**Background**

The Little Sandy River is a tributary of the Ohio River in northeastern Kentucky. The basin’s drainage area is approximately 721 square miles including parts of Greenup, Boyd, Carter, Lawrence, Elliott and Rowan Counties, Kentucky. Terrain is mostly hilly to mountainous draining in a northerly direction, heading north-northeast. The Little Sandy River originates a few miles southwest of Sandy Hook, Kentucky (Elliott County) flowing north-northeast through rugged hills to the town of Grayson, through rolling hills, and then finally connecting with the Ohio River (river mile 336.4) at Greenup, Kentucky. Little Sandy’s two largest tributaries are Little Fork and East Fork each draining approximately 132 and 151 square miles, respectfully.

Grayson Dam is in Carter County, Kentucky on the Little Sandy River at 51.2 river miles above its confluence with the Ohio River. Grayson is an earthfill dam with random rockfill and a central impervious core. Its height is 120 feet and a top length of 1,460 feet. The impoundment area above the dam is 196 square miles. Stream gage locations are Grayson Lake (03216300), Little Sandy River below Grayson Dam (03216350), Little Sandy River at Grayson, KY (03216500) and Ohio River at Greenup Dam Near Greenup, KY (03216600).

**Overview**

Writing effective code with the Jython language requires a basic knowledge of the Python syntax. Writing effective CAVI scripts adds an addition level of complexity having to understand how Java classes can be used within the Jython environment and what those classes are specific to the CWMS CAVI environment. The workshop has X tasks that build upon one another. The initial task “installs” a package to help simplify some of the unknowns first learning to script within the CAVI. RTS-UTILS is essentially a Python package with wrappers around some of the HEC Java classes that access CAVI and active watershed objects and attributes. The following are tasks in this workshop:

* RTS-UTILS installation into the CAVI environment
* Manually running CAVI scripts
* Setup and execute CAVI scripts with automation (Schedule Script Job)
* Setup CAVI program order scripting (Program Order Scripts)

Each task should be completed in the order described above. The first task is critical, and required, to successfully complete all but the last workshop task. Once the first task is complete, most of the Jython code is written and ready to execute except for specific configurations and minor changes in the code. The last task (Program Order Scripts) does not use RTS-UTILS as a dependency, because “currentAlternative” and “computeOptions” Java classes provide the necessary objects.

The CWMS Watershed Little Sandy River Basin has been provided for each workshop task. There are four USGS sites and 11 CWMS RADAR pathnames. These gage locations will be used to configure the data acquisition scripts. Table 1 and Table 2 summarize USGS and CWMS RADAR locations and pathnames.

Table 1. Get USGS Locations

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| USGS ID | SHEF | A | B | F | Parameters |
| 03216350 | GYLK2 |  | Little Sandy blw Grayson | OBS-USGS | Stage, Precip, Temp-Water |
| 03216500 | GYNK2 |  | Little Sandy River at Grayson | OBS-USGS | Stage, Flow |
| 03216600 |  |  | Ohio River at Greenup | OBS-USGS | Stage, Elev |
| 03216300 |  |  | Grayson Lake Near Leon | OBS-USGS | Stage |

Table 2. CWMS RADAR Pathnames

|  |  |
| --- | --- |
| CWMS Pathname | DSS Pathname |
| LRH/Grayson-Outflow.Stage.Inst.15Minutes.0.OBS | //Grayson-Outflow/Stage//15MIN/OBS/ |
| LRH/GraysonKY.Stage.Inst.15Minutes.0.OBS | //GraysonKY/Stage//15MIN/OBS/ |
| LRH/Leon.Stage.Inst.15Minutes.0.OBS | //Leon/Stage//15MIN/OBS/ |
| LRH/Ashland.Stage.Inst.15Minutes.0.OBS | //Ashland/Stage//15MIN/OBS/ |
| LRH/GreenupLD-Pool.Stage.Inst.15Minutes.0.OBS | //GreenupLD-Pool/Stage//15MIN/OBS/ |
| LRH/GreenupLD-Tailwater.Stage.Inst.15Minutes.0.OBS | //GreenupLD-Tailwater/Stage//15MIN/OBS/ |
| LRH/Grayson-Lake.Flow.Inst.15Minutes.0.OBS | //Grayson-Lake/Flow//15MIN/OBS/ |
| LRH/Grayson-Outflow.Flow.Inst.15Minutes.0.OBS | //Grayson-Outflow/Flow//15MIN/OBS/ |
| LRH/GraysonKY.Flow.Inst.15Minutes.0.OBS | //GraysonKY/Flow//15MIN/OBS/ |
| LRH/Leon.Flow.Inst.15Minutes.0.OBS | //Leon/Flow//15MIN/OBS/ |
| LRH/Grayson-Lake.Elev.Inst.15Minutes.0.OBS | //Grayson-Lake/Elev(29)//15Min/OBS/ |

**RTS-UTILS installation**

Instructions for installation can be found at <https://github.com/USACE/rts-utils>. Follow the “How to install” section in the GitHub Wiki page. Installation of the Python package will require a CWMS CAVI restart. Template scripts and template configuration files for some of those scripts are provide with the installation. Some scripts do require modification to their respective configuration file (i.e., cwms\_radar and get\_usgs) but others manager their own (i.e., cumulus). Each of the provided scripts depend on the “status” module. This module contains wrapper methods for the CAVI status and can be used when developing additional user defined scripts.

The data acquisition portion of the workshop will focus on the template scripts “cwms\_radar” and “get\_usgs”. Slight modification to these scripts will be required as you progress through the workshop. These two scripts do require their configuration files updated with the proper attributes for this workshop, which are provided in the tables above.

**Manually Running CAVI Scripts**

*CWMS User’s Manual, section 31.7.1*

This section will focus on modifications to the scripts’ configurations and executing them manually. There are three ways in which to execute scripts manually, **Script Editor**, **Scripts** **Run…**, and via a **Time Series Icon**. First, start with updating scripts and configurations.

**Script Configurations (get\_usgs.py and cwms\_radar.py)**

1. Select **Scripts** | **Editor…** in the CWMS CAVI to open the **Script Editor**
2. Select [**get\_usgs** | **cwms\_radar**]
3. Save a copy selecting **File** | **Save As…**
4. Append “\_manual” to the name
5. Find the corresponding configuration file in the watershed’s “shared” directory *ROOT*\watershed\Lil\_Sand\_Berry\shared; each file name will be prepended with “template\_”
6. Using a text editor, add attributes from the appropriate table above following the existing format in the template configuration file.

If the configuration file is renamed, the script referencing the file will need to have its name changed also. Template scripts already reference the time window of the active module tab (status.get\_timewindow()) and an output DSS file (set\_dssfilename()). If the CAVI time window is changed, the **Refresh** button must be clicked for the change to take effect. Downloaded data will be saved in a DSS file under the “database” directory. Default naming of the DSS file takes the name of the active watershed; get\_usgs script appends *“-usgs-data”* to the filename.

**Script Editor Execution**

1. Select **Scripts** | **Editor…** in the CWMS CAVI to open the **Script Editor**
2. Activate [**get\_usgs\_manual** | **cwms\_radar\_manual**] by either selecting **File** | **Open…**, Ctrl + o, or double-click the script name
3. Click **Save/Run** to execute
4. Using **Tools** | **Console Output…** review log output for any errors, warnings, or information messages
5. Review downloaded data located in *path/to/watershed\_location*/database/*watershed\_name*.dss (cwms\_radar) or *path/to/watershed\_location*/database/*watershed\_name*-usgs-data.dss (get\_usgs)

**Scripts Run… Execution**

1. Select **Scripts** | **Run…** in the CWMS CAVI to open the **Run Script** dialog
2. Select [**get\_usgs\_manual** | **cwms\_radar\_manual**]
3. Select **Run**
4. Using **Tools** | **Console Output…** review log output for any errors, warnings, or information messages
5. Review downloaded data located in *path/to/watershed\_location*/database/*watershed\_name*.dss (cwms\_radar) or *path/to/watershed\_location*/database/*watershed\_name*-usgs-data.dss (get\_usgs)

**Time Series Icon Execution**

*CWMS User’s Manual, section 17.8*

Time series icons displayed in the Map Window represent a site with available information. This information can be in the form of time series data, static images, web pages, webcams, scripts, etc. This section will focus on configuring a CAVI icon with a script for data acquisition using **cwms\_radar** script.

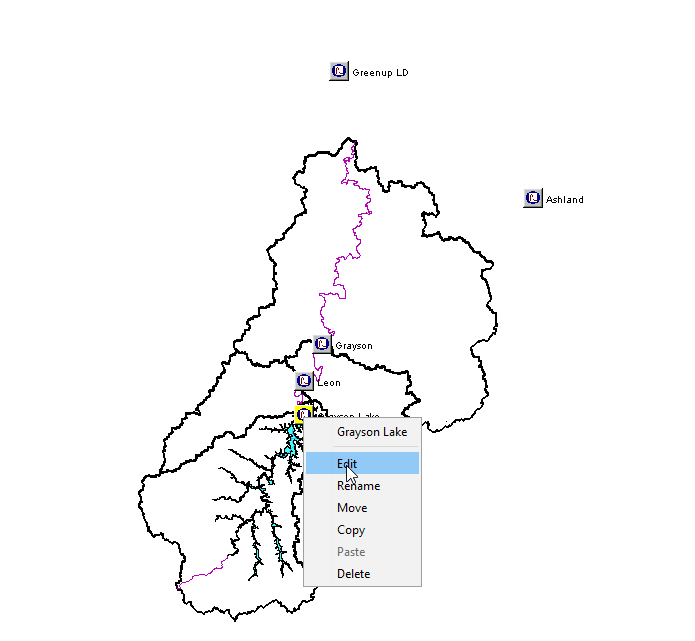
**Script Configuration**

1. Select **Scripts** | **Editor…** in the CWMS CAVI to open the **Script Editor**
2. Select **cwms\_radar**
3. Save a copy selecting **File** | **Save As…**
4. Append “\_icon” to the name.
5. Open the new script in the Editor
6. Comment-out (#) the line starting with **cwmsdat.read\_config…**
7. Change **cwmsdat.set\_tsids()** to **cwmsdat.set\_tsids(var.split(“,”))**
8. Click **Save** and close the **Script Editor**

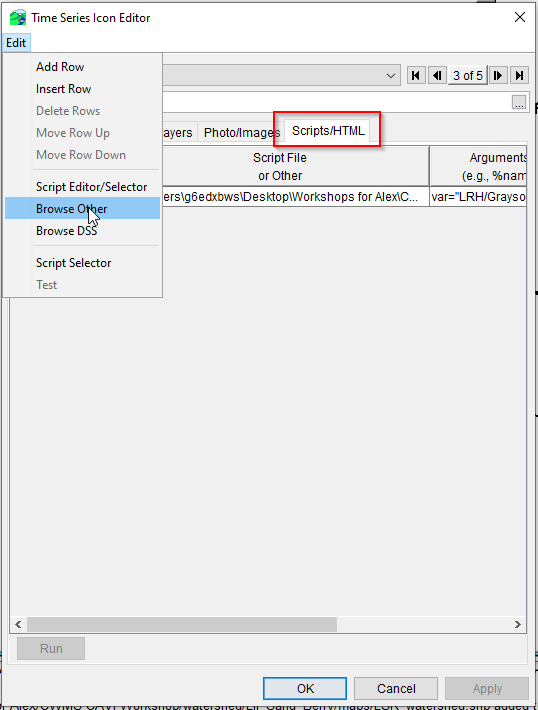
The variable **var** is passed to the script. The value of **var** (String) is split by the delimiter (“,”) setting the RADAR TSID and the resulting DSS pathname.

**Setup Time Series Icon**

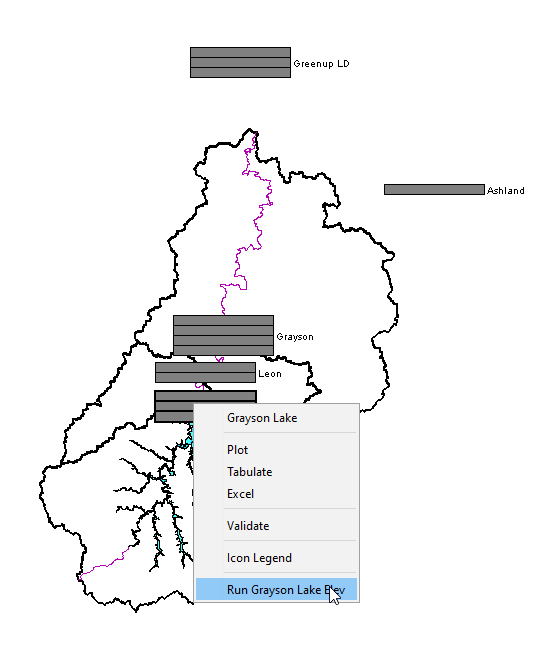
1. Click the CAVI **Setup** tab
2. Right-click **Grayson Lake** time series icon, from the shortcut menu and click **Edit**. This opens the **Time** **Series** **Icon** **Editor**.



1. Select the **Scripts/HTML** tab
2. From the **Edit** menu, click **Browse** **Other**, select the script “*cwms\_radar\_icon.py*” and click **Open**. The selected script’s name will fill the **Display** **Name** column and its path on disk will fill the **Script** **File** **or** **Other** column.



1. (Optional) Change the **Display Name** to a more descriptive name
2. Add an appropriate argument to the **Arguments** column following the format in the example string below
   1. Example String: var="LRH/Grayson-Lake.Flow.Inst.15Minutes.0.OBS, LRH/Grayson-Lake.Elev.Inst.15Minutes.0.OBS"
3. Click **Apply** and **OK**
4. Click File | **Save Watershed** to save changes
5. Select **Acquisition**, **Visualization**, or **Modeling** tab (Modeling requires an open forecast), find the icon that has the script referenced, right-click and select “Run *display\_name*”



1. Using **Tools** | **Console Output…** review log output for any errors, warnings, or information messages
2. Review downloaded data located in *path/to/watershed\_location*/database/*watershed\_name*.dss (cwms\_radar)

**Schedule Script Job**

*CWMS User’s Manual, section 31.7.2*

Running scripts automatically in CWMS can be done by scheduling script jobs. This section describes steps to setup a scheduled job using the **get\_usgs** script. A slight code modification is required to have the time window relative to the script’s execution time.

**Modify Code**

1. Select **Scripts** | **Editor…** in the CWMS CAVI to open the **Script Editor**
2. Select **get\_usgs**
3. Save a copy selecting **File** | **Save As…**
4. Append “\_job” to the name
5. Activate **get\_usgs\_job** by either selecting **File** | **Open…**, Ctrl + o, or double-click the script name
6. Remove or comment-out ***MessageBox.showInformation…*** so a message box is not presented each time the script runs
7. Remove or comment-out the code block assigning start and end time from the time window

tw = status.get\_timewindow()

if tw != None:

st, et = tw

print("Time window: {}".format(tw))

else:

MessageBox.showError("No Forecast open or in 'Setup Tab'")

raise Exception("No Forecast open or in 'Setup Tab'")

1. Replace code in the previous step with the following to make start and end times relative to script execution. Set the number of hours (h) to subtract from the end time (et).

from hec.heclib.util import HecTime

st = HecTime()

et = HecTime()

et.setCurrent()

st.setCurrent()

st.subtractHours(h)

1. Update **retrieve.set\_begin\_date(st)** and **retrieve.set\_end\_date(et)** as follows

retrieve.set\_begin\_date(st.dateAndTime())

retrieve.set\_end\_date(et.dateAndTime())

HecTime()’s **dateAndTime()** method is used to **set\_begin\_date** and **set\_end\_date** argument as a string (<https://github.com/USACE/rts-utils/wiki/CAVI-Scripts#get-usgs>).

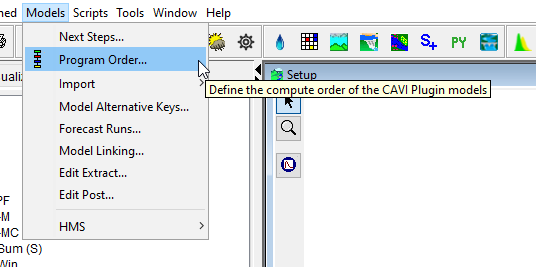
1. Click **Save** and close **Script Editor**
2. Select **Scripts** | **Schedule Script Job** in the CWMS CAVI to open the **Schedule Script Job** dialog
3. From the **Script** drop-down, select the script (Job Name will auto-populate)
4. **Run on Client** should be selected as the **Location**
5. Select either **Single Job** or **Recurring** **Job**
6. Populate **Start** **Date**, **Start** **Time** and/or **Repeat** every *X interval*
7. **Apply**/**OK** to confirm
8. Select **Scripts** | **Script Job Status…** in the CWMS CAVI to open the **Script Job Status** dialog showing job status

**Program Order Scripts**

*CWMS User’s Manual, section 16.3 (HEC Programs in the Program Order) and 16.5 (Scripting in the Program Order)*

**Adding a Program**

1. Select the **Setup** tab
2. Select **Models** | **Program Order…**  in the CWMS CAVI menu module (not the Models tree) to open the Program Order dialog



1. Select the last program in the program order (ResSim)
2. Selecting **Edit** | **Insert** **After** opens the **Select Program** dialog
3. From the **Name** list, select a program, click **OK**
4. Click **Apply**, a **Confirm Save** window will display. This window is warning you that changing a program order after it is in use can affect existing forecast runs.
5. Click **Yes**, the **Confirm Save** window will close. You now have added a scripting to the program order.

**Create a Scripting Program**

1. On the **Setup**  tab and from the **Watershed Tree**, right-click **Scripting**
2. From the shortcut menu, click New. The Create New Scripting Program dialog opens
3. Enter the name of the script, **Post Processing**
4. Enter a description in the **Description** box (optional)
5. Click **OK** and the **Scripting Program** dialog opens
6. In the **Scripting Program** dialog, specify the script in the **Jython Script** box with the fully qualified path and script name, or use the ellipsis to open a browser to locate and select the script
7. Click **Apply**  and **OK**

**Run Forecast**

1. Create a forecast
2. Run forecast through Post Processing

**Excel Power Query**

1. Query and Connection
   1. Data | From Text/CSV
   2. Transform Data Type to datetime
   3. Sort Date Ascending
2. New Blank Query
   1. Right-click under Queries
   2. New Query | Other Sources | Blank Query
   3. Select Applied Steps | Source
   4. Add Table.Profile(Query Name)
   5. Close and Load
3. Charting
   1. Hold down Control key selecting Date and data columns to plot
   2. Insert | Scatter with straight lines
   3. Right-click chart area | Move Chart | New Sheet