

McIDAS-V Tutorial

An Introduction to Scripting

updated June 2009 (software version 1.0beta2)

McIDAS-V is a free, open source, visualization and data analysis software package that is the next generation in SSEC's 35-year history of sophisticated McIDAS software packages. McIDAS-V displays weather satellite (including hyperspectral) and other geophysical data in 2- and 3-dimensions. McIDAS-V can also analyze and manipulate the data with its powerful mathematical functions. McIDAS-V is built on SSEC's VisAD and Unidata's IDV libraries, and contains "Bridge" software that enables McIDAS-X users to run their commands and tasks in the McIDAS-V environment. The functionality of SSEC's HYDRA software package is also being integrated into McIDAS-V for viewing and analyzing hyperspectral satellite data.

McIDAS-V is currently a beta version of the software, which means that the software may contain bugs and not always work as expected. If you encounter any errors, please send them to the McIDAS Help Desk at mug@ssec.wisc.edu. You will be notified at the startup of McIDAS-V when new versions are available on the McIDAS-V webpage - <http://www.ssec.wisc.edu/mcidas/software/v/>. More training materials are available on the McIDAS-V webpage and in the Getting Started chapter of the McIDAS-V User's Guide, which is available from the Help menu within McIDAS-V.

This tutorial assumes that you have McIDAS-V installed on your machine, and that you know how to start McIDAS-V. If you cannot start McIDAS-V on your machine, you should follow the instructions in the document entitled *McIDAS-V Tutorial – Installation and Introduction*.

Introduction

One of the primary uses of scripting is to run processes in the background that can generate display data for use in other environments – usually a web page. Scripting in McIDAS-V is provided by two languages: a comprehensive set of tools in the ISL (IDV Scripting Language) which was developed by Jeff McWhirter at Unidata and uses an XML structure. The second language is Jython – a subset of routines were implemented in Jython to allow a more familiar language for McIDAS-V users; however, it does not allow for the wide range of functions that ISL does.

Jython – Generating Images

1. Load the **Scripting-Data.mcvz** bundle.
 - a. In the “Open Bundle” dialog box select the *Replace session* option and click **OK**
 - b. In the *Field Selector*, select the **GOES-12 VIS sequence** Data Source and click **Create Display** to display all 9 times as an “Image Display”.
 - c. After the display is finished, change the Projection to South America using the **Projections** menu.
 - d. Re-center the image display by zooming (mouse wheel) and roaming (CTRL+right button drag)
2. When you get a pleasing display, save it as a zipped-bundle (*.mcvz).
 - a. Select **File -> Save As...**

- b. Locate a directory on YOUR disk (for example: /home/aos/)
 - c. For the “File Name”, type: **myloop.mcvz**
 - d. For the “Files of Type” select: **Zipped McIDAS-V Bundles (*.mcvz)**
 - e. Be sure that all boxes are checked under “What should be saved”
 - i. Do NOT check the “Save with relative paths” box.
 - ii. Click **Save**.
 - f. In the “Save Data” dialog box, select the “GOES-12 VIS sequence to save the 9 images and click **OK**.
3. Remove all displays and data by selecting **Edit -> Remove -> All Layers and Data Sources**.
 4. Load up the **myloop.mcvz** bundle to check it saved properly.
 - i. Select **File -> Open File** and then locate your **myloop.mcvz** bundle file and click **Open**.
 - ii. In the “Open Bundle” pop-up options, select *Merge with active tab(s)*.
 - iii. Click **OK**
 - iv. Make sure it runs as you saved it.
 5. Shut down you McIDAS-V session completely!
 6. Open up a text editor (Desktop: Applications -> Accessories -> Text Editor will open gedit)
 - a. Create a text file that reads (all lines **must** start in the first column!):


```
loadBundle("/home/aos/mcv/myloop.mcvz")
pause()
writeImage("/home/aos/mcv/myloop.jpg")
writeMovie("/home/aos/mcv/myloop.gif")
print "Done. . . ."
```
 - b. Save this file as **/home/aos/mcv/myloop.py**
 7. Open a terminal window.
 - i. **cd** to the directory where McV is installed (/home/aos/McIDAS-V)
 - ii. Type: **runMcV -isfile /home/aos/mcv/myloop.py**
 - iii. You can check the status of this by looking at the mcidasv.log file in \$HOME/.mcidasv
 8. When the script is done running, look at the files in the /home/aos/mcv directory and view the myloop.jpg image by clicking on it in the File Manager.

9. The animated GIF must be viewed through the browser, so open **Firefox** and use **File->Open** and then navigate to **/home/aos/mcv/myloop.gif**.

Just a note: we used the **.gif** extension so specify an animated GIF since that was what we had a player for (the browser). You may use the extension to specify the format (for example: **.avi**, **.mov**) for the movies, as well as the single-frame images (for example: **.gif**, **.png**).

A little more Jython

1. Re-edit your **/home/aos/mcv/myloop.py** file

- a. Comment-out the “writeMovie” call by putting a # in the first column:

```
#writeMovie("/home/aos/mcv/myloop.gif")
```

- b. Change the “writeImage” call to specify a specific size of the saved image, and a color bar and a title.

(Note: Jython syntax allows you to put a “\” character at the end of a line text that is to be continued on the next line; you may do this, or just use one, very long line). So, either do this:

```
writelnimage("/home/aos/mcv/myloop.jpg", \
"resize height=200 width=200; \
colorbar width=150 height=10 anchor=UM place=UM, 0, 2; \
overlay text=My Image! place=LM, 0, -20 anchor=LM \ color=orange")
```

or this:

```
writelnimage("/home/aos/mcv/myloop.jpg", "resize height=200 width=200; colorbar
width=150 height=10 anchor=UM place=UM, 0, 2; overlay text=My Image! place=LM, 0, -20
anchor=LM color=orange")
```

- c. **Save** the file (you may use the same or a different name).
- d. Back in the McIDAS-V installation directory (**/home/aos/McIDAS-V**), run the **runMcV** program exactly how you did before (Step 7 in the previous section).
- e. When it is finished, check out your new 200x200 image.

ISL – The “other” scripting language

1. Edit a new file – call it **myloop.isl**

```
<i sl >  
<bundle file="/home/aos/mcv/myloop.mcvz" wait="true" />  
<image file="/home/aos/mcv/myloop-isl.jpg">  
  <resize height="300" width="200" />  
</image>  
<movie file="/home/aos/mcv/myloop-isl.gif" />  
</i sl >
```

2. Save this file to the **/home/aos/mcv** directory.
3. Back in the McIDAS-V install directory, run McIDAS-V program as before, but instead of “myloop.py” use **“myloop.isl”**

For running these “background” jobs, the choice of language (Jython or ISL) depends on whether you need the advanced capabilities of ISL. It provides many more features than Jython....and both with continue to evolve!

For more information on ISL and Jython in McIDAS-V, visit the McIDAS-V scripting page at:

http://www.ssec.wisc.edu/mcidas/doc/mcv_guide/current/index.php?page=isl/index.html