

Autoplot and Java Libraries for HAPI and Science Processing

Jeremy Faden
Jon Vandegriff
Bob Weigel

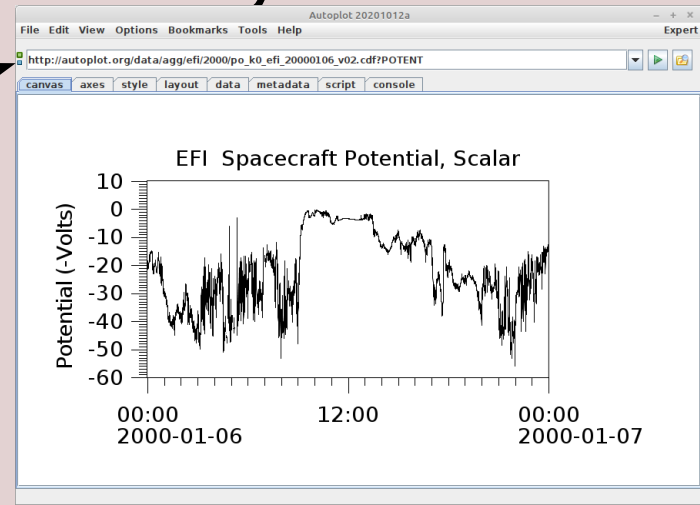
Introduction-Autoplot

- Autoplot is software for looking at Heliophysics data
- Data from various sources are read into a standard data representation and displayed as spectrograms, line plots, and other forms
- Uses the interactive Das2 Java libraries for displaying data.

CDF Files

- CDF files are popular, and Autoplot reads data from these binary files and metadata like axis labels and units.
- Autoplot uses dataset “URIs” to identify data:

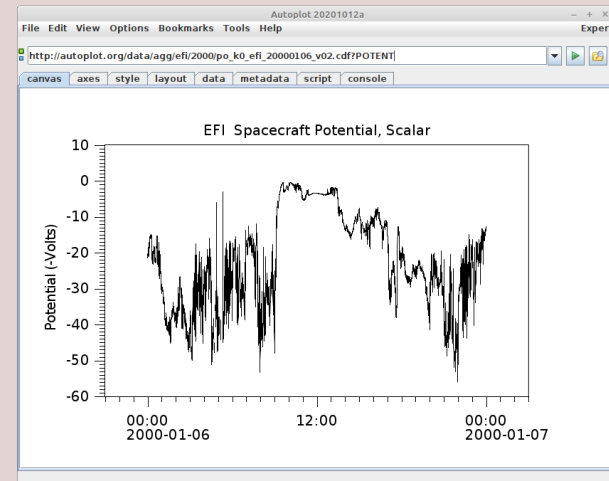
http://autoplot.org/data/agg/efi/2000/po_k0_efi_20000106_v02.cdf?POTENT



Autoplot Aggregation

- It can also combine data using its aggregation feature, where a template is used to identify the data files.
- Here is an Autoplot URI using aggregation to combine data from multiple files:

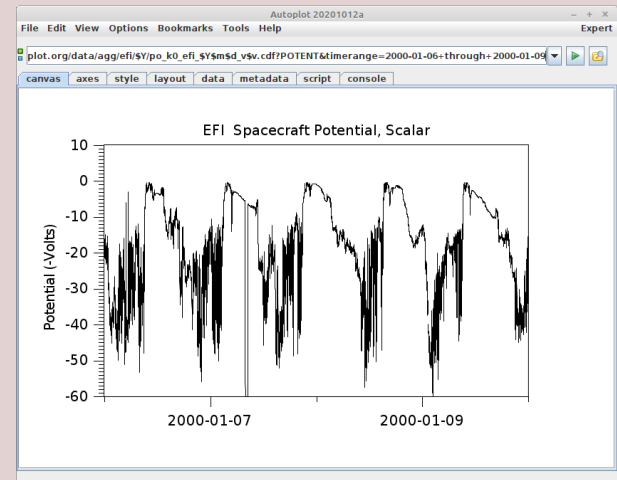
http://autoplot.org/data/agg/efi/2000/po_k0_efi_2000-01-06_v02.cdf?POTENT



Autoplot Aggregation

- It can also combine data using its aggregation feature, where a template is used to identify the data files.
- Here is an Autoplot URI using aggregation to combine data from multiple files:

[http://autoplot.org/data/agg/efi/\\$Y/
po_k0_efi_\\$Y\\$m\\$d_v\\$v.cdf?POTENT
&timerange=2000-01-06/2000-01-10](http://autoplot.org/data/agg/efi/$Y/po_k0_efi_$Ymd_v$v.cdf?POTENT&timerange=2000-01-06/2000-01-10)



Autoplot Aggregation

Autoplot's aggregation will list the remote directory, and interpret each file name to see if it falls within the aggregation time range.

Each file is then downloaded and read in, and the data is combined to make a long time series.

Aggregation and URI_Templates

- CDAWeb, Jon Vandegriff, and I were all using these template strings. Around 2015, we all agreed on a standard specification for these, which we called URI_Templates.
- Autoplot's aggregation handling was extended to support these templates

Aggregation and URI_Templates

- `Ym$d` for 4-digit year, 2-digit month and 2-digit day.
- Corner cases, like
 - `$(subsec;places=3)` and
 - `$(enum;values=A,B)` and
 - `$(periodic;offset=2285;start=2000-346;period=27)`are supported.
- <https://github.com/hapi-server/uri-templates/wiki/Specification>

Aggregation and URI_Templates

- Until recently, there has not been a reference implementation of the URI_Templates spec.
- This has been extracted from Autoplot's Java code base to make a stand-alone library.

URI Templates

- 75 KByte jar file at <https://github.com/hapi-server/uri-templates/blob/master/UriTemplatesJava/store/UriTemplatesJava.jar>
- <https://github.com/hapi-server/uri-templates/tree/master/UriTemplatesJava>
- handles "Formatting":
 `Ym$d.dat,2000-08-01` → `20000801.dat`.
- handles "Parsing":
 `Ym$d.dat,20000801.dat` → `2000-08-01`
- Handles "format range":
 `Ym$d.dat,2000-08` → `20000801.dat, 20000802.dat, ..., 20000831.dat`
- Times are handled as 7-element int array of Y,m,d,H,M,S,N

URI Templates-TimeUtil

- TimeUtil class provides many necessary and useful time handling codes:
 - day of year,
 - next day,
 - format and parse of ISO8601 time strings,
 - toMillisecondsSince1970 (toCdfTT2000 coming soon)
 - Unit tests have 100% coverage and are useful for reference documentation:
<https://github.com/hapi-server/uri-templates/blob/master/UriTemplatesJava/test/org/hapiserver/TimeUtilTest.java>
 - JavaDoc at https://jfaden.net/~jbf/hapi/uri_templates/doc

URI Templates-command line

- There is also a command line interface, so you can experiment with this:

```
wget -N https://github.com/hapi-server/uri-templates/raw/master/UriTemplatesJava/store/UriTemplatesJava.jar
```

```
java -jar UriTemplatesJava.jar --formatRange --range='1999-01-01/1999-01-03'  
--template='http://example.com/data_$(d;pad=none).dat'
```

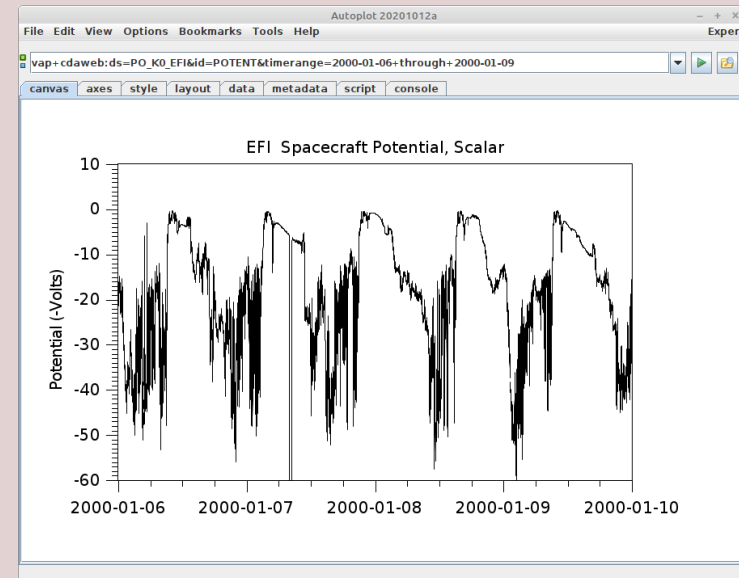
```
java -jar UriTemplatesJava.jar --parse  
--template='data_$(d;pad=none;Y=1999; m=5).dat' --name='data_1.dat'
```

Data From Servers

Autplot URIs can also point to servers to retrieve data.

- like CDAWeb:
 - `vap+cdaweb:ds=PO_K0_EFI&id=POTENT`
`&timerange=2000-01-06/2000-01-10`
- and Das2 Servers:
 - `vap+das2server:http://jupiter.physics.uiowa.edu/das/server`
`?dataset=Juno/WAV/Survey`
`&timerange=2020-07-24/2020-07-26`
- and HAPI Servers:
 - `vap+hapi:https://cdaweb.gsfc.nasa.gov/hapi`
`?id=PO_K0_EFI¶meters=Time,POTENT`
`&timerange=2000-01-06/2000-01-09`

From CDAWeb:



Data From Servers

- For each server type, custom software is written for Autoplot so that it can talk to the server.
- HAPI was motivated by a need for a common interface many servers could support.

HAPI Client Extracted

- In Autoplot's code base there was code to interact with HAPI servers, but this was also mixed in with other Autoplot and Das2 codes.
- This code has been extracted to make a stand-alone library.
- I hope that both of these libraries will be useful to those building Java software systems, but these can be useful in Python, IDL, Matlab, and Scala as well.

HAPI Client

- <https://github.com/hapi-server/client-java>
- 131 KByte jar file is available
(note JSON library is embedded within) at
<https://github.com/hapi-server/client-java/blob/master/store/hapi-client-java.jar>
- javadoc at
<https://jfaden.net/~jbf/hapi/client-java/doc/>

HAPI Client

- Catalog and info responses are returned as JSONObject, like a map containing name/value pairs.
- Convenience methods return array of dataset ids.
- Streaming version of the API reads data record-by-record.
- getAllData methods will read everything into native types (double[], String[], etc)

HAPI Client Caching

- Server responses are cached on the client side, so that repeated queries don't download the same data again and again. (The server must respond with a 304 "use what you have" response.)
- This also allows use when the Internet is not available.
- This caching is will also be supported by the Python HAPI client, so the two client implementations will share the same cache.

Use in IDL

IDL-Java bridge allows use of Java codes in IDL.

```
IDL> setenv, 'CLASSPATH=hapi-client-java.jar'
```

```
IDL> hc= OBJ_NEW('IDLjavaObject$ClientJava', 'org.hapiserver.HapiClient')
```

```
IDL> u= hc.url( 'https://cdaweb.gsfc.nasa.gov/hapi' )
```

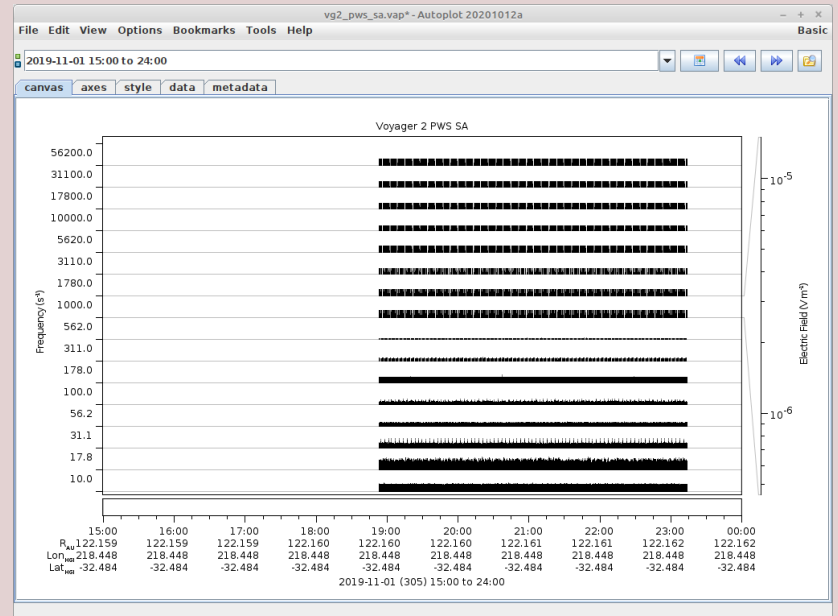
```
IDL> js= hc.getCatalogIdsArray(u)
```

```
IDL> dd= hc.getAllData(u,'PO_K0_EFI','POTENT','2000-01-06Z','2000-01-10Z')
```

```
IDL> plot(dd.get('POTENT') )
```

Autoplot new this year

- read/write IDLSave files, Matlab files
- Constant tweaks to CDF support, large files.
- ASCII/CSV readers combined
- Guess aggregation improved
- Review of “Basic” mode to support specific apps
- New Windows .exe release provided by Chris Piker at U. Iowa.
- Recent work on Mash-Up Tool



Summary

- Autoplot client is available (autoplot.org) to look at data from many sources.
- Its URI_Templates support has been extracted for use by other systems.
- A HAPI client which provides caching is available to Java systems.
- Autoplot will soon be refactored to use these libraries as well.
- Directly supported by NASA and used extensively at U. Iowa, and continues to mature with feedback from the scientists using it.