Accessing SDO Data through the VSO IDL Client (updated)

At the spring AAS meeting, we presented a poster on how to access SDO data through the Virtual Solar Observatory’s IDL client in SolarSoft, and solicited comment on additional features that scientists would like to see.

With the data now flowing, we have had a number of suggestions from the science community, and have added new keywords, clarified documentation, and hopefully improved the client overall.

We will present information both on the current status and future planned updates for the VSO IDL client, as well as solicit for additional comments on how to improve it.

Searching for SDO data in IDL:

```
IDL> aia=vso_search( start_time, [end_time], inst='aia', wave=wavelength )
IDL> results=vso_get( aia )
```

```

Interaction with the HEK

We will be modifying the VSO IDL client to accept output from Sam Freeland’s functions to search the Heliophysics Events Knowledgebase (in the ‘ontology’ package in SolarSoft) as search parameters. This will allow you to ask for data taken near the time of a given event:

```
IDL> query = vso_her_make_query(’2010-05-01’,’2010-05-18’,/FL, $search_array=[’langle;ge;>31’])
```

```
IDL> events = vso_her_query( query )
```

```
IDL> aia = vso_search( inst=’aia’, events=events fid() )
```

In the future, we hope to use the HCR (Coverage Registry) to support spatial searching for cutouts of SDO data as well.

STATUS: In development, need syntax for time windowing; we hope to have a solution by mid-January.

VSO Registry Search

As new series are made available through VSO, users need to be able to easily find what datasets and options are available:

```
Information about using vso_search() and vso_info():
```

```
IDL> vso_info
```

```
Listing the instruments available through VSO:
```

```
IDL> vso_info, /instruments
```

Getting a structure with the instrument list:

```
IDL> vso_info, output=output, /instruments
```

```
STATUS: Basic listings completed; additional work to be done to allow restricting the list (eg, list only instruments that produce images in 175 Ångstrom, or only those observing on a given day)
```

VSO Online Help

As the VSO uses a series of wrappers around a main object, and the use of `extra’ to pass values, many IDL commands to tell you what the appropriate variables are don’t return anything useful. Therefore, we have a command `vso_help’ that gives information about the VSO IDL commands:

```
IDL> vso_help
```

```
IDL> vso_help, search ; docs for the vso_search command
```

```
IDL> vso_help, get ; docs for the vso_get command
```

```
IDL> vso_help, info ; docs for the vso_info command
```

```
STATUS: Done; in SolarSoft
```

```
we hope to expand it to pull news from an RSS feed and include documentation on the vso object.
```

Download Tarballs

Dominic Zarro’s HTTP client that the VSO IDL client uses has been modified to support downloading files without knowing the size of the file. This allows us to stream tarballs as they are generated without the need for staging the data. The HTTP client also now supports the Content-Disposition header, allowing web services to suggest filenames for files to be transferred, rather than using the URL of the program generating the tarball.

```
STATUS: Complete, in SolarSoft; however work is underway to use staged data to reduce the overhead at the caching nodes due to memory consumption.
```

Staged SDO Data

Streaming of the data creates problems if your connection breaks, as you cannot continue downloading, you must start all over again. For those with spotty internet connections, we are implementing staging the data. You will need to provide an e-mail address to be notified of when the data is available for download.

```
status = vso_get( ... , /norice )
```

```
status = vso_get( ... , email=’ ... ’ , /staging )
```

```
download.
```

```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```
```