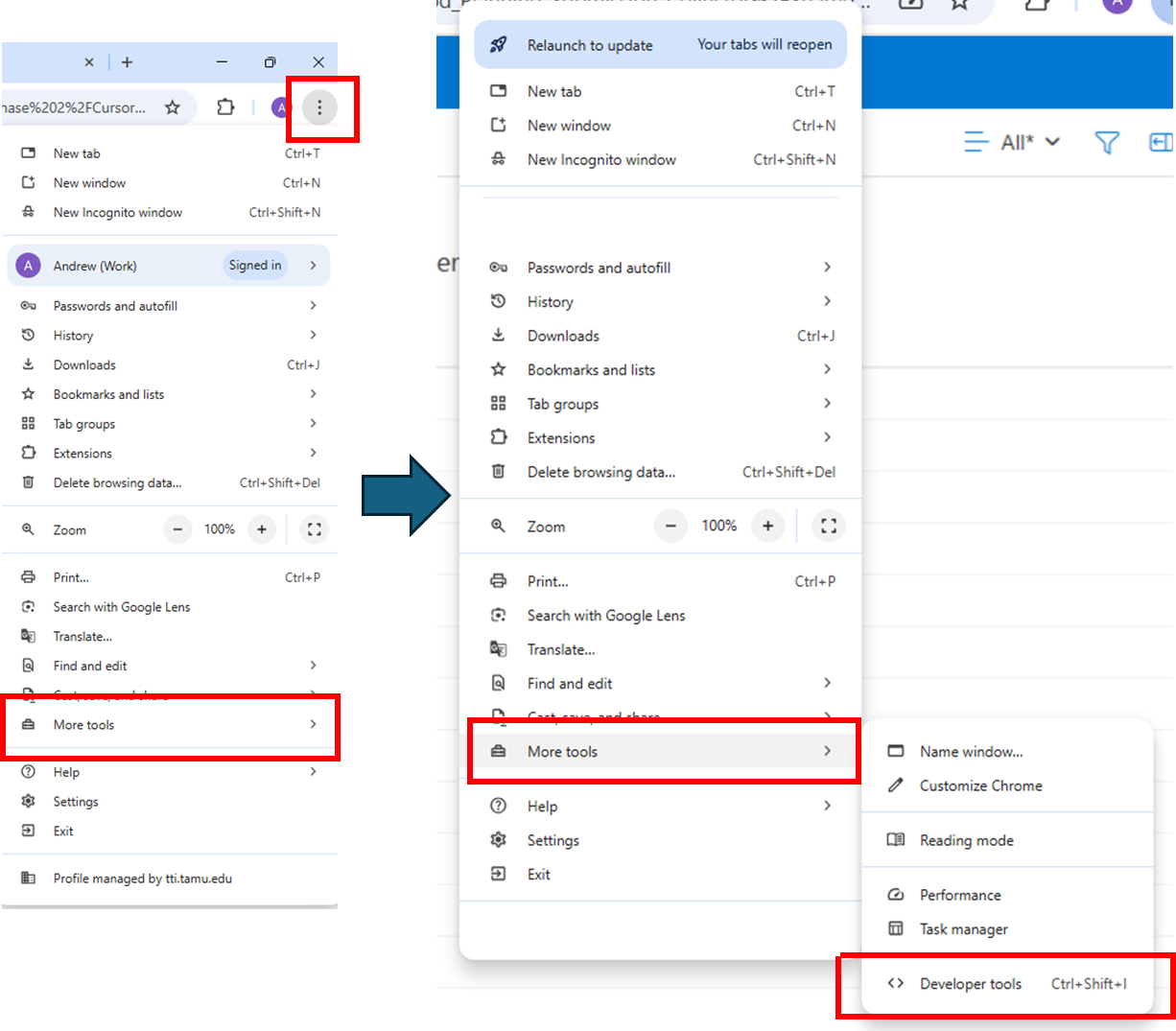
A screenshot of a computer

AI-generated content may be incorrect.

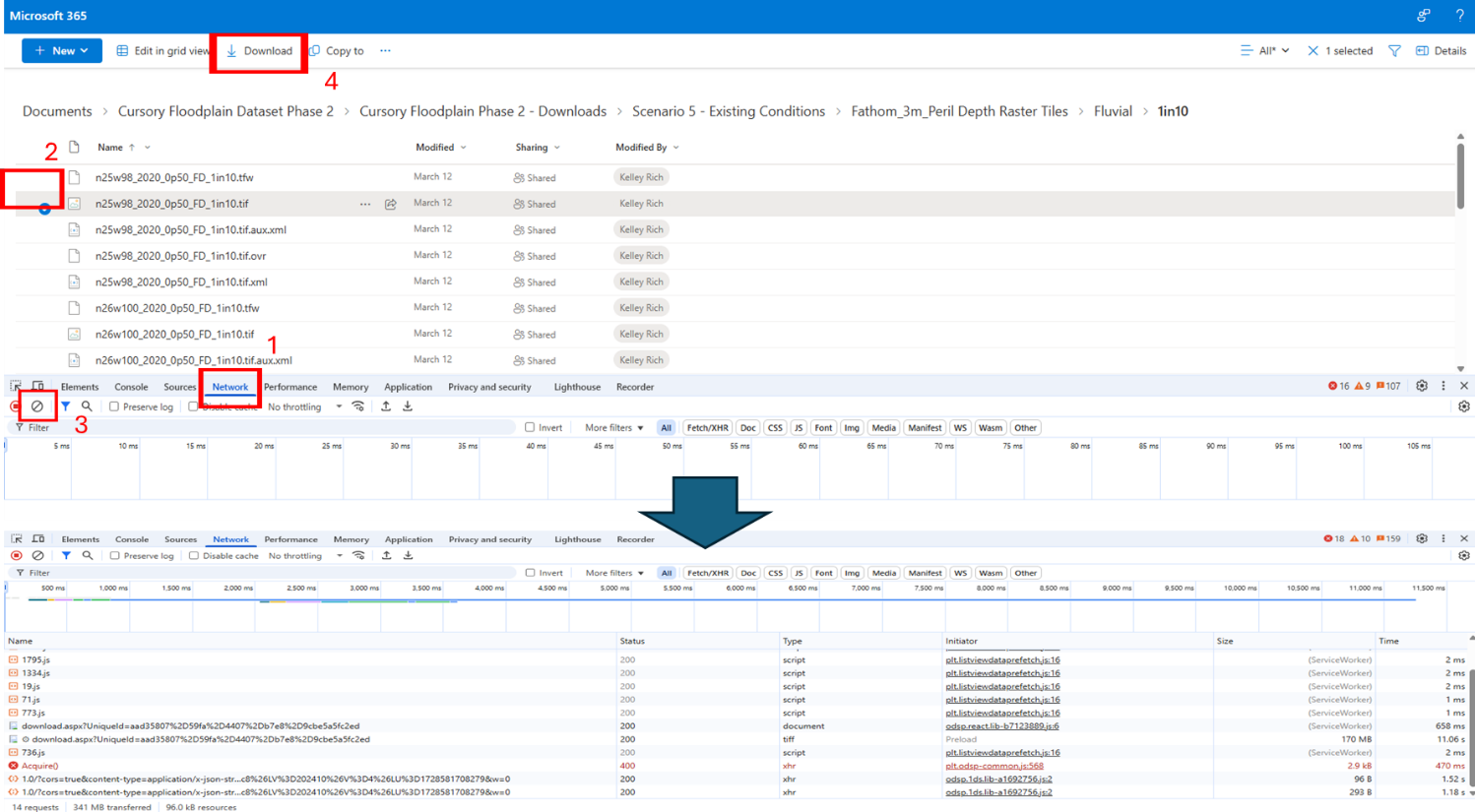
1. Select one of the flood types e.g., “Pluvial” and then any of the flood frequencies e.g., “1in100”.



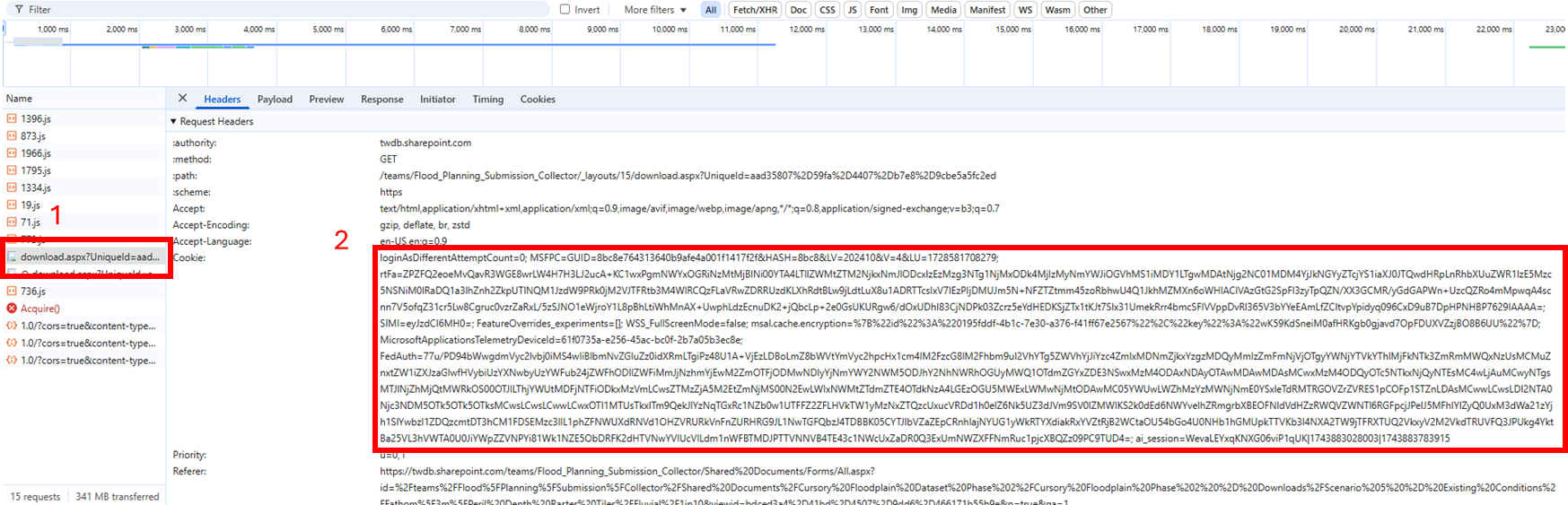
1. Open Chrome Developer Tools. The developer tools window just provides a ‘developer’ view of the associated webpage and any communications it has with its server.



1. The developer tools window may open in a new panel on the right or below the web browser (as shown below). Click on Network in the developer window, then select any link to a \*tiff using the radio box in SharePoint, then clear the list of network requests to make it easy to see the relevant requests. Finally click the download button in the SharePoint site. This should initiate a download of the specified \*.tif to your computer.



1. Clicking the download link will cause the specified \*.tif file to be downloaded to the downloads folder of your computer. More usefully, the network tab of the developer tool will show the requests that have been made to the server to initiate the download. Click on the request to “download.aspx”, then navigate to the headers section of the request, and scroll to the bottom to the “cookie” content. Highlight all the text that makes up the content.



1. Open the Python program and paste the token into the GetCookies function. The entire string can be pasted as is but ensure it is enclosed by quotations. If this is completed successfully, the Python code will use the cookie copied from Google Chrome whenever it makes a request to the SharePoint server, and part of this cookie contains the authentication token (there is no need to isolate the token). With an up to date token in place, any request to the sharepoint site should return a code 200 (success), but if the token is out of date or invalid a code 403 (forbidden) is returned (see comments in code).

A screenshot of a computer

AI-generated content may be incorrect.

1. Note that the Coastal scenarios are unaffected by this problem because they contain a limited number of \*.tif files. [↑](#footnote-ref-1)