

**AN INTERNET – BASED INTERFACE FOR AGGREGATING AND
SUMMARIZING CLIMATE DATA: DESCRIPTION OF THE SOURCE CODE
AND STANDARD OPERATING PROCEDURES FOR INSTALLING AND
MAINTAINING THE WEB SITE**

BY

MIKE TERCEK
WALKING SHADOW ECOLOGY
PO BOX 1085
GARDINER, MT 59030
www.ClimateAnalyzer.org
Tercek@YellowstoneEcology.com

Funding provided through a task agreement with:
The Sonoran Institute
44 E. Broadway Blvd.
Tucson, AZ 857071

January 2013

Table of Contents

Background.....	3
Getting Ready to set up the web site.....	4
The basic skills needed.....	4
Required hardware and software.....	4
Setting up the web site.....	4
Conceptual Background.....	4
Setting up the CGI half of the web site.....	5
Setting up the Zope half of the web site.....	6
Setting up the Googlemaps interface.....	8
Adding a new weather station to the site.....	8
Modifying the web site: Adding or tweaking features.....	9
Literature Cited.....	10
Appendix A: File structure as seen from the Zope Management Interface.....	11

Background

In June 2011, the National Park Service entered into a Cooperative Ecosystem Studies Unit (CESU) task agreement with the Sonoran Institute and the author of this report to write Python code for a web site that aggregates and summarizes climate data. Work began in December 2011. All the work items specified under the task agreement are now complete. The working web site is located at www.ClimateAnalyzer.org. This report documents the source code used to create the site and provides instructions for installing the site on a web server and doing basic maintenance functions, such as adding new weather stations.

Much of the code used in the web site was based on algorithms used in a Windows-based desktop application called “The Climate Data Summarizer,” which can be downloaded for free at this web address: http://www.yellowstoneecology.com/research/COOP_data.html

Under the terms of the cooperative agreement, the source code described here will become public domain, which means that it can be modified and distributed by anyone free of charge. The current version of the web site located at www.ClimateAnalyzer.org was constructed as a demonstration of the code. The web server running www.ClimateAnalyzer.org and its URL are currently owned and maintained by the author of this report at his own expense. New additions or improvements to www.ClimateAnalyzer.org made after the publication of this report may not be funded by the NPS and are therefore the property of Mike Tercek unless specified otherwise under the terms of a new agreement.

This project was completed using 100% free, [open-source](http://www.gnu.org/licenses/old/licenses.html) software and programming tools. This report was also written with a free word processor that is part of an MS Office replacement suite called LibreOffice (<http://www.libreoffice.org/>).

The Python programming language is copyrighted by the Python Software Foundation, however the python language and all the extensions used in this project (e.g. numpy and matplotlib) can be used and distributed for free. The license for the Python programming language is available here: <http://docs.python.org/2/license.html>

Note that the code accompanying this report merely makes function calls to Python and its extensions. The code does not re-distribute python or its libraries, so a copy of the python license is not provided here. In order to use the code provided with this report the user will need to download Python and the appropriate libraries from the original sources as described below.

Getting ready to set up the web site

The basic skills needed to set up and maintain the web site

In order to set up and maintain the web site in its current form, the following skills are required:

1. Understanding of the Linux operating system. This includes command line manipulation of files through the terminal or over a secure shell (SSH) connection and installing web applications like Zope.
2. Familiarity with the Zope / Plone web architecture, including Template Attribute Language (TAL). This architecture is very widely used on the internet and forms the basis for hundreds of popular web sites, including Discover Magazine (<http://discovermagazine.com/>) and NASA Science (<http://nasascience.nasa.gov/>). For more information visit www.zope.org and <http://plone.org/>
3. Competence in writing HTML and javascript.
4. Understanding of how to pass values between web pages as Posts or Query Strings.
5. Ability to use an FTP client to send files to a remote server, modify file permissions, etc.
6. Additionally, in order to make significant modifications to the web site, competence in Python programming and the dependent libraries numpy, scipy and matplotlib are required.

This report is written for an audience that has these skills and therefore does not define most technical terms. The most up to date information source for all the skills just described is Google.

Required hardware and software

You will need:

1. A web server. The current version of the web site is running on an Apache server with CentOS Linux. Any system that runs CGI scripts should work fine.
2. An FTP client. Filezilla is a good choice: <http://filezilla-project.org/>
3. An SSH client like Putty: <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
4. A web browser.
5. A text editor. If you use Windows on your local machine, choose an editor that can save files with new line separators ("\n") instead of carriage returns ("\r"). If you use the wrong separator, scripts that you edit probably will not run on the web server. Do not use notepad, which is the default editor in Windows.
6. Optional: The Climate Data Summarizer. This desktop application is available as a free download at www.Yellowstoneecology.com. You can run the web site without this tool, but it will save you time by automatically making the monthly files described in step 9 below.

Setting up the web site

Conceptual Background

There are two primary components to the web site. The first is the Zope database, which manages the metadata for each weather station, decides which graphs and tables can be applied to which weather station types, keeps track of which years have available data, holds the web page templates (written in HTML, javascript, Python, and TAL), generates the web pages that the users see, and calls the analytical python scripts. The second component is the “CGI half,” which contains the actual climate

data and the analytical python scripts. These two components do not necessarily need to be on the same server because the Python scripts are written as functions [e.g. `def graph1(station_name, year1, year2):`] and the arguments for the functions are passed as query strings in the calling URL. For example the following URL will create a 30-year departure graph for the Mammoth Hot Springs (Yellowstone) weather station. Click below to go straight to the graph:

http://mtercek.webfactional.com/graphers/30dep.py?station=yellowstone_park&yr=2012&title=Yellowstone%20Park%20-%20Mammoth&norm_per=1971_2000

Notice, first, that the server name is `mtercek.webfactional.com` rather than `ClimateAnalyzer.org`. `ClimateAnalyzer.org` contains the Zope database and calls the scripts located on `mtercek.webfactional.com`. In reality, these two addresses are located on the same server, but they could be anywhere. Second, notice that when the `30dep.py` script is called, the weather station name, year of interest, and the departure period are all contained in the URL that follows the script name. If you change 2012 in the URL above to 1956 and paste back into your browser you will get a new graph.

Setting up the CGI half of the web site

1. Set up a web server that supports Python scripting or rent space on one. The current version of the climate web site is hosted by Webfaction (<http://www.webfaction.com/>). They provide reasonable rates and have excellent customer service.
2. Install Python 2.5, 2.6 or 2.7 on the server. Following this, install numpy, scipy and matplotlib, and make sure they are for the version of Python that you just installed. The details for performing this step will differ depending on the setup of your server. In general, it is best to follow the instructions provided by the hosting company. If you use SSH, you should be able to use the `easy_install` command. This command will likely take the general form:
`PYTHONPATH=$HOME/{your root folder name}/Python-2.6/lib/python2.6/site-packages`
`easy_install-2.6 --install-dir=$HOME/{your root folder}/ Python-2.6/lib/python2.6/site-packages {package to install}`. The key is to make sure that you install the dependent libraries for the version of Python that you will be using. **Some servers have more than one running version of Python so this is important.
3. Using your FTP client or SSH, create a directory called “graphers” in your home directory and copy into it all of the Python scripts contained in the “graphers” folder that accompanies this report.
4. Make sure the hash-bang on the first line of the scripts that you just copied refers to the path for the version of Python that you installed (see step 2). The scripts that accompany this report all begin with this line: `#!/usr/bin/env python2.7`. Edit this as needed if you are using a different version of python or need a more specific path.
5. Create a folder called “daily” in your home folder and copy all of your daily (or hourly in the case of RAWS) weather station files into it. Information on how to obtain weather data is available in the instruction manual for the Climate Data Summarizer desktop application at this web address: www.YellowstoneEcology.com. For stream gage data, you will need both the daily data and the historical median files, as described in the download instructions. The historical files should be named with the format `{gagename}_hist.txt` and corresponding daily data would be named `{gagename.txt}`.
6. You can give your climate files any name you like, but don't use spaces or other characters that

will create URL escape problems. These names will be passed in the calling URLs as explained above. In general, use underscores instead of spaces and avoid special characters. The official station names and other metadata will be associated with the file names that you choose in the “all_stations.csv” file described below.

7. Create a directory called “monthly” in home.
8. Open the Climate Data Summarizer and use it to generate one monthly summary file for each of the daily files just mentioned. If you don't want to use the Climate Data Summarizer, create a csv file that has monthly averages in the format shown in the example files in the monthly folder accompanying this report. The monthly and daily files for a single weather station should have exactly the same name. For example, if you have a daily weather file named “Mammoth,” then you must also have a monthly file named “Mammoth.” Copy all the monthly files into the “monthly” directory created in step 7. At the time of writing, the menu options needed in the Climate Data Summarizer were: Create Tables > Create a single file for all months and years in a file.
9. Your python scripts need to know where you stored the climate data. To specify the data path, open the script called “read_daily_data.py” and find the second line, which says:
daily_file_path = '/home/mtercek/webapps/htdocs/daily/.' Edit this line so that it contains the path appropriate for your server. Do the same for the monthly file paths by editing the corresponding line in the script called “graph_init.py” Note that these two Python files are imported by many of the other scripts, so it is best to keep all the scripts together in one directory.
10. Create a file modeled on “all_stations.csv,” which accompanies this report. This file contains the Station name, location, station type, and other basic information for each weather station. Make sure this file is named “all_stations.csv” and copy it into home.
11. Create a file modeled on “1971_2000_averages.csv” which accompanies this report. For more information on gathering 30-year averages (“normals”) for your stations, see the instruction manual for the Climate Data Summarizer. Copy this file into the “monthly” directory on your server. Do the same for another file called “1981_2010” averages.
12. You can test that this half of the site is working by calling URLs of the form shown above in the conceptual overview, making necessary changes for your server name. Once you are sure that the python scripts are finding the data files and are successfully creating graphs and tables, you can move on to setting up the Zope interface.

Setting up the Zope half of the web site

Brief background on Zope Architecture

Zope uses an inheritance structure that is very similar to that used by object-oriented programming languages like Python. The root folder of the Zope hierarchy is analogous to a Python class and subfolders behave like instances of the class, inheriting the root namespace. For example, on www.ClimateAnalyzer.org, the index page template and the cascading style sheet are located in the root of the Zope file structure, and each weather station is represented by an empty subfolder. As a result, calling the URL for a particular weather station's subfolder, creates a web page based on the root page template, but with Header line replaced by the name of the subfolder. Thus, creating an empty subfolder named “Mammoth Hot Springs” in Zope root will create a web page that has all the buttons and navigation tools specified in the page template and a top line header called “Mammoth Hot

Springs.” **This subfolder's name will also be transmitted to the CGI side of the web site when the analytical scripts are called, so the weather station names in Zope must match the file names used to store the data on the CGI side. As a result of the name sharing between the Zope and CGI side of the site, when you create an empty subfolder for a weather station in Zope, you not only automatically create a set of web pages with the relevant graph and table choices, you also automatically set up the structure that the CGI side of the web site needs to find the data files and create tables and graphs. In reality, the web pages for each weather station don't really exist on the hard drive of the server. They are generated when needed. The empty weather station folder in the Zope file structure is just an entry in the database. When a web user calls the URL for a particular weather station, Zope finds the relevant folder, determines which page templates are inherited, and generates an actual web page from the templates on the fly.

Just as with Python classes, subclasses can be created that modify particular aspects of the superclass namespace. In the case of the ClimateAnalyzer.org web site, subclassing is used to control which options are available to different weather station types. For example, a subfolder called “raws” (a weather station type) was created in Zope root, new page templates with the same names as some of the templates in the root folder were copied into this “raws” folder, and weather station subfolders were created within this same raws folder. As a result, all the weather station subfolders within “raws” now call the new page templates rather than those located in root. Since the new page templates in the raws folder have the same names as some of the page templates in zope root, they are replacing the namespace of the superclass (root) with the subclass (raws). Since most of the page templates are the same for all weather station types, subfolders / subclasses need only contain new page templates for those pages that differ for a particular weather station type.

As an example, in the case of RAWS weather stations, only 4 page templates differ from root. “Graph_choices” controls which graphs the user can choose from. “Daily_table2” has a modification that warns the table-creating python script on the CGI side of the site that it is about to read an hourly file instead of daily file. “Cloud” and “Cloud2” create pages that let users choose options related to a graph type that is unique to RAWS stations.

Steps for configuring Zope

1. Install Zope on the server. If easy install does not work, follow the instructions provided by your server administrator. If a bare Zope install is not available, you can instead install Plone, which gives you access to the Zope Management Interface (ZMI).
2. The python script called “get_years.py,” which is located in the “graphers” folder that accompanies this report, determines which years are available (have data) for each weather station. Using the Zope Management Interface (ZMI), make this script into an external python method, following the instructions provided in the Zope documentation.
3. The folder structure that currently exists in the Zope side of the Climate Analyzer web site is shown in Appendix A. Use the buttons and drop down menus in the ZMI to re-create this file structure. Each weather station on your web site will be represented in Zope by an empty folder. ** The names that you use for these subfolders in Zope must match the names that you used for the data files on the CGI side of the web site because these names will be passed in the query string to the python scripts, as described above. Since you will quite likely be setting up your web site for a different geographic region than the current site, replace the names of the regions, e.g. “Greater Yellowstone” and individual weather stations as appropriate. **Notice that in

Zope folder names have two aspects: the ID and the Title. The ID is the system name, which must match the name of the weather station files. The title is the name that will be shown in the web page templates. For example, the ID of the Mammoth weather station folder is “mammoth” and the Title is “Mammoth Hot Springs, Wyoming.” Titles can contain spaces and other special characters while IDs should not.

4. The page templates used on the site have been exported as text files and accompany this report. Using the buttons and drop down menus in the ZMI, paste them into Zope as shown in Appendix A. Be sure to follow the file structure shown.
5. Because of the inheritance structure described above, it is important to copy the correct versions of the page templates into their corresponding subfolders. Be aware that many of these page templates (deliberately) have the same names and are located in different folders. See background on Zope (above) for more detail.
6. To re-iterate, even though you will have new names for your geographic regions and new weather station names, you must have subfolders that correspond to the weather station types: raws, snotel, ghcn, stream_gages. Each weather station must be in the correct subfolder, e.g. all the RAWs stations go in the raws folder. All the GHCN stations go in the ghcn folder, etc. The weather station names (IDs) that you use for the Zope subfolders must match the data file names on the CGI side of the web site.
7. Upload the icons located in the icons folder accompanying this report into the locations shown in Appendix A. These .png files appear in the googlemaps and represent different weather station types. Upload the files in “graph and table choice graphics” folder. These .jpg files appear when users are presented graph or table choices.
8. Once you have created all the necessary folders and copied the page templates into them, you are done.

Setting up the Google Maps Interface for the opening pages

The pages that create the googlemaps are written in javascript using the GoogleMaps API. Complete instructions for writing this type of code is provided here: <https://developers.google.com/maps/>. An example file that creates a map for the Yellowstone area is located in the file “gryn.html” which accompanies this report, in the “example javascript for googlemap” folder.

Adding a new weather station to the web site

1. Add an entry to the file called “all_stations.csv.” Fill in all the required fields.
2. Add a row for your new weather station into the files that contain 30-year averages. These files are 1971_2000_averages.csv and 1981_2010_averages.csv. For more instructions on obtaining 30-year averages, see the instruction manual for the Climate Data Summarizer.
3. Download the daily data for the weather station. For more instructions, see the instruction manual for the Climate Data Summarizer. Upload the daily file to the “daily” directory on the CGI side of the web site.
4. Use the Climate Data Summarizer to create a monthly summary file for your weather station. At the time of writing, the menu options needed in the Climate Data Summarizer were: Create Tables > Create a single file for all months and years in a file. Copy the resulting monthly file to the monthly folder on the CGI side of the web site.
5. Add the weather station to the GoogleMap by editing the javascript in your map page as needed.

6. Deleting a weather station is just a matter of undoing all the above steps.

Modifying the web site: adding or tweaking features

Making changes to the analytical scripts that handle the climate data requires advanced programming skills and is beyond the scope of this report. The algorithms in the analytical scripts have been reviewed by many climatologists and beta testers. If you decide to change them, be cautious. Good reference texts to get you started on the necessary Python skills are: Lutz (2006), Lutz (2008), Alchin (2010), and McKinney (2013). To understand the graphing routines, it is useful to consult programming manuals located on the web sites for scipy (www.scipy.org) and matplotlib (www.matplotlib.org), which distribute the python libraries that were used in this project. Specific programming problems can often be solved by doing a search on stackoverflow (www.stackoverflow.com).

Unless you add new features to the web site, you should not need to edit the Zope web page templates. Changing the page templates on the Zope side of the web site requires knowledge of HTML, javascript, Python and TAL. In practice, the templates are fairly simple compared to those found on many web sites, but even small changes to the code can have large consequences. Consult the Zope documentation (<http://docs.zope.org/>) for details of how these four languages interact when they are used together in one script.

Literature Cited

Alchin, M. 2010. Pro Python: advanced coding techniques and tools. Apress. www.apress.com

Lutz, M. 2006. Programming Python. O'Reilly Publishing. Cambridge.

Lutz, M. 2008. Learning Python. O'Reilly Publishing. Cambridge.

Mckinney, W. 2013. Python for data analysis. O'Reilly Publishing. Cambridge.

Appendix A – File Structure in Zope as seen from the Zope Management Interface

Because of the inheritance and sub-classing explained in the main text, this file structure is the database structure that controls the web site. Note that the “Files” and “dataloggers” folders shown below are not part of the web site under discussion. Remember that page templates with the same names can reside in different folders. When this happens, deeper level files replace the more root-ward files in the namespace, in a manner similar to Python subclasses (see page 6). Use this appendix in conjunction with the page templates that accompany this report to correctly recreate the Zope file structure. Folders designated with weather station names are empty and serve as entries in the Zope database. These entries are used to generate web pages from the templates when they are requested by a web user. A generalized tree follows the screenshots below.

Root Folder

The screenshot displays the Zope Management Interface (ZMI) for a folder named 'Climate'. The interface includes a navigation bar with tabs: Contents, View, Properties, Security, Undo, Ownership, Interfaces, and Find. Below the navigation bar, the folder path is shown as 'Folder at /Climate'. A table lists the contents of the folder, including files and subfolders. The table has three columns: Type Name, Size, and Last Modified. The contents include various files like '30dep', '30dep2', 'agdd_table', 'agdd_table2', 'arbit_monthly', 'arbit_monthly2', 'arbit_monthly3', 'arbit_monthly4', 'climogram', 'coop (coop)', 'daily_table', 'daily_table2', 'dataloggers (dataloggers)', 'front_html', 'get_water_years (get_water_years)', 'get_years (get_years)', 'glkn (glkn)', 'graph_choices', 'graphs', 'greater_yellowstone (Greater Yellowstone)', 'index_html', 'medn (Mediterranean Coast)', 'raws (raws)', 'snotel (snotel)', 'sonoran_desert (Sonoran Desert)', 'sopn (Southern Plains)', 'straight_table', 'straight_table2', 'straight_table3', 'stream (stream)', 'table_choices', and 'tables'. There are also subfolders like 'Files (File Library)', 'greater_yellowstone (Greater Yellowstone)', 'medn (Mediterranean Coast)', 'sonoran_desert (Sonoran Desert)', and 'sopn (Southern Plains)'. The 'Add Plone Site' button is visible in the top right corner. The 'ATContentTypes tool' dropdown menu is also present.

Type Name	Size	Last Modified
30dep	1 Kb	2012-04-26 10:47
30dep2	2 Kb	2012-04-26 10:48
Files (File Library)		2012-01-22 20:59
agdd_table	2 Kb	2012-04-27 18:04
agdd_table2	2 Kb	2012-04-27 18:04
arbit_monthly	1 Kb	2012-04-27 17:18
arbit_monthly2	2 Kb	2012-04-27 17:21
arbit_monthly3	4 Kb	2012-04-27 17:19
arbit_monthly4	5 Kb	2012-04-27 17:23
climogram	1 Kb	2012-04-26 10:52
coop (coop)	1 Kb	2012-01-22 18:05
daily_table	2 Kb	2012-04-27 18:04
daily_table2	2 Kb	2012-04-27 18:05
dataloggers (dataloggers)		2012-09-05 02:53
front_html	1 Kb	2012-04-27 10:11
get_water_years (get_water_years)		2012-10-03 19:21
get_years (get_years)		2012-10-03 19:08
glkn (glkn)		2012-10-17 12:53
graph_choices	1 Kb	2012-10-15 15:57
graphs	40 Kb	2012-03-05 17:57
greater_yellowstone (Greater Yellowstone)		2012-07-25 21:58
index_html	1 Kb	2012-04-27 12:41
medn (Mediterranean Coast)		2012-10-18 19:16
raws (raws)	1 Kb	2012-01-22 18:06
snotel (snotel)	1 Kb	2012-01-22 18:07
sonoran_desert (Sonoran Desert)		2012-04-27 19:41
sopn (Southern Plains)		2012-04-27 19:59
straight_table	2 Kb	2012-04-27 18:05
straight_table2	2 Kb	2012-04-27 18:05
straight_table3	5 Kb	2012-04-27 18:06
stream (stream)	1 Kb	2012-01-22 18:08
table_choices	1 Kb	2012-04-27 18:03
tables	10 Kb	2012-03-05 21:17

Root Folder (continued)

tables	10 Kb	2012-03-05 21:17
thirty_dep_table	2 Kb	2012-04-27 18:06
thirty_dep_table2	2 Kb	2012-04-27 18:06
thirty_dep_table3	3 Kb	2012-04-27 18:07
thirty_dep_table4	6 Kb	2012-04-27 18:07
uthresh_table	2 Kb	2012-04-27 18:07
uthresh_table2	2 Kb	2012-04-27 18:08
uthresh_table3	3 Kb	2012-04-27 18:08
uthresh_table4	4 Kb	2012-04-27 18:09
z_zoo.css	1 Kb	2012-03-05 20:05
z_zoo.pt	1 Kb	2012-10-17 13:00

GLKN Folder – This folder is for the Great Lakes Network I&M Program

glkn - Mozilla Firefox

zinstance.mtercek.webfactional.com/Climate/glkn/manage_main

Most Visited Getting Started BBC Radio 4 Comedy BBC - BBC Radio 4 Extr...

Contents View Properties Security Undo Ownership Interfaces Find

Folder at /Climate/glkn

Add Plone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
	ashland_exp_farm (Ashland Experimental Farm)		2012-10-15 23:49
	brule (Brule Ranger Station)		2012-10-15 23:52
	daily_table2	2 Kb	2012-10-16 00:11
	gurney (Gurney, WI)		2012-10-15 23:51
	index_html	1 Kb	2012-10-16 00:01
	madeline_island (Madeline Island)		2012-10-15 23:48
	map_html	1 Kb	2012-10-17 12:59
	mellen (Mellen, WI)		2012-10-15 23:51
	raws (RAWS)		2012-10-17 11:40
	rawss (raws)	1 Kb	2012-10-17 12:40
	stream_gages (stream_gages)		2012-10-15 23:58
	superior (Superior, WI)		2012-10-15 23:50
	two_harbors (Two Harbors, MN)		2012-10-15 23:55
	uthresh_table4	4 Kb	2012-10-16 00:11

Rename Cut Copy Delete Import/Export Select All

raws folder – With the exception of the weather station entries (the subfolders), this folder is the same whenever it occurs for the various I&M networks (e.g. GLKN, Greater Yellowstone, etc.) so is listed only once here.

Folder at /Climate/glkn/raws

Add Phone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
☐	 apostle_island (Apostle Island)		2012-10-17 11:40
☐	 bailey (Bailey Indiana)		2012-11-27 18:55
☐	 cloud	1 Kb	2012-10-17 11:40
☐	 cloud2	2 Kb	2012-10-17 11:40
☐	 daily_table2	2 Kb	2012-10-17 11:40
☐	 graph_choices	1 Kb	2012-11-27 15:25
☐	 graphs	42 Kb	2012-10-17 11:40
☐	 little_fork_raws (Littlefork Minnesota)		2012-11-27 15:08
☐	 meander (Meander Minnesota)		2012-11-27 15:09
☐	 uthresh_table4	3 Kb	2012-10-17 11:40

Rename Cut Copy Delete Import/Export Select All

medn folder – This folder is for the Mediterranean Coast I&M program.

Mediterranean Coast - Mozilla Firefox

zinstance.mtercek.webfactional.com/Climate/medn/manage_main

Most Visited Getting Started BBC Radio 4 Comedy BBC - BBC Radio 4 Extr...

Contents View Properties Security Undo Ownership Interfaces Find

Folder at /Climate/medn

Add Plone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
☐	burbank_valley_pump_corr (Burbank - Glendale Airport (Corrected Data))		2012-10-18 19:13
☐	chula_vista_corr (Chula Vista (Corrected Data))		2012-10-18 19:14
☐	ghcn (ghcn)		2012-10-18 19:13
☐	index_html	1 Kb	2012-04-27 19:55
☐	map_html	1 Kb	2012-04-14 18:22
☐	raws		2012-04-27 19:57
☐	rawss (raws)	1 Kb	2012-04-14 18:22
☐	san_diego_lindbergh_corr (San Diego Lindbergh Field (Corrected Data))		2012-10-18 19:14
☐	san_diego_sea_world_corr (San Diego Sea World (Corrected Data))		2012-10-18 19:15
☐	santa_monica_pier_corr (Santa Monica Pier (Corrected Data))		2012-10-18 19:15
☐	stream_gages		2012-04-26 10:27
☐	ucla_corr (UCLA (corrected data))		2012-10-18 19:16

Rename Cut Copy Delete Import/Export Select All

ghcn folder – This folder contains subclassed page templates for the GHCN weather station type.

Folder at [/Climate/medn/ghcn](#)

[Add Plone Site](#)

ATContentTypes tool [Add](#)

Type	Name	Size	Last Modified
Folder	burbank_valley_pump (Burbank Glendale Airport)		2012-10-18 18:47
Folder	chula_vista (Chula Vista)		2012-10-18 18:48
File	daily_table2	2 Kb	2012-10-18 18:46
Folder	san_diego_lindbergh (San Diego Lindbergh Field)		2012-10-18 18:49
Folder	san_diego_sea_world (San Diego Sea World)		2012-10-18 18:50
Folder	santa_monica_pier (Santa Monica Pier)		2012-10-18 19:02
Folder	ucla (UCLA)		2012-10-18 18:51
File	uthresh_table4	4 Kb	2012-10-18 18:46

[Rename](#) [Cut](#) [Copy](#) [Delete](#) [Import/Export](#) [Select All](#)

greater yellowstone folder

Folder at [/Climate/greater_yellowstone](#)

Add Plone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
☐	 bicamap_html	1 Kb	2012-03-26 01:28
☐	 canyon (Canyon)		2012-01-25 22:00
☐	 corr (corrected)		2012-08-02 14:23
☐	 ghcn (GHCN)		2012-08-01 13:03
☐	 index_html	1 Kb	2012-04-27 19:53
☐	 map_html	1 Kb	2012-07-23 19:39
☐	 moran (Moran)		2012-01-24 14:53
☐	 old_faithful (Old Faithful)		2012-01-25 20:55
☐	 snake_river (Snake River)		2012-01-24 14:54
☐	 snotel		2012-07-20 11:15
☐	 snotels (snotel)	1 Kb	2012-03-26 01:57
☐	 stream_gages (Stream Gages)		2012-07-20 12:50
☐	 yellowstone_park (Yellowstone Park - Mammoth)		2012-01-24 14:54

Rename Cut Copy Delete Import/Export Select All

/greater_yellowstone/corr This folder refers to weather station files that were corrected and modified by Climatologist Phil Farnes. The subfolders listed here are empty. They are just entries for the weather stations. The Farnes data themselves are stored in COOP format on the CGI side of the site.

Add Plone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
☐	 cooke_city_corr (Cooke City, Montana (Corrected Data))		2012-07-25 21:59
☐	 darwin_ranch_corr (Darwin Ranch (Corrected))		2012-08-02 14:20
☐	 deaver_corr (Deaver (Corrected))		2012-08-02 14:22
☐	 gardiner_corr (Gardiner, Montana (Corrected Data))		2012-07-25 21:59
☐	 jackson_corr (Jackson, Wy (Corrected Data))		2012-07-25 22:00
☐	 lake_yellowstone_corr (Lake Yellowstone (Corrected Data))		2012-08-01 13:16
☐	 lamar_rs_corr (Lamar Ranger Station (Corrected))		2012-08-02 14:21
☐	 lovell_corr (Lovell, WY (Corrected))		2012-08-02 14:23
☐	 old_faithful_corr (Old Faithful (Corrected Data))		2012-07-26 14:18
☐	 tower_corr (Tower Falls (Corrected))		2012-08-02 14:21
☐	 yellowstone_park_corr (Yellowstone Park (Corrected))		2012-08-02 14:19
☐	 yellowtail_dam_corr (Yellowtail Dam (Corrected Data))		2012-08-02 14:23

Rename Cut Copy Delete Import/Export Select All

snotel folder With the exception of the weather station entries (the subfolders), this folder is the same whenever it occurs for the various I&M networks (e.g. GLKN, Greater Yellowstone, etc.) so is listed only once here.

Folder at /Climate/greater_yellowstone/snotel

Add Plone Site

ATContentTypes tool Add

Type	Name	Size	Last Modified
	30dep	1 Kb	2012-04-26 10:31
	30dep2	2 Kb	2012-04-26 10:32
	arbit_monthly	1 Kb	2012-04-26 10:32
	arbit_monthly2	2 Kb	2012-04-27 17:42
	arbit_monthly3	4 Kb	2012-04-27 17:43
	arbit_monthly4	5 Kb	2012-04-27 17:44
	bald_mountain (Bald Mountain)		2012-03-25 16:30
	base_camp (Base Camp)		2012-03-25 16:05
	beartooth_lake (Beartooth Lake)		2012-07-18 18:50
	beaver_creek (Beaver Creek)		2012-07-18 18:51
	black_bear (Black Bear)		2012-07-18 18:52
	blackwater (Blackwater)		2012-07-18 18:52
	bone_springs (Bone Springs Divide)		2012-03-25 16:31
	box_canyon (Box Canyon)		2012-07-18 18:53
	burgess_junction (Burgess Junction)		2012-03-25 16:31
	canyon (Canyon)		2012-03-04 14:09
	carrot_basin (Carrot Basin)		2012-07-18 18:53
	climogram	2 Kb	2012-04-26 10:39
	daily_table2	2 Kb	2012-04-27 18:00
	evening_star (Evening Star)		2012-07-18 18:54
	fisher_creek (Fisher Creek)		2012-03-25 16:05
	grand_targhee (Grand Targhee)		2012-03-25 16:06
	granite_creek (Granite Creek)		2012-07-18 18:54
	graph_choices	1 Kb	2012-04-26 10:36
	graphs	67 Kb	2012-04-01 18:12
	grassy_lake (Grassy Lake)		2012-03-25 16:06
	gros_ventre_summit (Gros Ventre Summit)		2012-07-18 18:55
	gunsight_pass (Gunsight Pass)		2012-07-18 18:56
	island_park (Island Park)		2012-07-18 18:56
	lewis_lake_divide (Lewis Lake Divide)		2012-07-18 18:57
	lick_creek (Lick Creek)		2012-07-18 18:58
	lone_mountain (Lone Mountain)		2012-07-18 18:58
	madison_plateau (Madison Plateau)		2012-07-18 18:59
	monument_peak (Monument Peak)		2012-07-18 18:59
	northeast_entrance (Northeast Entrance)		2012-03-25 16:45

snotel (continued)

	parker_peak (Parker Peak)		2012-03-25 16:04
	phillips_bench (Phillips Bench)		2012-07-18 19:00
	shell_creek (Shell Creek)		2012-03-25 16:29
	shower_falls (Shower Falls)		2012-07-18 19:00
	snake_river_station (Snake River Station)		2012-07-18 19:01
	snotels (snotel) 📄	1 Kb	2012-03-26 01:56
	snow_arbit_yr 📄	2 Kb	2012-10-03 19:23
	snow_arbit_yr2 📄	2 Kb	2012-10-03 19:31
	snow_one_yr 📄	2 Kb	2012-10-03 19:22
	sylvan_lake (Sylvan Lake)		2012-03-25 16:03
	sylvan_road (Sylvan Road)		2012-03-25 16:03
	thumb_divide (Thumb Divide)		2012-03-25 16:04
	togwotee (Togwotee Pass)		2012-03-25 12:30
	two_ocean (Two Ocean Plateau)		2012-03-25 16:07
	uthresh_table 📄	2 Kb	2012-04-26 10:44
	uthresh_table2 📄	2 Kb	2012-04-27 17:50
	uthresh_table3 📄	3 Kb	2012-04-27 17:50
	uthresh_table4 📄	4 Kb	2012-04-27 17:51
	west_yellowstone (West Yellowstone)		2012-07-18 19:01
	whiskey_creek (Whiskey Creek)		2012-07-18 19:03
	white_elephant (White Elephant)		2012-07-18 19:04
	white_mill (White Mill)		2012-03-25 16:45
	wolverine (Wolverine)		2012-07-20 11:15
	younts_peak (Yount's Peak)		2012-07-18 19:05

stream_gages folder With the exception of the weather station entries (the subfolders), this folder is the same whenever it occurs for the various I&M networks (e.g. GLKN, Greater Yellowstone, etc.) so is listed only once here.

Folder at [/Climate/sonoran_desert/stream_gages](#)

ATContentTypes tool

Type	Name	Size	Last Modified
Folder	dry_beaver_ck (Dry Beaver Creek near Rimrock AZ)		2012-03-08 16:26
File	graph_choices	1 Kb	2012-04-26 11:16
File	graphs	63 Kb	2012-03-08 16:27
File	index_html	1 Kb	2012-04-26 11:16
File	multi2	2 Kb	2012-04-27 18:32
File	multi3	2 Kb	2012-04-27 18:33
File	multi4	3 Kb	2012-04-27 18:34
File	multi_stream	2 Kb	2012-04-26 11:18
Folder	rincon_ck_tucson (Rincon Creek Near Tucson AZ)		2012-03-08 16:24
Folder	san_cruz_tubac (Santa Cruz River at Tubac AZ)		2012-03-08 16:23
File	simple_stream	1 Kb	2012-04-26 11:19
File	stream_table	1 Kb	2012-04-27 18:35
File	stream_table2	2 Kb	2012-04-27 18:36
File	table_choices	1 Kb	2012-04-27 18:31
Folder	verde_clarkdale (Verde River near Clarkdale AZ)		2012-03-08 16:24
Folder	wet_beaver_ck (Wet Beaver Creek near Rimrock AZ)		2012-03-08 16:25

Generalized Zope Tree (conceptual):

/root/ (called 'Climate" in the screen shots)

- cascading style sheet (zoo.css)
- primary page template (z_zoo.pt)
- superclass page templates (30dep, 30dep2, etc.)
- /folder for each I&M network (Greater Yellowstone, etc.)
 - /empty folders for COOP stations (ashland exp farm, etc.)
- /raws
 - subclass page templates for RAWs
 - /empty folders for RAWs stations (apostle_island, etc.)
- /ghcn
 - /empty folders for GHCN stations (yellowstone_park, etc)
 - subclass page templates for GHCN stations
- /snotel
 - /empty folders SNOTEL stations
 - subclass page templates for SNOTEL stations
- /stream_gages
 - /empty folders for each stream gage
 - subclass page templates for stream gages

... other station types added with same structure.