Guide to SDO Data Analysis

edited by
Marc DeRosa & Greg Slater

December 10, 2010

Contents

1 Introduction to SDO

1.1 Synopsis of the SDO Mission

The Solar Dynamics Observatory (SDO) is the first mission to be launched under NASA's Living With a Star (LWS) program. This program involves a diverse set of research topics that aim to provide a greater understanding how the activity and variability of the Sun affect life on Earth. The purpose of SDO is to understand how the Sun's magnetic field is generated and structured, and how this stored magnetic energy is converted and released into the heliosphere and geospace in the form of solar wind, energetic particles, and variations in the solar irradiance.

SDO was launched aboard an Atlas V rocket on 11 February 2010 from Kennedy Space Center in Florida. It is now located in a nearly geosynchronous orbit that allows continuous contact with its dedicated ground stations in New Mexico. The observatory has been operating nominally since launch, without any major problems.

The observatory contains three instruments (presented in alphabetical order):

- The Atmospheric Imaging Assembly (AIA) is an array of 4 telescopes that together provide full-disk images of the solar corona at $1''$ resolution (4096×4096-pixel images) in 10 UV and EUV wavelengths every 10 seconds;
- The Extreme ultraviolet Variability Explorer (EVE) measures the EUV spectral irradiance with unprecedented spectral resolution, temporal cadence, accuracy, and precision; and
- The Helioseismic and Magnetic Imager (HMI) provides full-disk, high-cadence Doppler, intensity, and magnetic images at $1''$ resolution (4096×4096-pixel images) of the solar photosphere, allowing studies of the sources and evolution of activity within the solar interior.

Together, this suite of instruments enables the monitoring of the solar interior, chromosphere, and corona with high spatial and continuous temporal coverage. Low-level processing and calibration for data from the two imaging instruments (AIA and HMI) occurs at the Joint Science Operations Center (JSOC), located on the campus of Stanford University. EVE data is processed at the Laboratory for Atmospheric and Space Physics (LASP) at the University of Colorado. Brief descriptions of the data products produced by each of the three SDO instruments are given below in Sections 1.3, 1.4, and 1.5.

1.2 About this Guide

The massive amount of image data produced by SDO (of order several terabytes per day) requires new techniques for processing, browsing, retrieving, and analyzing these data. Indeed, significant samples of full-disk, full-resolution images from either AIA or HMI can easily exceed hundreds of gigabytes. Consequently, remote users will not be able to download very much SDO image data before it is realized that either one's internet bandwidth is not sufficient to download the amount of data requested in a reasonable amount of time, or one's available on-site computing power and storage are not sufficient to interact or process the data once it resides locally (or both).

The purpose of this guide is to demonstrate the ways in which users can more readily progress through the sequence of browsing, finding, downloading, and (eventually) analyzing SDO data. The guide is a living document, and materials will be added as needed.

The guide contains sections describing:

- **How to browse data**: Various "Sun-in-time" pages, summary images and movies, and popular event and feature listings (Section 2)
- **How to find data**: Heliophysics Event Knowledgebase, a searchable database that is useful for finding events and features of interest during (and prior to) the SDO era for which the dates are not known (Section 3)
- **How to get data**: Multiple interfaces to the data centers through which SDO data can be downloaded (Section 4)

- **How to find and get data using SolarSoft**: Demonstration of SolarSoft IDL commands useful for finding and downloading data (Section 5)
- Listings of **Frequently Asked Questions** (Section 6) and **useful links** (Section 7)

The online version of this guide can be found either at http://www.lmsal.com/sdouserguide.html, or by navigating to the SDO Documentation (SDOdocs) webpage where it is the first link in the list. Alternatively, a PDF version of this guide is also available.

1.3 Data Products from AIA

AIA obtains full-Sun images in multiple EUV and UV passbands, as summarized in Figure 1 and described in more detail on the AIA Instrument webpage. These images are used to generate higher-level data products, including browse-quality data and feature and event entries in the Heliophysics Events Knowledgebase (see Section 3.1), all of which in turn support the science requirements of AIA. A summary of this process appears in the flowchart presented in Figure 2. Links to more detailed information on the production of the science data can be found on the AIA Data webpage.

The different data-processing levels for AIA are summarized as follows, with more details available from this source:

- Data at **Level 0** are images that have been constructed from the raw telemetry stream.
- Data at **Level 1.0** are images that have been converted from Level 0, with processing including bad-pixel removal, despiking and flat-fielding. All higher-level products for AIA are based on Level 1 data.
- Data at **Level 1.5** are images that have been adjusted to a common $0.6''$ plate scale, and that share common centers and rotation angles, but are *not* exposure-time corrected.

1.4 Data Products from EVE

EVE produces a multitude of spectral and photometer data products that support the EVE primary science objectives. these data products are summarized in Figure 3 and available through the EVE data access webpage.

1.5 Data Products from HMI

HMI produces a multitude of data products that support the HMI science investigation. The four main observables are: Dopplergrams (maps of solar photospheric velocity), continuum images (images of the solar photosphere near the 6173Å absorption line of Fe I), and both line-of-sight and vector magnetograms (maps of the photospheric magnetic field). For more details and specifications, see the HMI instrument overview webpage at Stanford, or the accompanying analysis pipeline flowchart in Figure 4.

The different data-processing levels for HMI are summarized as follows, with more details available from this source:

- Data at **Level 0** are images that have been constructed from the raw telemetry stream.
- Data at **Level 1.0** are images that have been converted from Level 0, with processing including bad-pixel removal, flat-fielding, and quality assessment checks, but otherwise not having undergone any irreversible data alterations.
- Data at **Level 1.5** are images of the physical observables (Dopplergrams, magnetograms, and continuum images), which were constructed from the individual Level 1.0 filtergrams.
- Data at **Level 2** have been irrevocably filtered, time-sequence-merged, Fourier-transformed or otherwise changed from Level 1.5 data in a way that is irreversible. Level 2 products include intermediate products for later production of mission science data products, such as helioseismic inferences of solar subsurface flows.

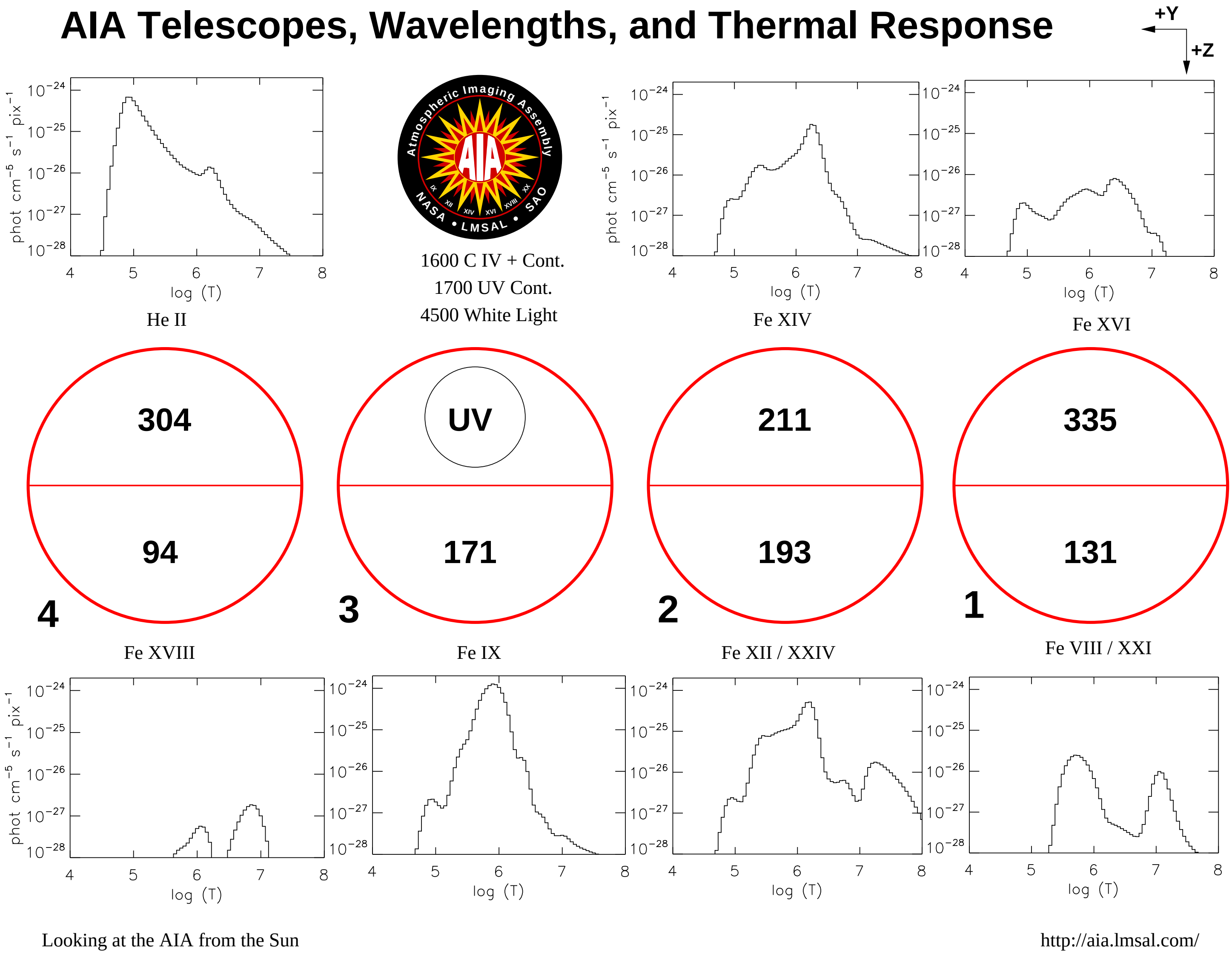


Figure 1: A summary of the passbands of AIA, showing the filter responses of each of the EUV channels, and their arrangement on the instrument as viewed from the sun (sourced here).

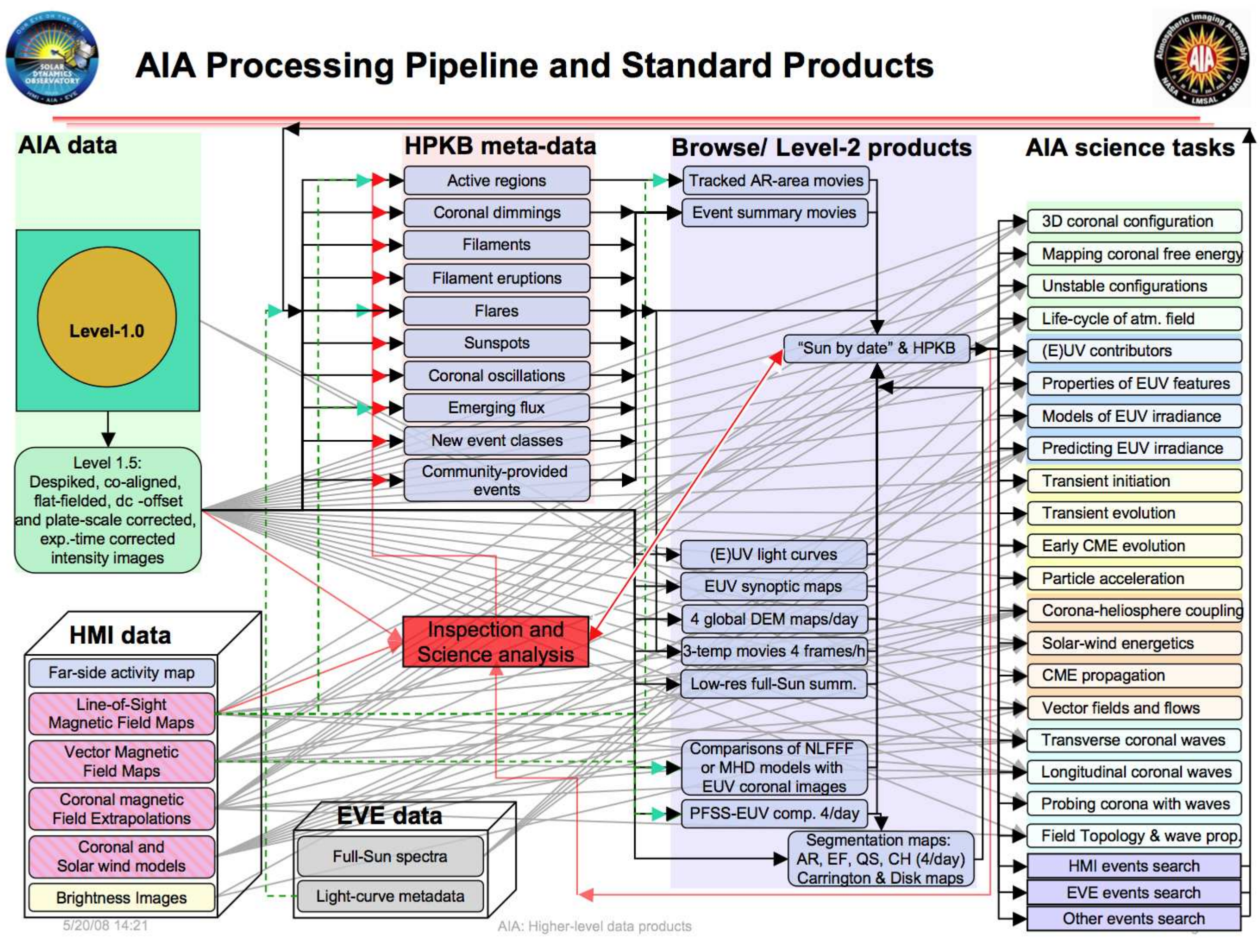


Figure 2: AIA science analysis pipeline, illustrating how the main filtergram data, along with data from EVE and HMI produce the data products that support the science goals of AIA. Note that AIA Level-1.5 data are *not* exposure-time corrected (as is incorrectly indicated in the figure).

Level	Description	Components	Wavelength Coverage	Wavelength Sampling	Temporal Sampling	Time Span of Data File	Daily size (GB)	Latency of Availability
L0C	**Space Weather Product:** Crudely calibrated irradiances* (from Ka-Band data)	ESP bands + quads (flare)	0.1-7, 18.2, 25.6, 30.4, 36.6 nm	broadband ~4-nm	1-min	Latest 15-min and current 1-day (growing file)	0.004	<15 min
		MEGS-P	121-122 nm	1-nm				
		MEGS-A, B	5-105 nm	1-nm	1-min		0.005	
		MEGS-A, B, proxies	Select lines and bands**	Varies by band	1-min		0.01	
L0CS	**Fastest Space Weather Product**: Crudely calibrated irradiances* with least latency (from S-Band)	ESP bands + quads (flares)	0.1-7, 18.2, 25.6, 30.4, 36.6 nm	broadband ~4-nm	1-min	Latest 15-min and current 1-day (growing file)	0.005	< 1 min
		MEGS-P	121-122 nm	1-nm				
		XRS & SEM model	Proxies	Varies by band				
L1	**Photometer Data:** fully calibrated and corrected photometer irradiances	ESP	0.1-7, 18.2, 25.6, 30.4, 36.6 nm	~4-nm	1/4-sec	1-hour	0.03	1 Day
		SAM	0.1-7 nm***	0.1-1-nm	1- & 5-min		varies	
		MEGS-P	121-122 nm	~1-nm	1/4-sec		0.006	
L2	**Spectra**: fully calibrated and corrected spectral irradiances at instrument resolution	MEGS-A, B	5-105 nm	0.02 nm	10-sec	1-hour	1.2	1-2 Day
L2	**Lines & Broadband irradiances:** fully calibrated and corrected photometer irradiances and extracted spectral lines and bands	MEGS-A, B, P, ESP	select lines & bands	Varies by band	10-sec	1-hour	0.01	1-2 Day
L3	**Merged Spectra:** fully calibrated, corrected, and merged spectral irradiances	ESP, SAM, MEGS-A, MEGS-B, MEGS-P	0.1-105 nm	0.02, 0.1 & 1 nm	1-day	1-day	<0.001	1-2 Day

*All products are corrected to 1-AU except L0C and L0CS.
** Lines spanning Log T = 3.8-7.1, plus AIA and ESP bands.
*** SAM is a research project, L1A will have 4 element event list: time, location (x,y), and energy.

Figure 3: Table of EVE data products (sourced here), showing the range of wavelengths covered, their resolution, and the different levels of processing applied.

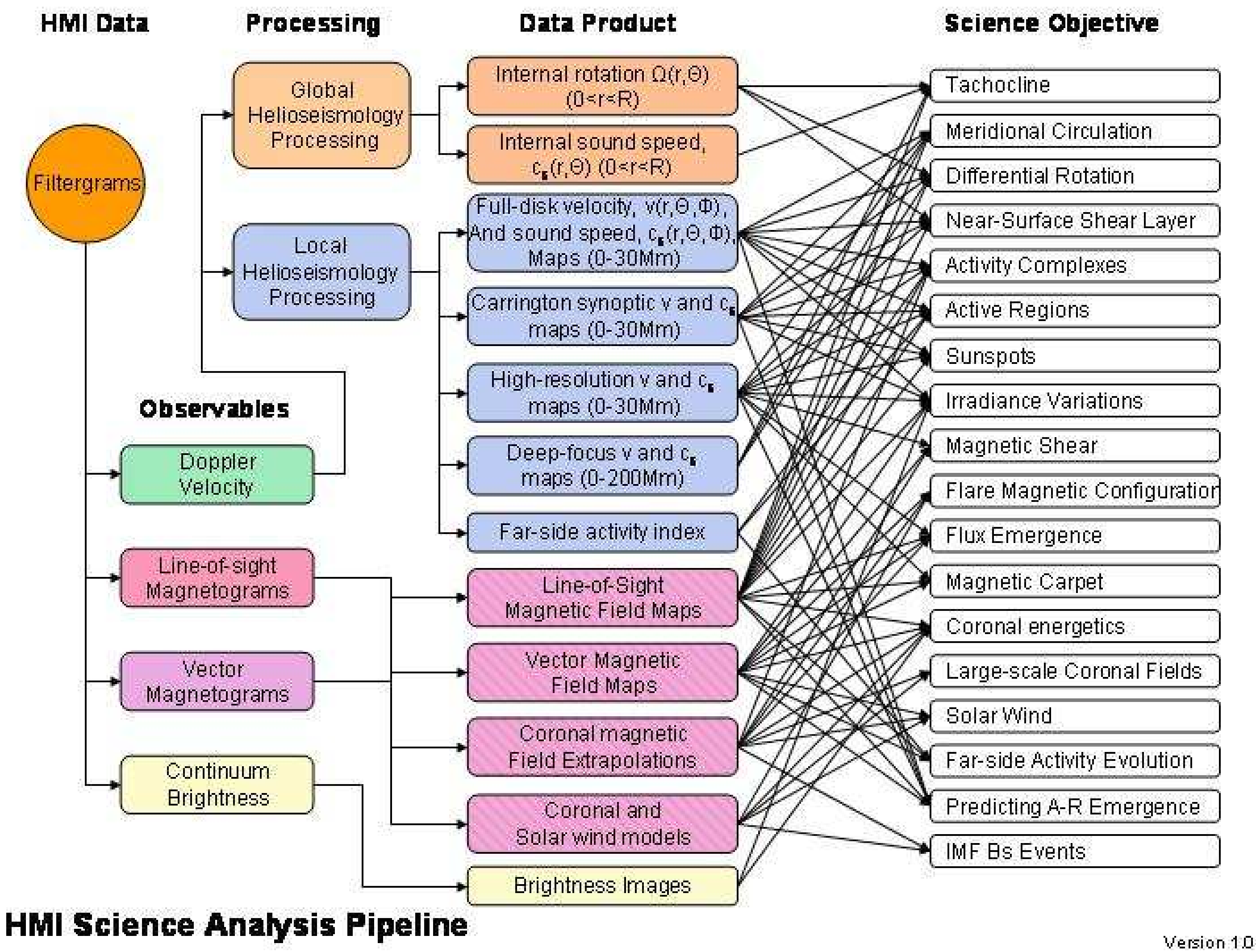


Figure 4: HMI science analysis pipeline flowchart (sourced here), showing how the four main observables are used to create the many data products that support the science objectives of the HMI instrument.

2 How to Browse SDO Data

In this section, we briefly illustrate a few of the online resources that allow users to browse SDO data: "The Sun Now" (Section 2.1), "The Sun Today" (Section 2.2), SolarMonitor (Section 2.3), and Helioviewer and JHelioviewer (Section 2.4).

2.1 "The Sun Now"

The SDO mission webpage hosted by NASA's Goddard Space Flight Center contains a wealth of information about SDO. The first entry under the "Data" tab is "The Sun Now", which contains thumbnail images, links to full-size and downsampled images, movies, and composite images from AIA and HMI. Also present are a soft X-ray image from the pinhole camera on EVE as well as a heliographic representation of the corona assembled from the two STEREO spacecraft and AIA. The other entries under the "Data" tab provide additional functionality, including templates that enable users to generate a customized image sequence or movie. The top portion of "The Sun Now" webpage is shown in Figure 5.

2.2 "The Sun Today"

"The Sun Today", located at **`http://sdowww.lmsal.com/suntoday.html`**, is a webpage summarizing images, movies, events, and features that occurred on the Sun through any given day in the SDO mission. The page keys off the date indicated at the top of the page, which can be changed to any date of interest either by using the plus and minus buttons in the upper-right corner of the webpage, or by entering a URL with the **`suntoday_date`** parameter set to the date of interest. For example, this link (**`http://sdowww.lmsal.com/suntoday.html?suntoday_date=2010-08-01`**) takes the user to the summary page for 2010 Aug. 1, as illustrated in Figure 6. The page layout (currently) includes three panels that contain: a series of AIA and HMI images for the day of interest, the results of an iSolSearch query (see Section 3.2) for that day, and a plot of AIA light curves for that day.

The image panel contains a row of four preview images, under which are lists labeling various wavelength channels from AIA and HMI. In each column of labels, the label of the channel that is underlined corresponds to the preview image atop the column. Mousing over any of the other wavelength channel labels changes the preview picture, enabling users to compare images from up to four channels at once.

Clicking on the links next to each wavelength channel will open a new browser window containing a larger image from that channel. For example, the "1K" link will open a 1024$\times$1024-pixel image, the "4K" link will open a 4096$\times$4096-pixel image, and the "PFSS" link will open a 4096$\times$4096-pixel image with fieldlines from a potential-field source-surface model overlaid. Links to summary movies of four of the wavelength channels appear above the row of preview images.

To put these images in some context, the results of an iSolSearch query and a series of selected AIA light curves for the date of interest are displayed in the panels underneath the image preview panel. For more information on iSolSearch, as well as instructions on how to use the full search tool, please see Section 3.2.

2.3 SolarMonitor

The SolarMonitor website provides a straightforward way to browse a variety of solar imagery, including magnetograms, continuum images, Hα images, and images in ultraviolet and X-ray passbands. The front page for 2010 Aug. 3 is shown in Figure 7 below. Zoomed-in summary images for each NOAA-identified active region are also available (search for these first by date, and then use the links along the left-hand sidebar or underneath the image display to view the active-region summaries). Links along the right-hand sidebar will take the user to external websites that provide other ways to monitor the sun, space weather, and the heliosphere.

2.4 The Helioviewer Project

The helioviewer.org data browser is a user-friendly way to align and display multiple images from a selection of recent solar and heliospheric observatories, as shown in Figure 8. The tool consists of a web-based interface that enables users to interactively pan and zoom an image, overlay images from different observatories with the proper registration, and indicate the location of features and events

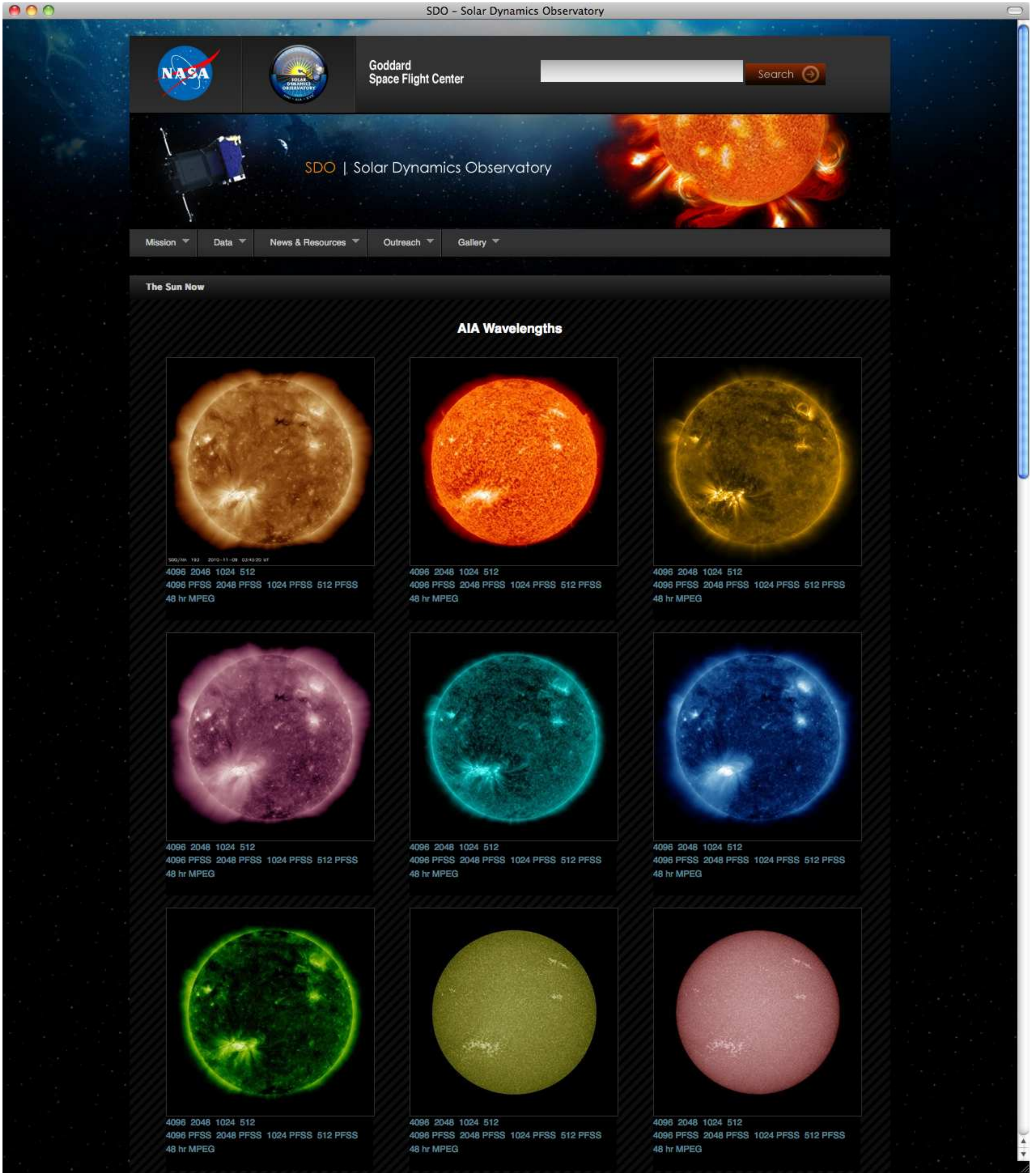


Figure 5: The top portion of the "The Sun Now" webpage showing the array of AIA image channels.

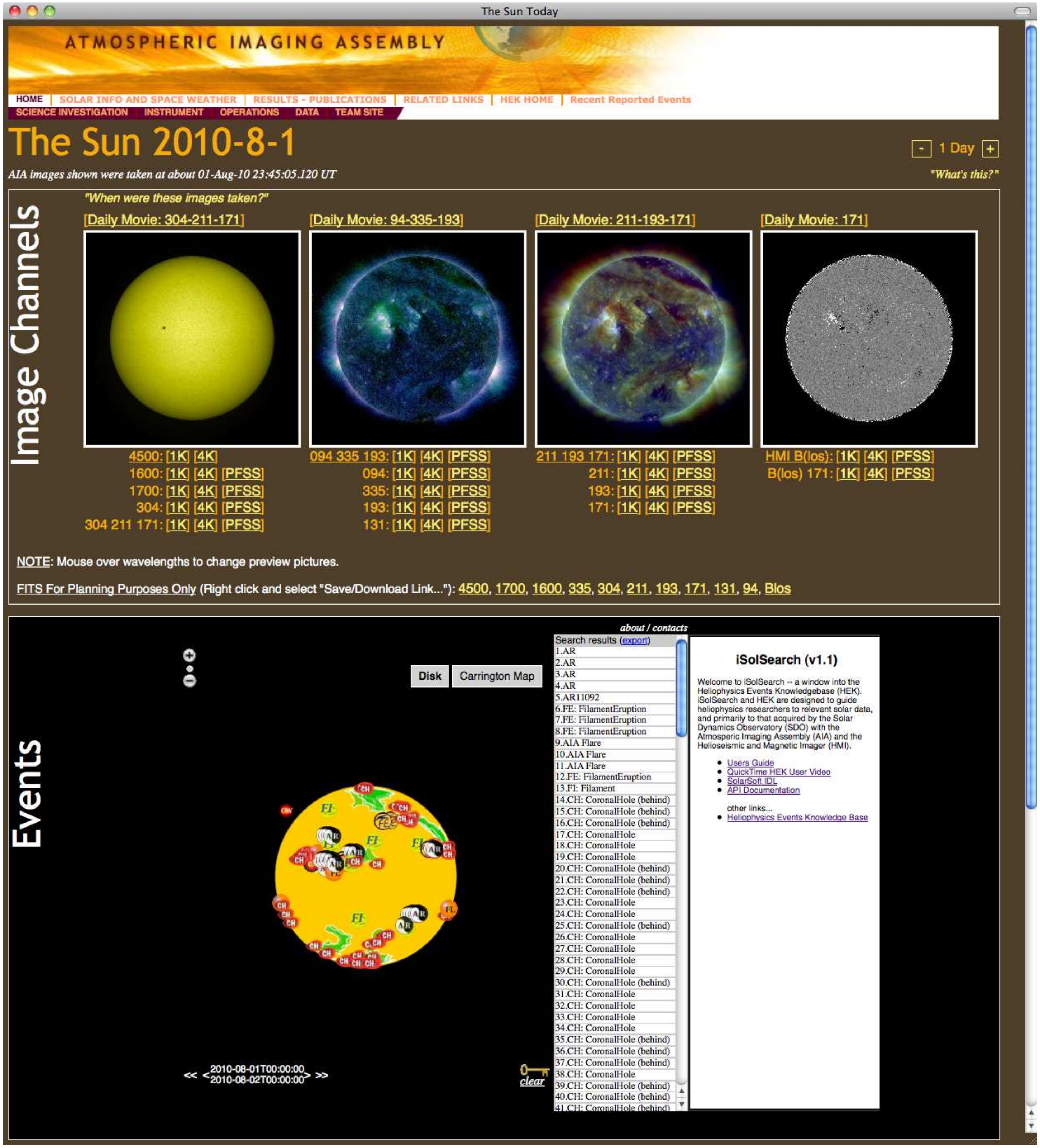


Figure 6: The top two sections of “The Sun Today” webpage for 2010 Aug. 1, showing the assortment of preview images as well as the results of an iSolSearch query for that day.

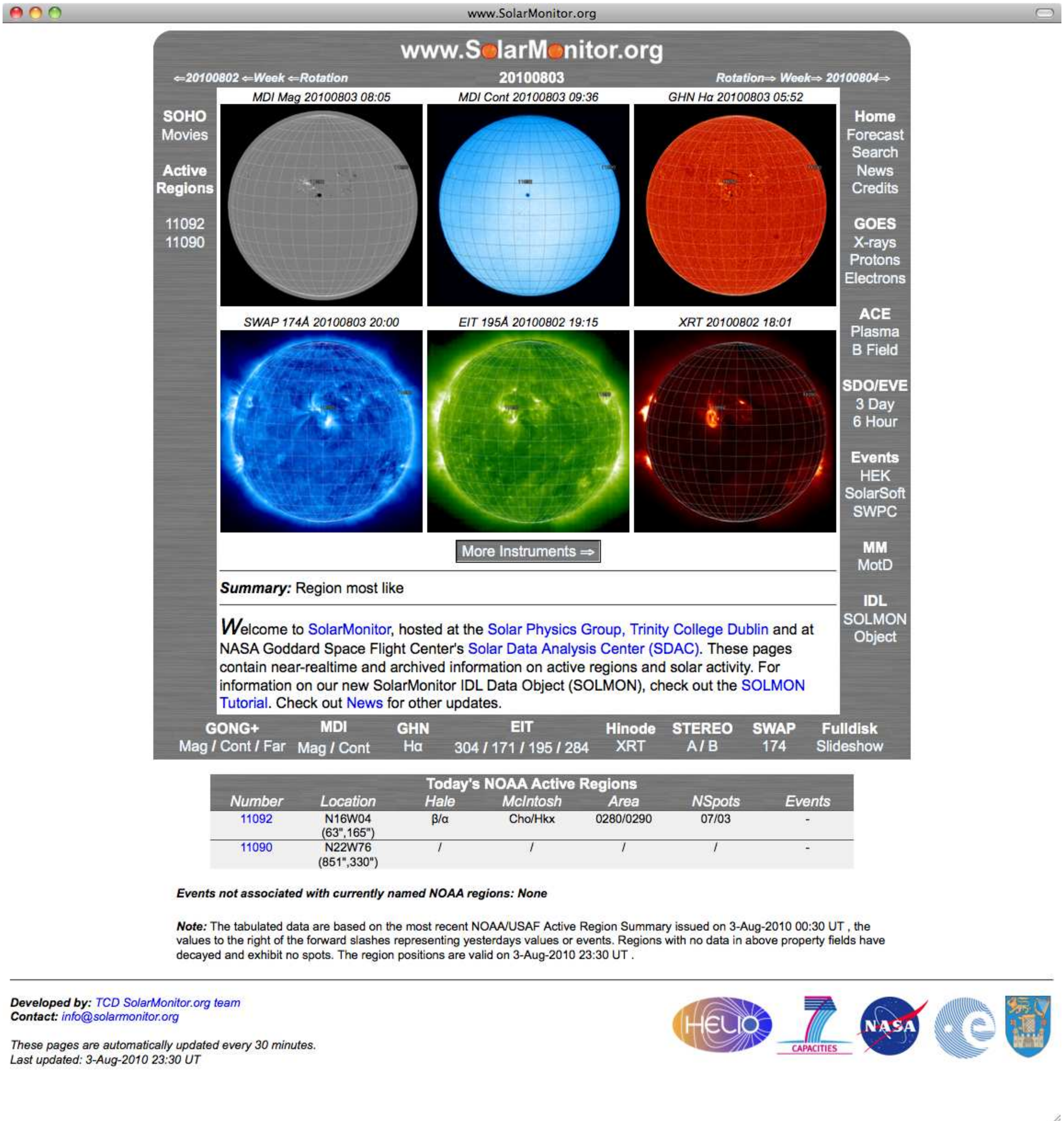


Figure 7: The SolarMonitor.org front page for 2010 Aug. 3.

of interest. The interface downloads image and event data asynchronously (using AJAX), which allows the user to continue browsing while the tool downloads additional data. For more information, please visit the website and/or refer to the Helioviewer.org User Guide.

JHelioviewer is a Java/OpenGL-based application to browse time series of solar images. JHelioviewer overlays image series from SOHO and SDO in space and time and makes use of the JPEG2000 standard to stream large data sets interactively in real-time. Users can, for example, zoom and pan full-size SDO movies, apply basic image processing methods (including feature tracking) and overlay event markers from the Heliophysics Event Knowledgebase (see Section 3.1). JHelioviewer is open-source and has a plugin architecture that allows users to expand its functionality. More information can be found by visiting the website, and in particular reading the JHelioviewer Handbook.

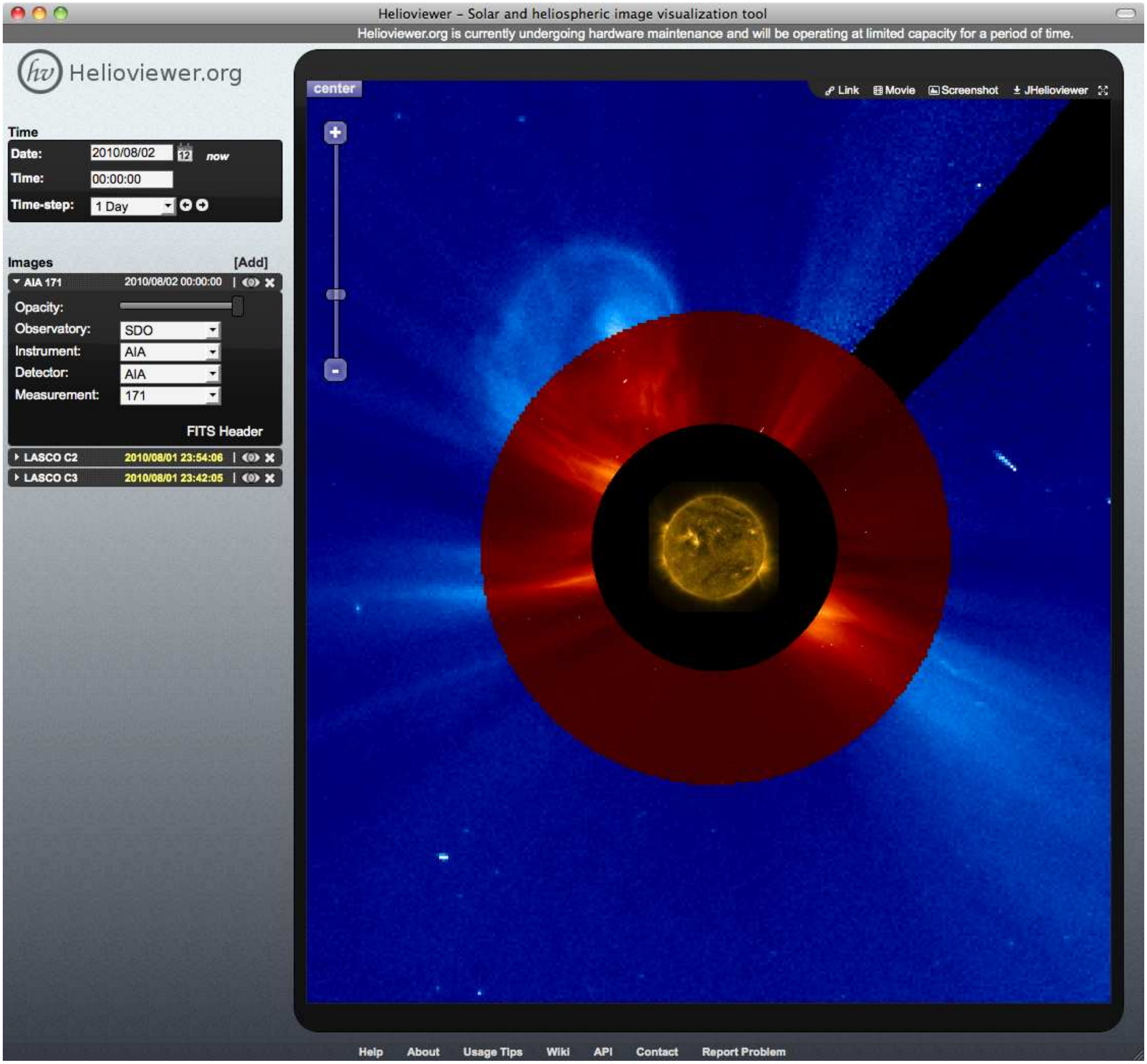


Figure 8: The helioviewer.org display showing an eruption visible in LASCO C2 and C3 images, as well as a (mostly) cotemporal and coaligned image from the AIA 171Å channel. The three images are all taken within 15 minutes of 2010 Aug. 2, 00h UT.

Figure 9: A screen shot of the JHelioviewer application illustrating a zoomed-in portion of an AIA image with icons associated with Heliophysics Events Knowledgebase overlaid on the image.

3 How to Find SDO Data

In this section, we will demonstrate how to use the Heliophysics Events Knowledgebase (HEK). The HEK is essentially a list of observational sequences combined with a catalog of features and events that have occurred on the sun. It is useful in situations when the user knows *that* a particular type of event occurred, but who cannot remember exactly *when*. It is also useful in situations when the user wants to know whether a particular sequence of observation has occurred.

The HEK allows users to report new features and events, and to contribute information on existing ones. Consequently, the HEK is useful in an environment where browsing large amounts of data is cumbersome (as will be the case during the SDO era), and allows users to focus on establishing connections between similar events and features. More information on the HEK is available in this article, a preprint of which is available here (PDF file), and which will eventually appear in Solar Physics.

3.1 Heliophysics Events Knowledgebase

The HEK is comprised of two major components: the Heliophysics Events Registry (HER) and the Heliophysics Coverage Registry (HCR). The HER tells you what features and events have been identified, and the HCR tells you which data sequences are available. There is an interplay between the HER and HCR (see Figure 10), as searches for entries in the HER also reveal related observations cataloged in the HCR. We now discuss each one in sequence.

As illustrated in the schematic in Figure 11, the HER accepts data from a multitude of providers, and various Feature Recognition Algorithms (FRMs) operate on these data streams and report events and features of interest. FRMs encompass both automated feature-finding algorithms, or can be human annotators. Users access the HER by tools such as iSolSearch (see Section 3.2), or can use the specifications given by the HEK Application Programming Interface (API). Additionally, for users familiar with SolarSoft, there exist SSWIDL routines for querying with the HER (see Section 5.2).

Each feature or event entry in the HEK contains information about its data source (e.g., observatory, instrument, wavelength), its spatio-temporal location, and the FRM (automated or human) used to identify the item. The full list of event classes tabulates the required and optional attributes for each class of event, and it is these attributes than can be used to query the HEK.

Because the name of the FRM is an attribute for each event or feature stored within the HEK, multiple FRMs can generate events belonging to the same event class. There are many cases where the same physical feature has been identified by different FRMs, but no choice is made as to which one is "better" or "more useful" as such qualitative assessments often depend on their intended use within a research program. Using coronal mass ejection (CME) events as an example, multiple catalogs of CMEs based on different observatories and/or event specifications (such as, for example, the LASCO (CDAW) CME Catalog, the SEEDS Catalog, the LASCO ARTEMIS Catalog, the STEREO COR1 Catalog, and the Cactus CME Catalog) are already in use by the community, and it is likely that the research community has benefited from having more than one independently generated catalogs of the same event. The same benefits can be expected by being able to query events from multiple FRMs seeking to identify each event class in the HEK.

The HCR is a tool that tracks data requests. Users can, for example, view recent or view popular data sequences from AIA, or view the recent events from SDO reported by human observers. Users can make more advanced searches using the HCR search form. As data (especially, the starting and ending dates, instrument, and wavelength) are entered into the HCR search form, the number of observational sequences that satisfy the query is updated at the bottom of the page.

Clicking on the "Show Search Result Details" button reveals more details about each observation, including pointing information and a brief description of the science goal of the observation. These observations should be readily available via the "Get All Data" link on the summary page for each observation.

3.2 iSolSearch

The easiest way to interact with the HEK is probably by using iSolSearch, which enables users to easily query the HEK for a listing of events between two points in time. In this section, the basic usage of iSolSearch is summarized. Alternatively, there is a brief YouTube video (with audio) that also demonstrates the basic usage of the iSolSearch tool. Users are encouraged to ask questions in the YouTube comments section.

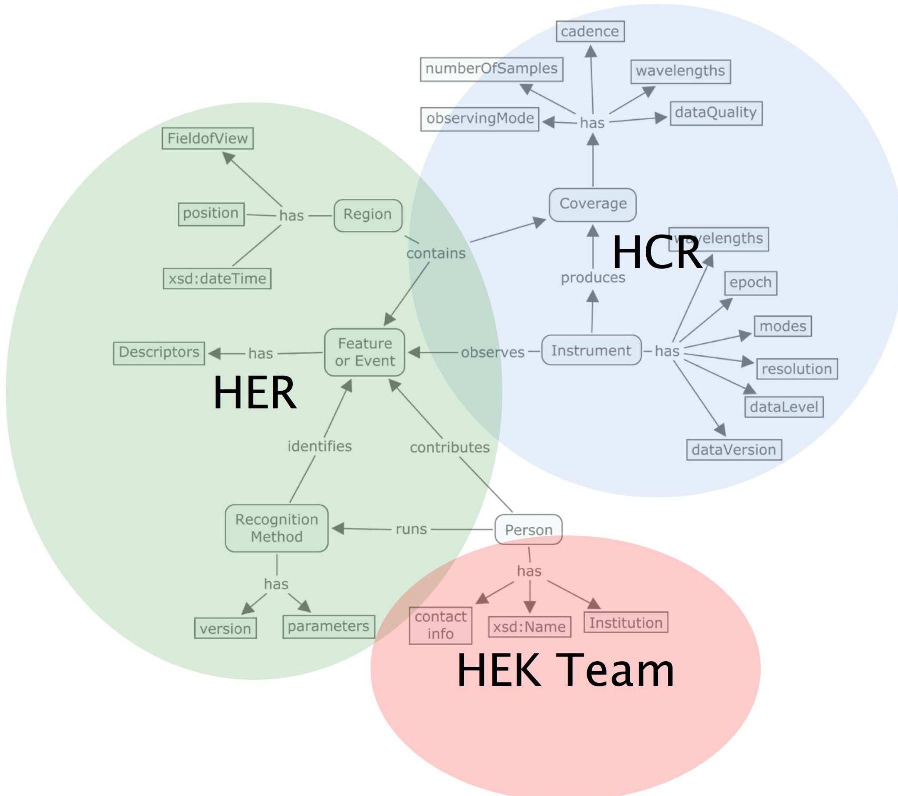


Figure 10: A schematic illustration of the operation of the HEK, showing the interrelationship between the HER and HCR.

To query the HEK and learn what happened on the sun during a particular time frame, simply enter two dates in the search panel, choose a selection of event classes, and press the "Search" button underneath the event-class listing. Any events in the HEK that satisfy this query will appear as icons on the disk view panel, as well as in a listing in the results panel to the right of the disk view. The results from a typical search are illustrated in Figure 12.

Mousing over a particular icon will on the disk view will cause it to be highlighted in yellow in the event listing. Likewise, mousing over an event in the event listing will enlarge the icon in the disk view. When clicking on an event (either its icon in the disk view, or its corresponding entry in the event listing), the tool will display more detailed information about this event in the gray information panel, as shown in Figure 13. There, the user will find a more detailed description of the event, a summary image or movie (if available), and a link to the underlying XML code (in the "VOEvent XML" link) associated with this event. Users can also add a reference or comment regarding this event, and rate this event, by clicking on the small icons at the top of the information panel.

Observational sequences that overlap the selected event in both space and time, as found from a HCR search based on the event properties, are listed under the "Observations in the neighborhood" heading in the information panel. Clicking on one of these related observational sequences will reveal a summary page as well as a link to related data sequences that already exist (thus allowing users to avoid longer wait times when requesting data). If none of the HCR observational sequences satisfy the needs of the user, clicking on the "Request data" link leads to the request form for the cutout service (see Section 4.3), where data customized by the user can be requested.

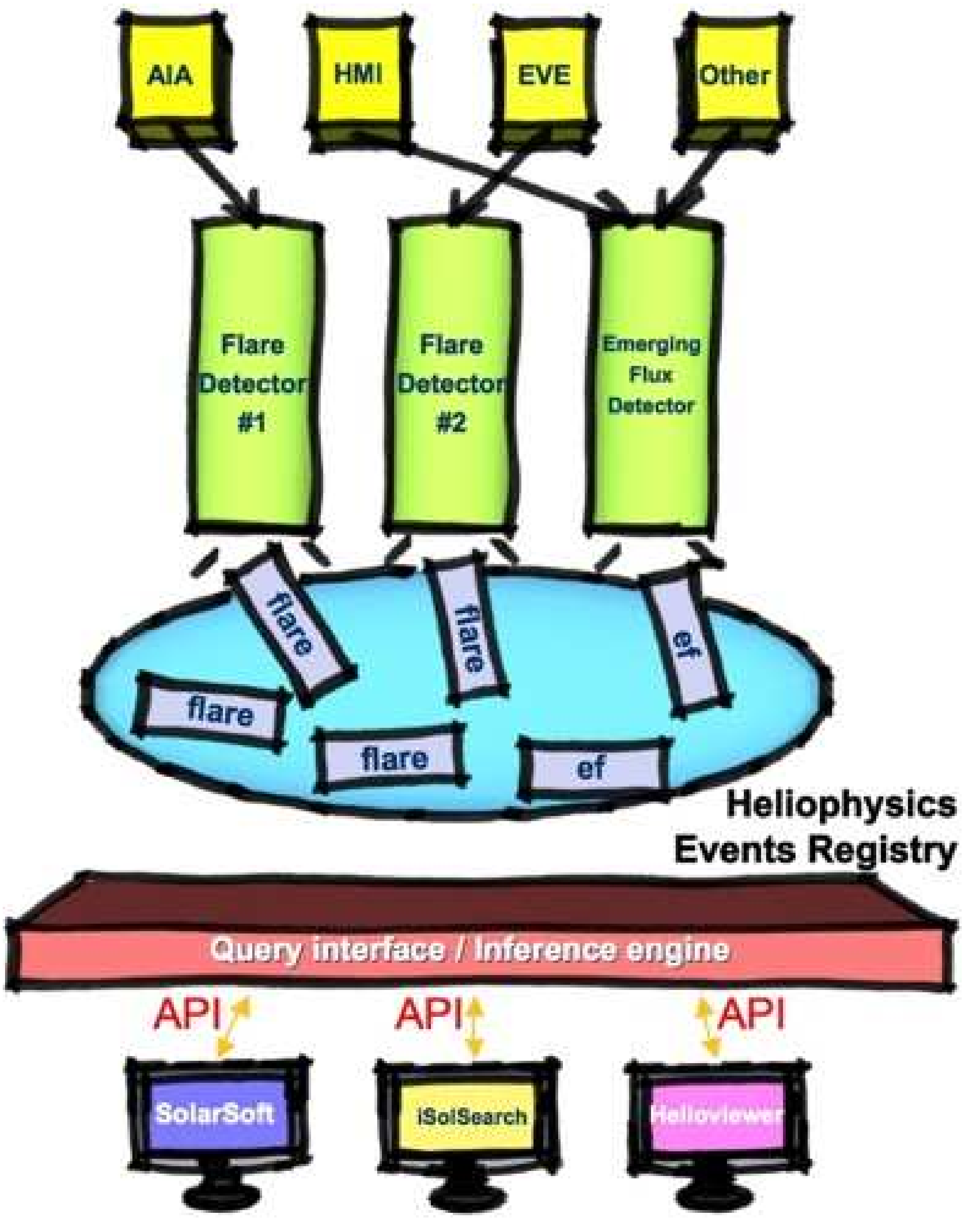


Figure 11: A schematic illustration of the operation of the HER.

To customize the event display, the disk view panel has clickable buttons that zoom in and out in the upper-left corner, and clickable buttons that switch between an orthographic and (Carrington) latitude-longitude projections of the sun in the upper-right corner. Zooming in and out can also be done using the mouse scroll wheel when the cursor is located within main portion of the disk view panel. In the lower-left corner of the disk view panel, clicking on the arrows will rerun the existing query after adding or subtracting one day (single arrows) or one week (double arrows) to or from the start and end dates of the existing query. Clicking on the key icon in the lower-right corner will take the user to the aforementioned list of event classes, where the icons in the disk view panel are defined in terms of the corresponding event classes.

More complex queries can be constructed by clicking on the "Filters" tab in the search panel. Users can add multiple filters that screen for, or screen out, events that satisfy a set of user-defined conditions. As an example, suppose one wants to find all flares having GOES Class M2 or greater that occurred in the year 2010. Under the "Search" tab, set the start and end dates to **`2010-01-01`** and **`2010-12-31`**, respectively, and clear all of the event-class checkboxes except for "Flare". Then, under the "Filters" tab, adding a constraint like **`FL_GOESCls >= M2.0`** is achieved by choosing **`FL_GOESCls`** (flare GOES class) from the "Attribute" drop-down menu, setting the operator ("Op") to **`>=`**, choosing **`M2.0`** from the "Value" drop-down menu, and then pressing the "Add" button. After pressing the "Add" button, the constraint should appear in the white box. After going back to the "Search" tab, press "Search" button to query the HEK. At the time of this writing, this query returned 18 events.

After the query has produced a listing of results within iSolSearch, the query itself and the associated list of results are available by clicking on the "export" link at the top of the panel showing the event listing. In the menu that appears, the "Query link" will open a window containing a URL of the query and the "SolarSoft call" link will open a window containing the same query formatted as an SSWIDL command using the SolarSoft API (thus allowing the event listing to be exported to SolarSoft). The listing of results is available by clicking on the "Table(CSV)" link. In this way, queries and their results can be saved for later use, or published in journal articles so as to be repeatable by other researchers. Furthermore, users can even subscribe to the query (allowing the listing of results to be updated, and the user notified) by clicking on the "RSS" link in order to set up an RSS feed.

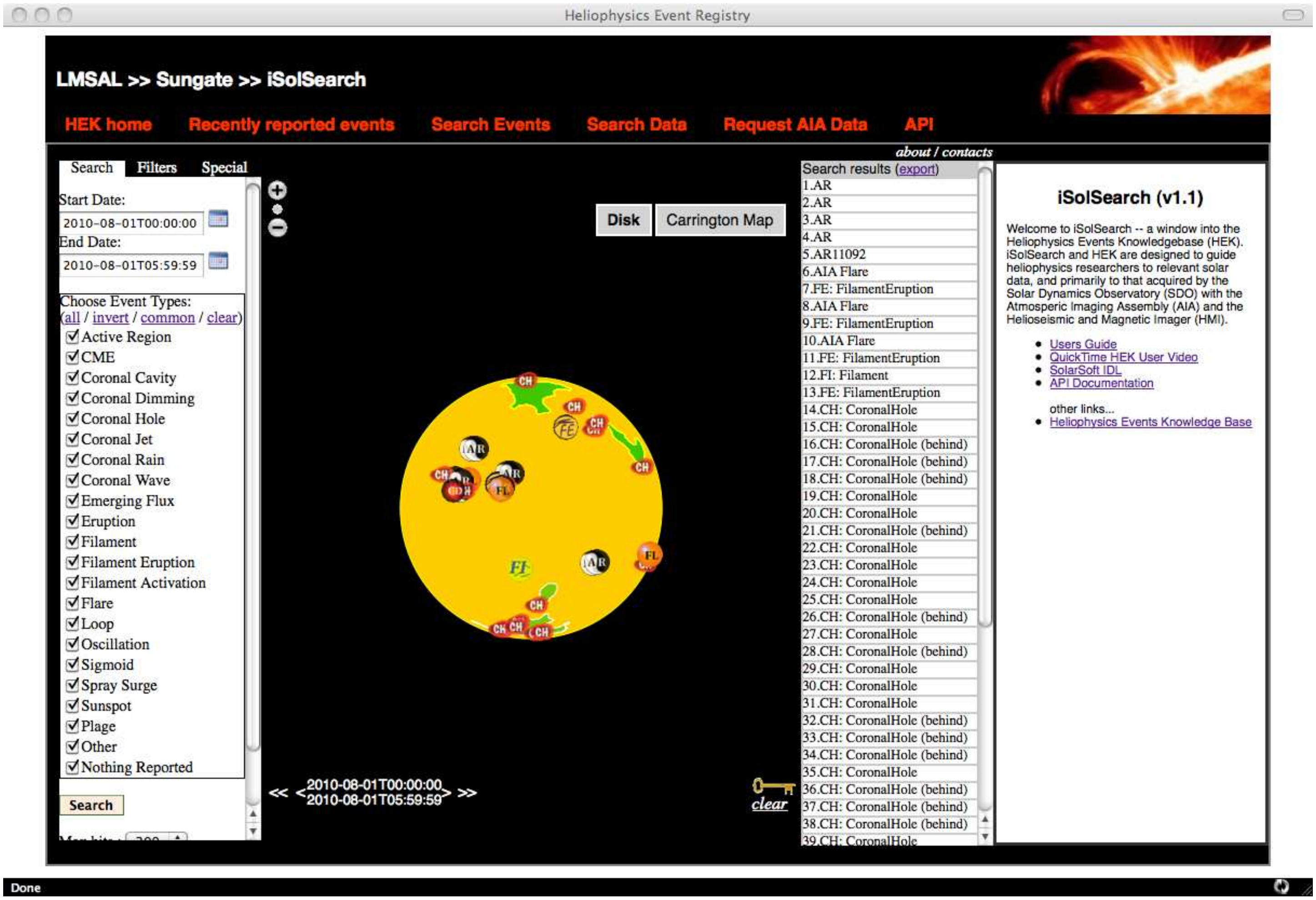


Figure 12: The iSolSearch tool, showing the results of a query for 2010 Aug. 1. The tool displays an assortment events that happened on this day, including several filament eruptions, coronal holes, active regions, flares, and even a coronal dimming event. The various events are displayed as icons (based on event class) in the disk view, and as a list in the results panel to the right of the disk display.

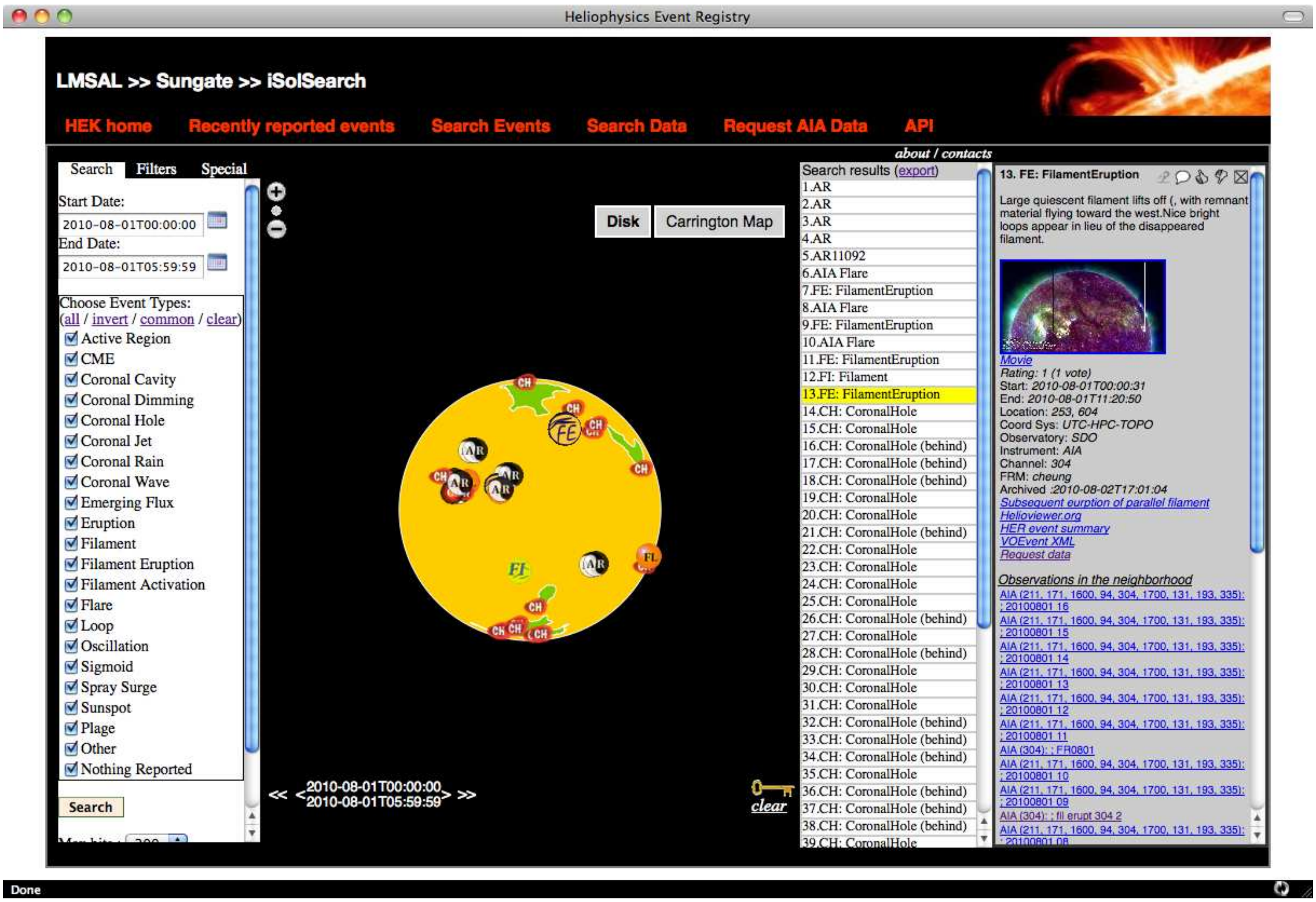


Figure 13: The iSolSearch tool, showing the gray information panel that provides more details about the filament eruption event from the results of the query shown in Figure 12. In the right-most panel, more details about this event are shown, including a description, links to a summary frame and movie, the starting and ending time, observatory, and FRM. Additional links to related resources also appear, including a link to the cutout request form that will allow users to download the associated data underlying this event from AIA. The icons at the top of the information panel permit users to add references or comments and rate the particular event being displayed.

4 How to Get AIA and HMI Data

Low-level processing and calibration for all HMI and AIA observables occurs at the Joint Science Operations Center (JSOC), located at Stanford University. After such "pipeline" processing has taken place, the JSOC is responsible for cataloging and archiving the data products and serving it to all users. In this section, the basic elements of the JSOC are briefly described (Section 4.1), followed by descriptions of various ways to retrieve data (Sections 4.2, 4.3, and 4.4). As the data are typically exported in a compressed format, the final section (Section 4.5) illustrates the various methods of uncompressing and reading in the data. Additionally, for users familiar with SolarSoft, there exist SSWIDL alternatives for retrieving data from the JSOC (see Sections 5.3, 5.4, and 5.5).

4.1 How Data is Organized at the JSOC

To better comprehend what is happening when retrieving data, it is useful to have a general understanding of how the data are cataloged in the JSOC. The system employed by the JSOC is called the Data Record Management System (DRMS), and was created by software developers at Stanford to manage the large volume of data and metadata produced by HMI and AIA. In the DRMS, data are organized into *dataseries*, which is the name given to a sequence of related or similar items. These items are called *records*, which in turn are comprised of *keywords*, their values, and the associated data *segments*.

Examples of dataseries include **`hmi.V_45s`** (HMI Dopplergrams with a 45s cadence), **`aia_test.lev1`** (AIA Level 1 images), or even **`mdi.fd_M_96m_lev18`** (full-disk MDI Level 1.8 magnetograms with a 96m cadence). Most dataseries are essentially sequences of images in time, although this is not always necessarily the case. Additionally, dataseries are not limited to being image data, and may eventually include "derived" data products such as synoptic or diachronic charts, or cubes of remapped Dopplergrams for use in helioseismology.

Several points about the DRMS cataloging system are worth mentioning:

- A key aspect of the way the DRMS catalogs data is that *all records of the same dataseries have the same set of allowed keywords*. Of course, for different records these keywords will take on different values. Together, the list of records and list of keywords (and their values) form a database that is easily searched by DRMS.

- *The keywords and their values are stored separately from the data they describe*. This allows the metadata, i.e., the values of the keywords, to be accessed efficiently (and updated) without having to deal with large image files.

- In some dataseries, such as **`hmi.V_45s`**, records contain exactly one image segment (such as the Dopplergram image in this example). In others, records contain multiple segments of related data. It is important to note that, *regardless of how many segments a record contains, all segments of a record share many, if not most, of the same keyword values*. As an example, most of the records in the **`aia_test.lev1`** dataseries contain a data segment (the image) and another segment containing a record of pixels that were changed during the despiking process. Another example of dataseries that contain records with multiple segments are the various HMI vector magnetogram dataseries, for which the Level 1 HMI filtergram data are processed to yield various magnetic field parameters, including $|B_{\rm total}|$, inclination and azimuth angles, filling factors, and uncertainties. Each of these quantities will be stored in a different image segment, and yet because these segments will be part of the same record, they will share the same set of keyword data.

For more detailed information on the JSOC and DRMS, we refer the reader to the JSOC home page, and in particular the JSOC wiki.

4.2 The *lookdata* Tool

The *lookdata* tool is an online Javascript front end to the DRMS at Stanford, and can be used to browse names of dataseries, to get dataseries keyword lists, to examine metadata for sets of records, and to generate export requests. The *lookdata* tool is comprised of several tabs that, when progressed from left to right, allow the user to locate and export data from the JSOC. Throughout this process, there are several steps; bold labels in the subsections below correspond to bold labels in the tool. The tool itself provides some additional help, available by clicking on the yellow question marks that are visible from within the tool.

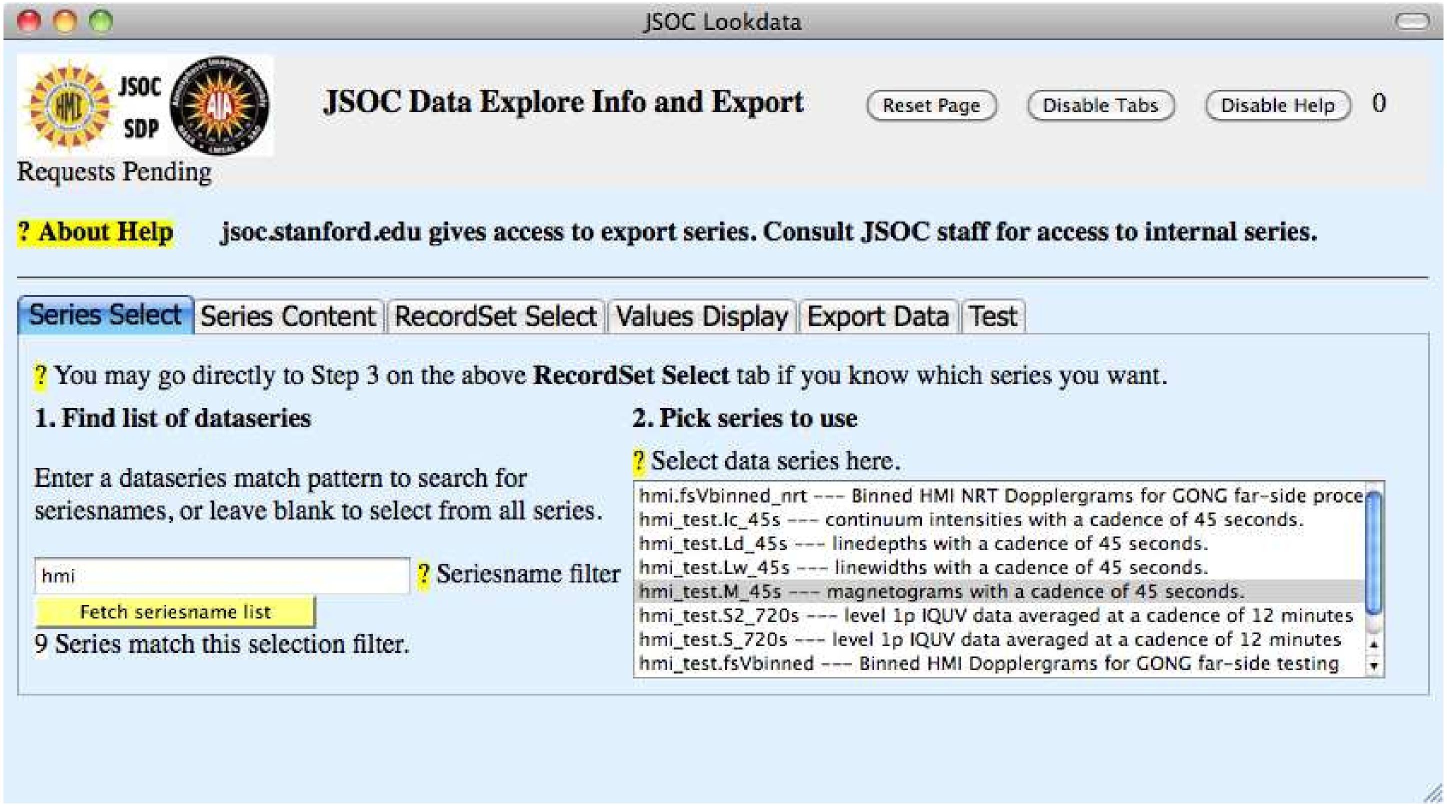


Figure 14: The "Series select" tab from the *lookdata* tool, showing Steps 1 and 2. Here, in Step 1 the user has searched for dataseries names containing the search string **`hmi`**, and has selected the **`hmi_test.M_45s`** dataseries in Step 2.

4.2.1 Getting Dataseries Names

The first step is to determine the name of the dataseries containing the data you wish to examine (and possibly export). To search for available dataseries, enter a search string in the "Seriesname filter" field (**Step 1**) under the "Series Select" tab on the *lookdata* webpage, as shown in Figure 14. Pressing the "Fetch seriesname list" button will return a list of all dataseries names containing that search string in the larger white box to the right of the "Series Select" field. In this white box (**Step 2**), click on the desired dataseries name from the list, and the *lookdata* tool will switch to the "RecordSet Select" tab.

> TECHNICAL NOTE: Entering **`NOT`** before the search string in the "Seriesname filter" field in Step 1 will return the list of dataseries names that exclude that search string.

> TECHNICAL NOTE: Prepending the caret (circumflex) character ˆ to the search string in the "Seriesname filter" field in Step 1 will return only those dataseries names that start with the search string.

4.2.2 Selecting Records

The next step is to filter the full set of data records in the chosen dataseries down to a more manageable number. This is done under the "RecordSet Select" tab, shown in Figure 15, where a DRMS query can be entered into the white box (**Step 3**). You will notice that the query box is already pre-populated with the dataseries name selected in Step 2. Typically, users will want to refine the query by selecting a particular range of dates and times. This is achieved by adding to the dataseries name a date or dates enclosed in square brackets. For informational purposes, near the top of the "RecordSet Select" tab, the dates of the earliest and most recent records in the dataseries are displayed. The following examples demonstrate this syntax (here the **hmi_test.M_45s** dataseries is used, but the syntax is generally applicable to most other dataseries):

- **hmi_test.M_45s[2010.07.25_13:30_TAI-2010.07.25_14:00_TAI]** — This query requests all HMI magnetograms for July 25, 2010 between 13:30 and 14:00 TAI. It will return 41 records (endpoints are inclusive).
- **hmi_test.M_45s[2010.07.25_13:30_TAI/30m]** — This query is almost identical to the above query, except here the forward-slash modifier is used to indicate the time range (the **/30m** means "over the next 30 minutes"). It will return 40 records (exclusive of the final endpoint).
- **hmi_test.M_45s[2010.07.25_13:30_TAI/30m@3m]** — This query is similar to the above queries, except that now the cadence is reduced from 45 seconds to 3 minutes (the **@3m** means "once per 3 minutes"). It will return 10 records.

Queries can be further refined by using keyword logic. A list of the full set of keywords for the selected dataseries can be found in the "Series Content" tab, located between the "Series Select" and "RecordSet Select" tabs. The following examples demonstrate queries using keyword logic (using the **aia_test.lev1** dataseries):

- **aia_test.lev1[2010.07.25_13:30/30m]** — As above, this query requests all AIA image records for July 25, 2010 for the half hour starting at 13:30. It will return 1177 records.
- **aia_test.lev1[2010.07.25_13:30/30m][? WAVELNTH=171 ?]** — This query now subselects the data from the above query to include only those image records for which the **WAVELNTH** keyword equals **171** (i.e., the 171Å channel of AIA). It will return 149 records (for one image about every 12 seconds).
- **aia_test.lev1[2010.07.25_13:30/30m][? WAVELNTH=171 or WAVELNTH=304 ?]** — This query returns those image records for which the **WAVELNTH** keyword equals either **171** or **304**. It returns 298 records.

After entering a query, press the "GetRecordCount" button, and the tool will return the record count. Optionally, entering a number n in the "Record Limit" field will limit the number of records displayed, with positive numbers limiting the display to the first n records and negative numbers limiting the display to the last n records. Pressing the "GetRecordCount" button also populates the "Select Keywords" and "Select Segments" boxes to the right of the query box with the respective lists of keywords and segments associated with the (now filtered) list of records, as shown in the example in Figure 15.

The user is now able to select which keywords to inspect (**Step 4**), which segments are desired (**Step 5**), and which links are desired (**Step 6**). Selecting ****ALL**** in these boxes will return all of each of these, but to speed things up users may want to select only a subset of keywords, segments, and/or links. Users can select individual elements in these lists by holding down the COMMAND or ALT key while clicking the left mouse button, and ranges of keywords can be selected by holding down the SHIFT key while clicking. Note that the "Select Links" box will often be empty.

Proceed to the next step by clicking on the "Fetch Keyword Values for RecordSet", at which time the *lookdata* tool will switch to the "Values Display" tab, as shown in the example in Figure 16. The table shown in the "Values Display" tab allows users to inspect the keyword values in the filtered set of records (**Step 7**). Refining the selected set of records is achieved by going back to the "RecordSet Select" tab (by clicking on the tab itself), adjusting the query, and repeating Steps 3 through 6.

> TECHNICAL NOTE: The elements in brackets following the dataseries name (and that do not involve keyword logic) in the queries shown here correspond to searches of *prime keys* for the dataseries. For nearly all dataseries of interest, **T_REC** or **T_OBS** will be a prime key, and this is what enables quick date and time searching without the use of more complicated keyword logic. One aspect of prime keys is that no two records can have the same set of prime key values (i.e., a record is uniquely specified by the values of its prime keys). The full set of prime keys for a selected dataseries is listed near the top of the "RecordSet Select" tab (see Figure 15).

TECHNICAL NOTE: Near the top of the "RecordSet Select" tab, the dates of the earliest and most recent records in the dataseries are displayed. However, several dataseries (such as **hmi_test.v_45s**, as shown in Figure 15) will list the first record as being in the year 4712 BC. This (nonsense) date is used to indicate records that have blank values of **T_REC**. The workaround for determining the date of the actual first record of a particular dataseries involves setting the date range of the query to be something like **[2010-01-01/365d]** (or whatever date range the first record is thought to be located in, for the dataseries of interest) and placing a **+1** in the "Record Limit" field below the query. Pressing the "GetRecordCount" button should yield one valid record (if not, try increasing the record limit), and pressing the "Fetch Keyword Values for RecordSet" button will then display the prime keys for this record (one of which is usually **T_REC** or **T_OBS**).

4.2.3 Exporting Data

If the user wishes to export the selected set of records, clicking on the "Export Data" tab will bring up **Step 8**. Clicking on the "Export" button will open a new window containing the "JSOC Export Data" form, the top section of which is shown in Figure 17. On the export form, the query and record count should already be pre-populated in the "RecordSet" and "Record Count" fields, respectively. The "Method" drop-down menu gives the user the option of choosing various export methods. For most remote users, we recommend the "url-tar" method. After selecting "url-tar", several new fields will appear, of which "Protocol" should be set to "FITS", and "Compression" set to "compress Rice". After pressing the "Submit Export Request" button, the user should make note of the ID tag that appears in the "RequestID" field below the button. Entering the ID tag in the darker-blue area at the bottom of the form will provide either an estimated time to completion or (if completed) a link to the archive containing the data. A completed export is illustrated in Figure 18.

TECHNICAL NOTE: Users who already know which records they want and who can construct the relevant query can save time by bypassing the *lookdata* tool altogether and entering the query directly onto a blank export form (using the link above).

4.3 The Cutout Service

The cutout service enables users to request AIA image sets (and soon HMI line-of-sight magnetograms and intensity images) to be prepared for downloading. It can either be accessed directly using the link above, or via the "Request data" link associated with events returned in iSolSearch queries (see Section 3.2). In the latter case, the data request form will be pre-populated with the spatiotemporal bounds of the originating event. In general, however, the service may be used for any other data request. In addition to the online form, there is also a SolarSoft interface to the cutout service, as demonstrated in Section 5.4.

As shown in Figure 19, the form has straightforward layout, containing input fields for spatial and temporal bounds of the cutout, as well as the wavelength channels desired. The spatial extent can be specified either by drawing a bounding box on the sample image or typing in coordinates directly into the appropriate parameter fields. Alternatively, instead of a bounding box, users can also specify "full-disk". Clicking on the "full-disk" link (underneath the sample image) will bring up radio buttons that allow the resulting image size to be specified, enabling the user to specify full-resolution images or images binned down by a factor of 2, 4, or 8.

In the last four fields of the request form users should input their name, e-mail address, a brief description of the data request, and the maximum number of image frames in the request. If the requested maximum number of frames is less than the actual number of frames requested, then the cutout service will keep the the requested duration but will subsample in time.

In the "Service" drop-down menu (underneath the sample image), users can request that either the "SSW Cutout Service" at LMSAL or the "JSOC Export Data" service be used. The "SSW Cutout Service", as the name implies, uses SSW routines to process the spatial and temporal extraction. The "JSOC Export Data" option automatically request an export from the JSOC as if the user had used the *lookdata* tool and then filled out the "JSOC Export Data" form (both described in Section 4.2). At present, "JSOC Export Data" option can only accommodate full-resolution and full-cadence data.

In either case, once the data set is generated, you will be sent an email with instructions on how to download the data. If the "SSW Cutout Service" option was chosen, the e-mail will contain a link to the service-request output page. There, the user can view a summary movie using the "JobWWW" link, or download the data by going into SSWIDL and using the command in the "Get_Data" field. This SSWIDL command causes the requested cutout to be downloaded into the local directory.

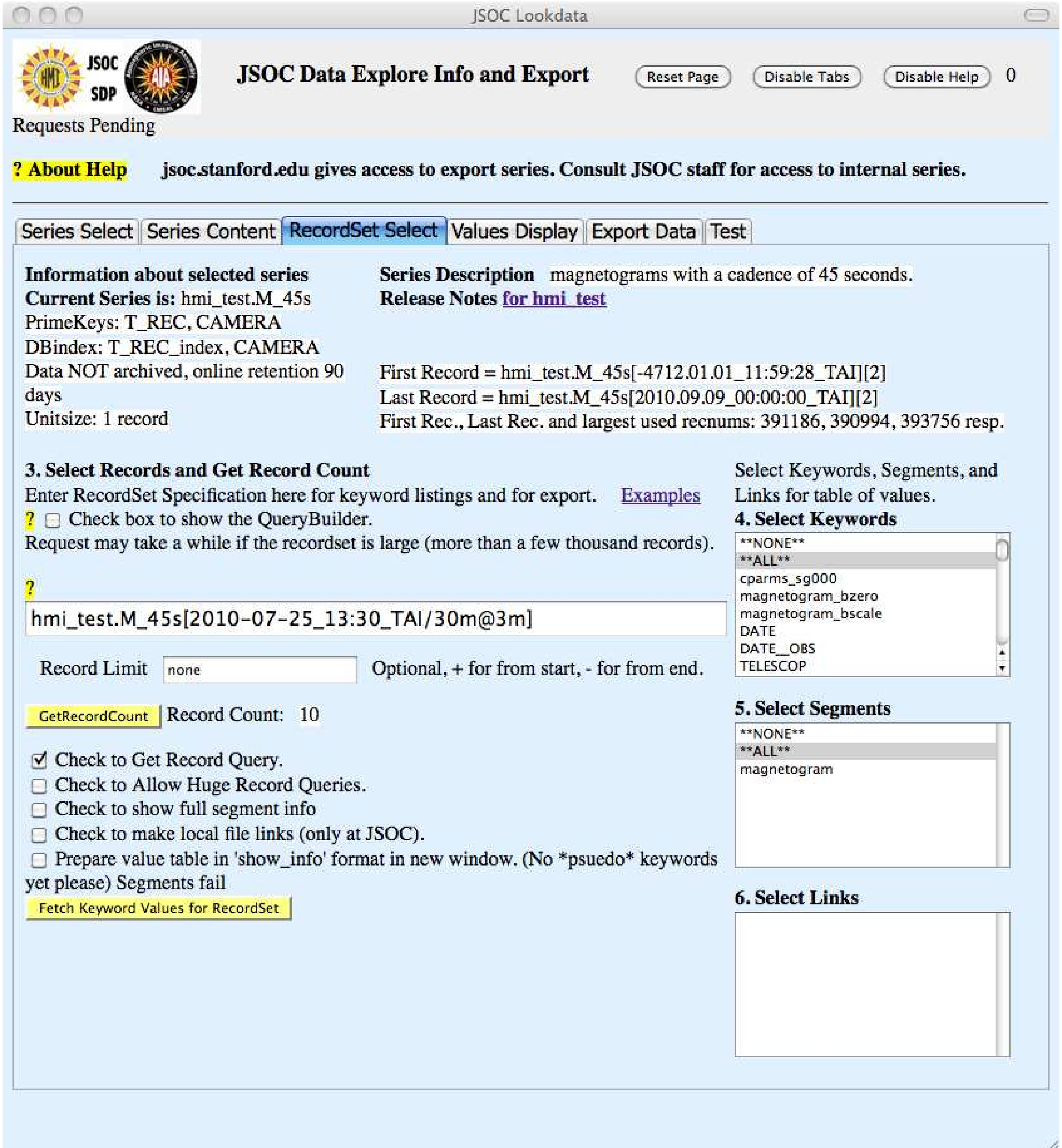


Figure 15: The "RecordSet Select" tab from the *lookdata* tool, showing Steps 3 through 6. Here, the user has entered the query **`hmi_test.M_45s[2010-07-25_13:30_TAI/30m@3m]`**, and the tool has indicated that 10 records match this query.

If, instead, the "JSOC Export Data" option was chosen, the e-mail will contain a link to the (temporary) location online at the JSOC from which the data can be downloaded. The data are presented as a listing of links, and as mentioned earlier, browser add-ons (such as DownThemAll! for Firefox) come in handy for downloading such lists of links in a few mouse-clicks.

If an error occurs in the request or if the request exceeds our currently tested capacity, it will stay in the queue and we shall try to process it in due time, or contact you on how to reformulate your request.

TECHNICAL NOTE: Choosing the "SSW Cutout Service" option creates the cutout from LMSAL's cache of Level-1.5 data, which is flat-fielded, despiked, and co-aligned. Choosing the "JSOC Export Data" option uses Level-1 data, which is flat-fielded and despiked, but not co-aligned).

TECHNICAL NOTE: The cutouts generated using the "SSW Cutout Service" option use data in the LMSAL cache, which contains only about the most recent two months. If you request older data, an error message will be sent. Similarly, the Level-1 data available at the JSOC extends for about three months, and if you request older data, nothing may happen until the JSOC implements the system to re-create Level-1 data from earlier data stored in the archive.

4.4 The Virtual Solar Observatory

The Virtual Solar Observatory (VSO) has been described as "one-stop shopping for (nearly) all of your solar physics needs". It allows users to query a multitude of data providers (e.g., SDO, Hinode, and other observatories) using standardized interfaces: either via the web or via SolarSoft. Here, in this section, we will demonstrate how to retrieve SDO data using the web-based interface, whereas the SolarSoft interface is demonstrated for SDO data in Section 5.5.

To use the web interface, navigate to the VSO web interface entry page, shown in Figure 20, where the user is presented with a list of attributes that will frame the subsequent VSO query. Note that "Time" is always selected. (If the date and time of interest are not already known, then the resources described in Section 3 of this guide will be of use.) For AIA or HMI, choosing the "Instrument / Source / Provider" and then "Instrument Only" options is a simple way to narrow down the source options. Choosing "Observable" (for HMI) or "Spectral Range" (for AIA) may also be of use when selecting SDO data. Pressing the "Generate VSO Search Form" button at the bottom of the entry page will display the search form.

On the search form, shown in Figure 21, the user enters a start and end time for the query, and checks off one or more instruments. Pressing the "Search" button will, after a few moments, return the results of the query.

At this point, the user selects which records they wish to download by marking the check-boxes in the left-most column. The "CheckBox Tools" section on the left-hand side may come in handy here, as it enables users to check or *un*check large swaths of records. The columns can also be sorted by checking off the boxes next to the column headings, and then clicking on the "Sort"/"Rearrange" buttons above the table.

The results page with some boxes checked is shown in Figure 22. Pressing the "Request Data" button, and selecting "URL-FILE_Rice" (for Rice-compressed FITS files), will bring up a list of links for download, as shown in Figure 23. Browser add-ons (such as DownThemAll! for Firefox) come in handy here for downloading all links in a few mouse-clicks.

Online help at any step is available either by clicking on the information icons, or by perusing the VSO FAQ webpage. The VSO blog may also be useful.

4.5 Viewing Compressed Images

Data from SDO is typically downloaded as Rice-compressed FITS files (in order to reduce the bandwidth needed for the download). Such files can be uncompressed and viewed using a number of freely-available image-manipulation software programs, including SDOImage DS9, and the **`imcopy`** program and Fpack package from the CFITSIO subroutine library. Additionally, the **`read_sdo.pro`** IDL routine, available via the **`ONTOLOGY`** SolarSoft package (see Section 5) will uncompress and read in Rice-compressed images into IDL.

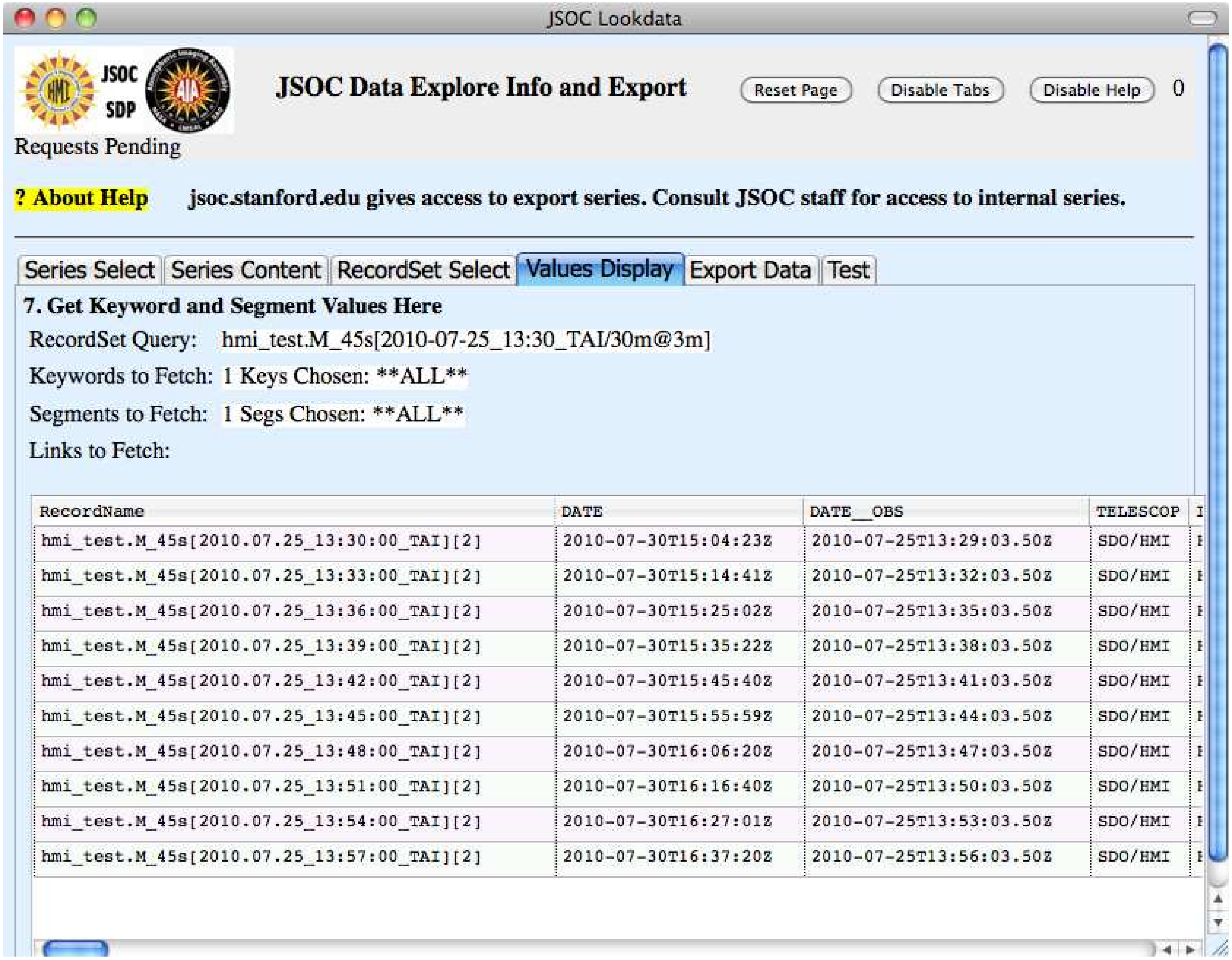


Figure 16: The "Values Display" tab from the *lookdata* tool, showing a listing of all keywords and their values for the 10 records matching the **`hmi_test.M_45s[2010-07-25_13:30_TAI/30m@3m]`** query illustrated in Figure 15.

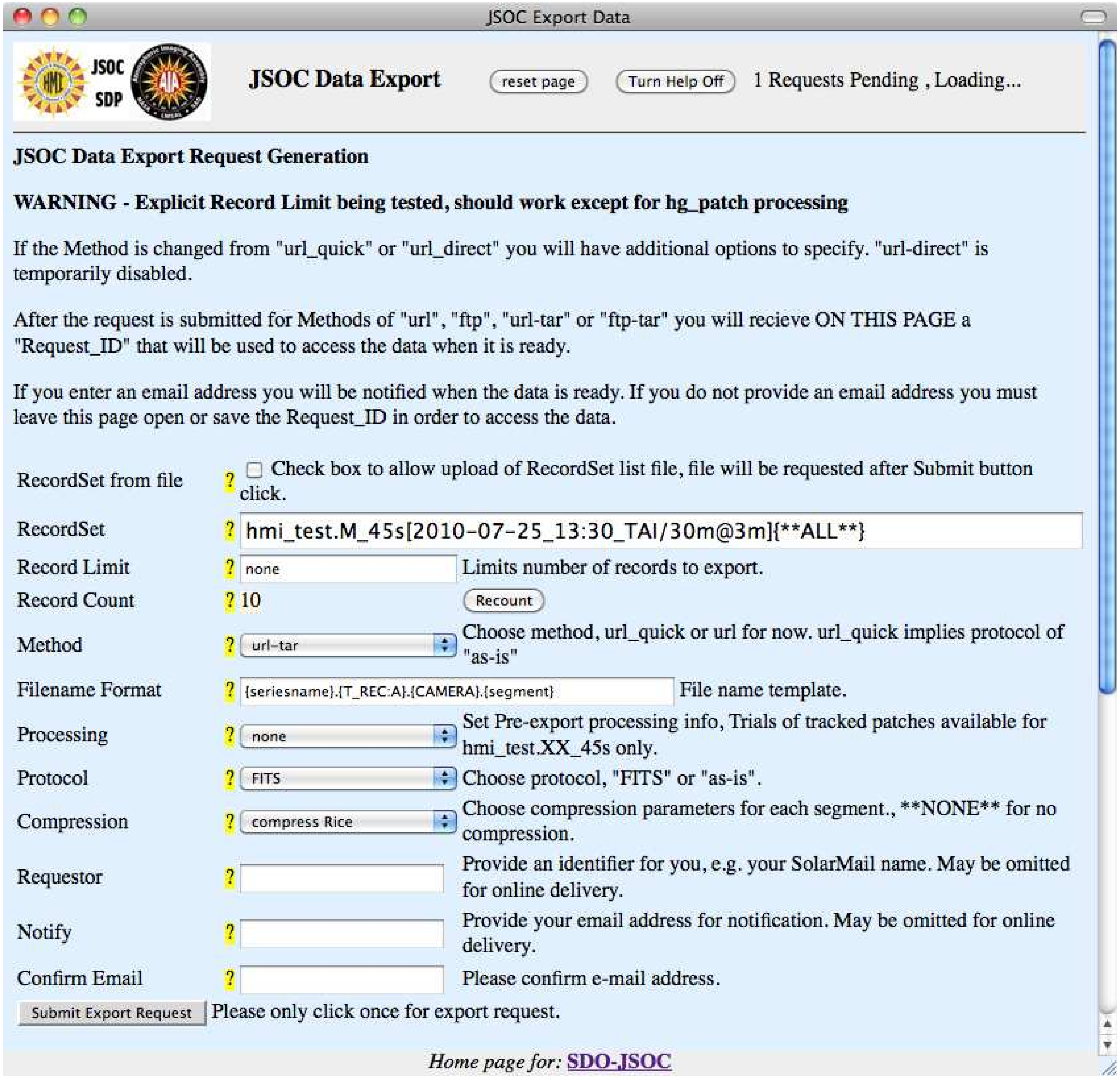
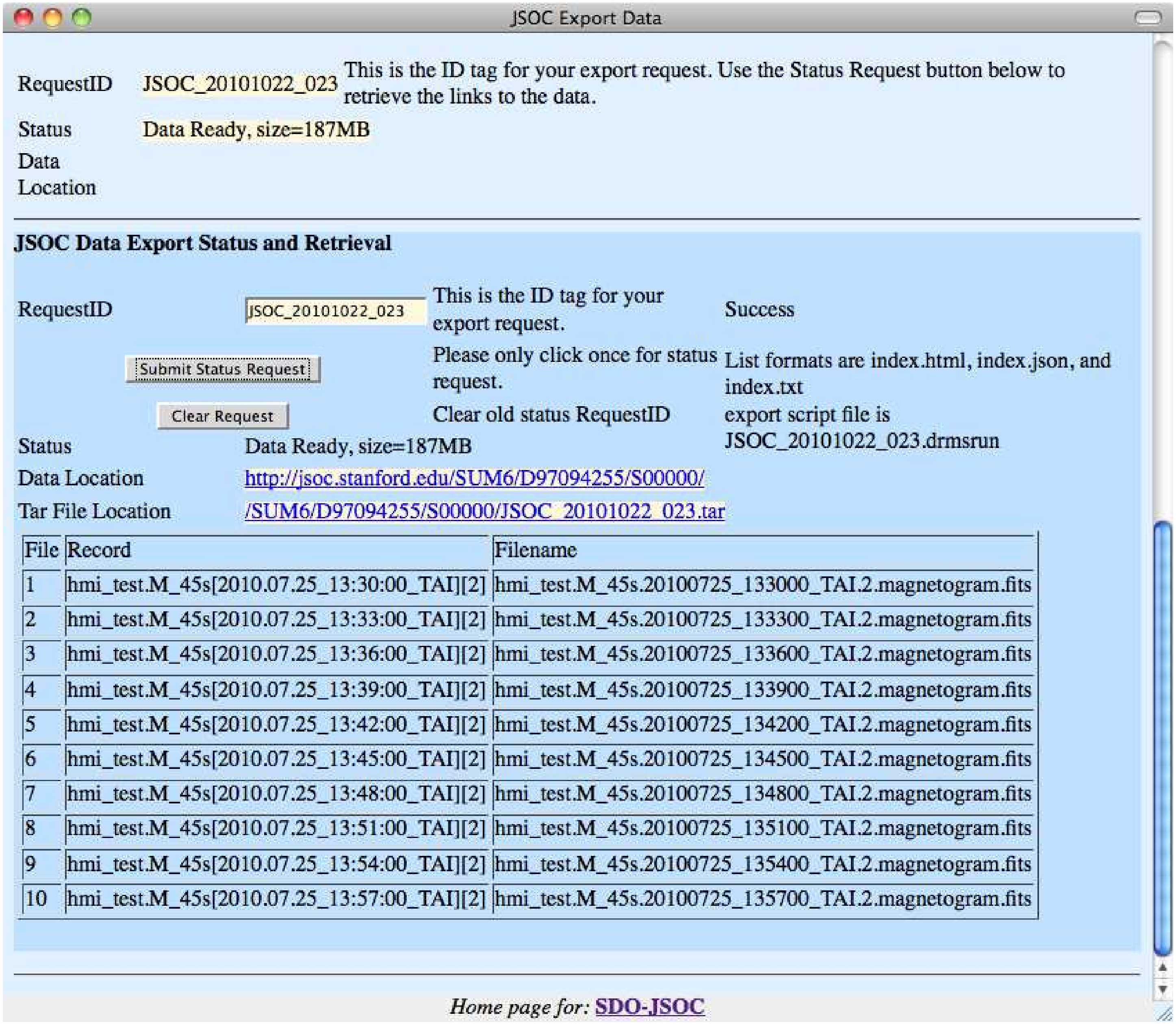


Figure 17: The top section of the "JSOC Export Data" form, illustrating how to export the 10 records matching the **`hmi_test.M_45s[2010-07-25_13:30_TAI/30m@3m]`** query illustrated in Figure 15. After pressing the "Submit Export Request" button and waiting for the export to process, the lower section will appear as in Figure 18.



JSOC Export Data

RequestID JSOC_20101022_023 This is the ID tag for your export request. Use the Status Request button below to retrieve the links to the data.

Status Data Ready, size=187MB

Data Location

JSOC Data Export Status and Retrieval

RequestID JSOC_20101022_023 This is the ID tag for your export request. Success

Submit Status Request Please only click once for status request. List formats are index.html, index.json, and index.txt

Clear Request Clear old status RequestID export script file is JSOC_20101022_023.drmsrun

Status Data Ready, size=187MB

Data Location http://jsoc.stanford.edu/SUM6/D97094255/S00000/

Tar File Location /SUM6/D97094255/S00000/JSOC_20101022_023.tar

File	Record	Filename
1	hmi_test.M_45s[2010.07.25_13:30:00_TAI][2]	hmi_test.M_45s.20100725_133000_TAI.2.magnetogram.fits
2	hmi_test.M_45s[2010.07.25_13:33:00_TAI][2]	hmi_test.M_45s.20100725_133300_TAI.2.magnetogram.fits
3	hmi_test.M_45s[2010.07.25_13:36:00_TAI][2]	hmi_test.M_45s.20100725_133600_TAI.2.magnetogram.fits
4	hmi_test.M_45s[2010.07.25_13:39:00_TAI][2]	hmi_test.M_45s.20100725_133900_TAI.2.magnetogram.fits
5	hmi_test.M_45s[2010.07.25_13:42:00_TAI][2]	hmi_test.M_45s.20100725_134200_TAI.2.magnetogram.fits
6	hmi_test.M_45s[2010.07.25_13:45:00_TAI][2]	hmi_test.M_45s.20100725_134500_TAI.2.magnetogram.fits
7	hmi_test.M_45s[2010.07.25_13:48:00_TAI][2]	hmi_test.M_45s.20100725_134800_TAI.2.magnetogram.fits
8	hmi_test.M_45s[2010.07.25_13:51:00_TAI][2]	hmi_test.M_45s.20100725_135100_TAI.2.magnetogram.fits
9	hmi_test.M_45s[2010.07.25_13:54:00_TAI][2]	hmi_test.M_45s.20100725_135400_TAI.2.magnetogram.fits
10	hmi_test.M_45s[2010.07.25_13:57:00_TAI][2]	hmi_test.M_45s.20100725_135700_TAI.2.magnetogram.fits

Home page for: **SDO-JSOC**

Figure 18: The lower section of "JSOC Export Data" form, illustrating the completed export requested in Figure 17.

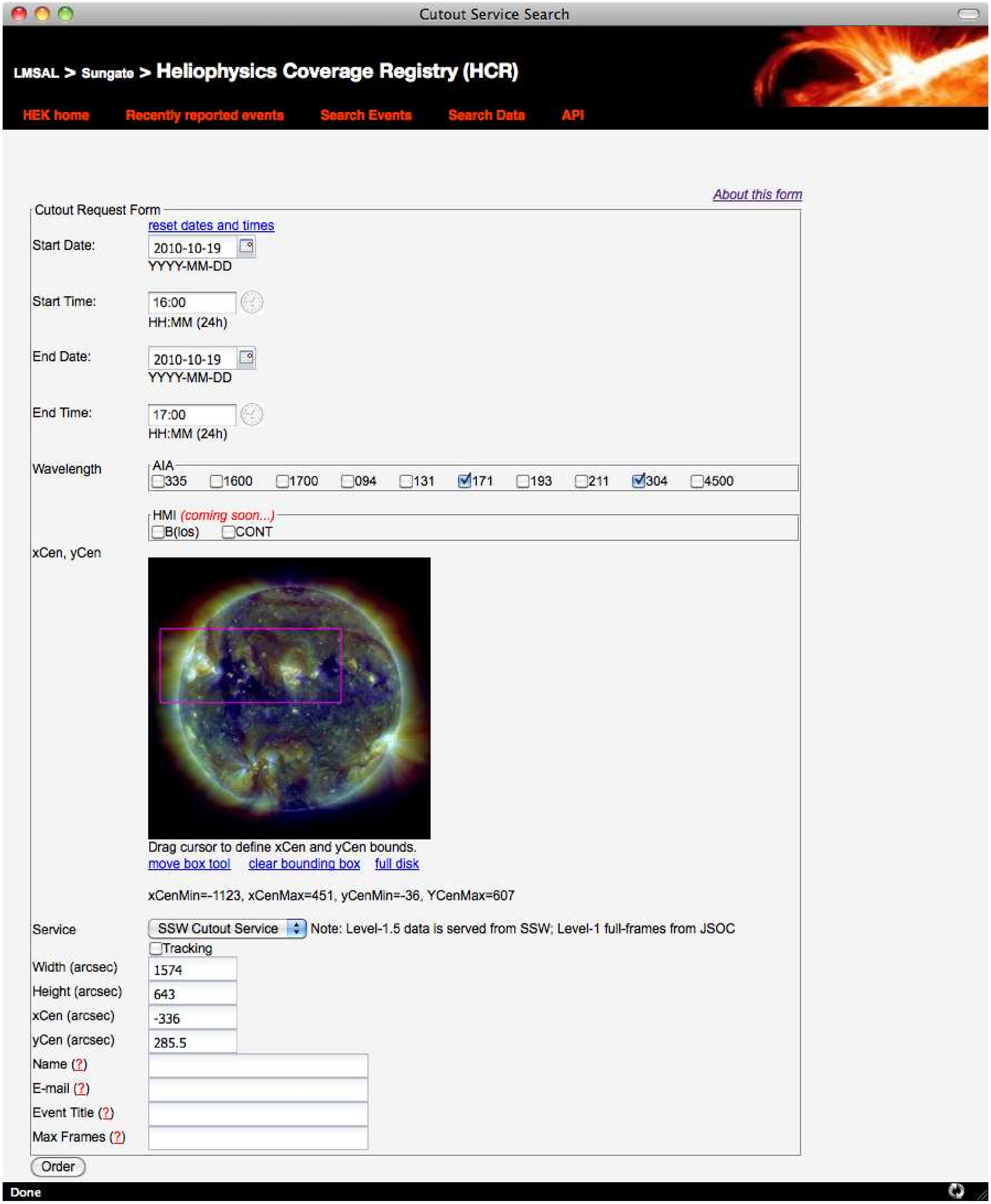


Figure 19: The cutout service request form, showing a request for data from the AIA 171Å and 304Å channels covering a localized region of the solar disk.

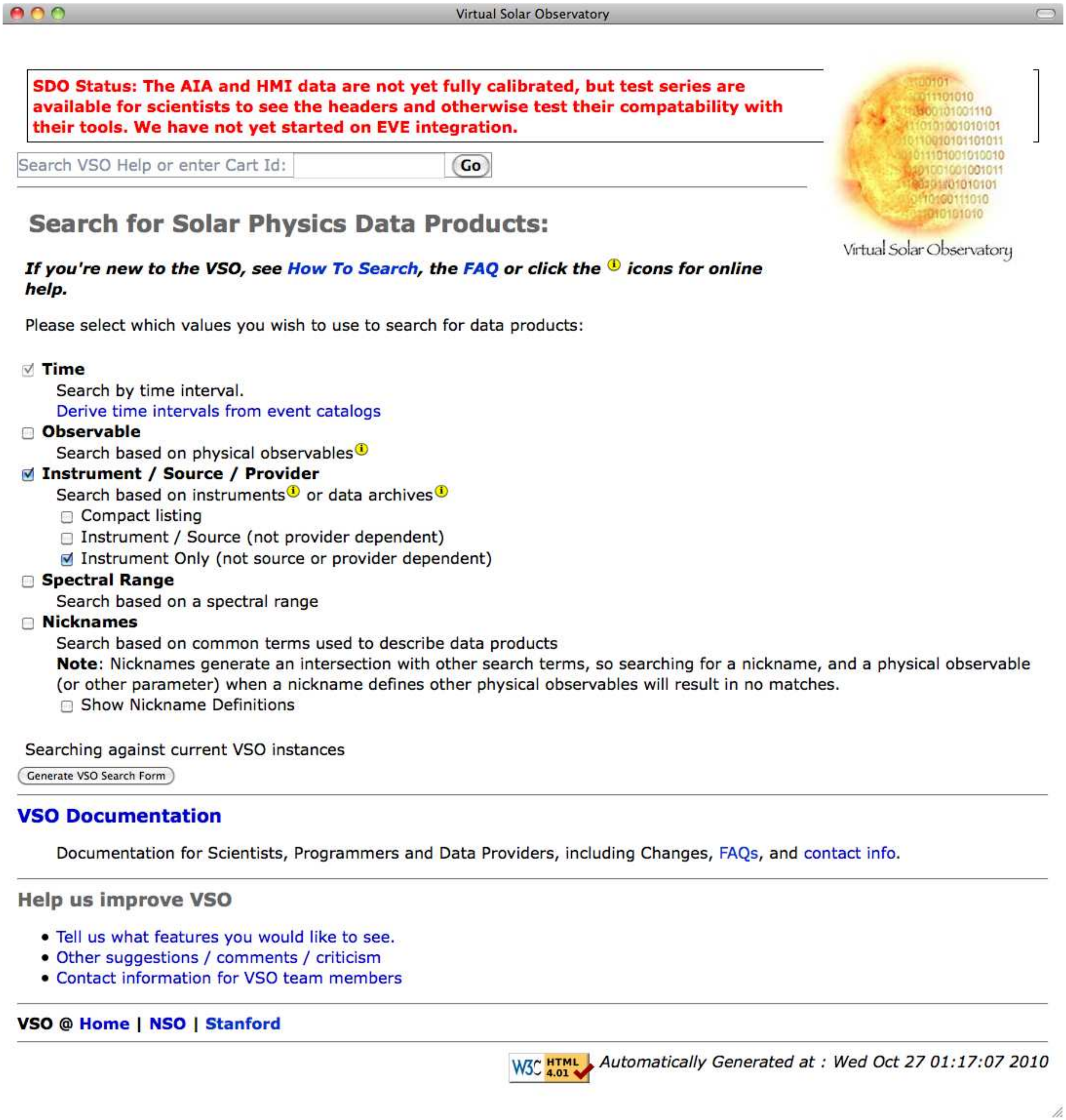


Figure 20: The VSO web interface entry page, with the "Instrument / Source / Provider" and "Instrument Only" options checked.

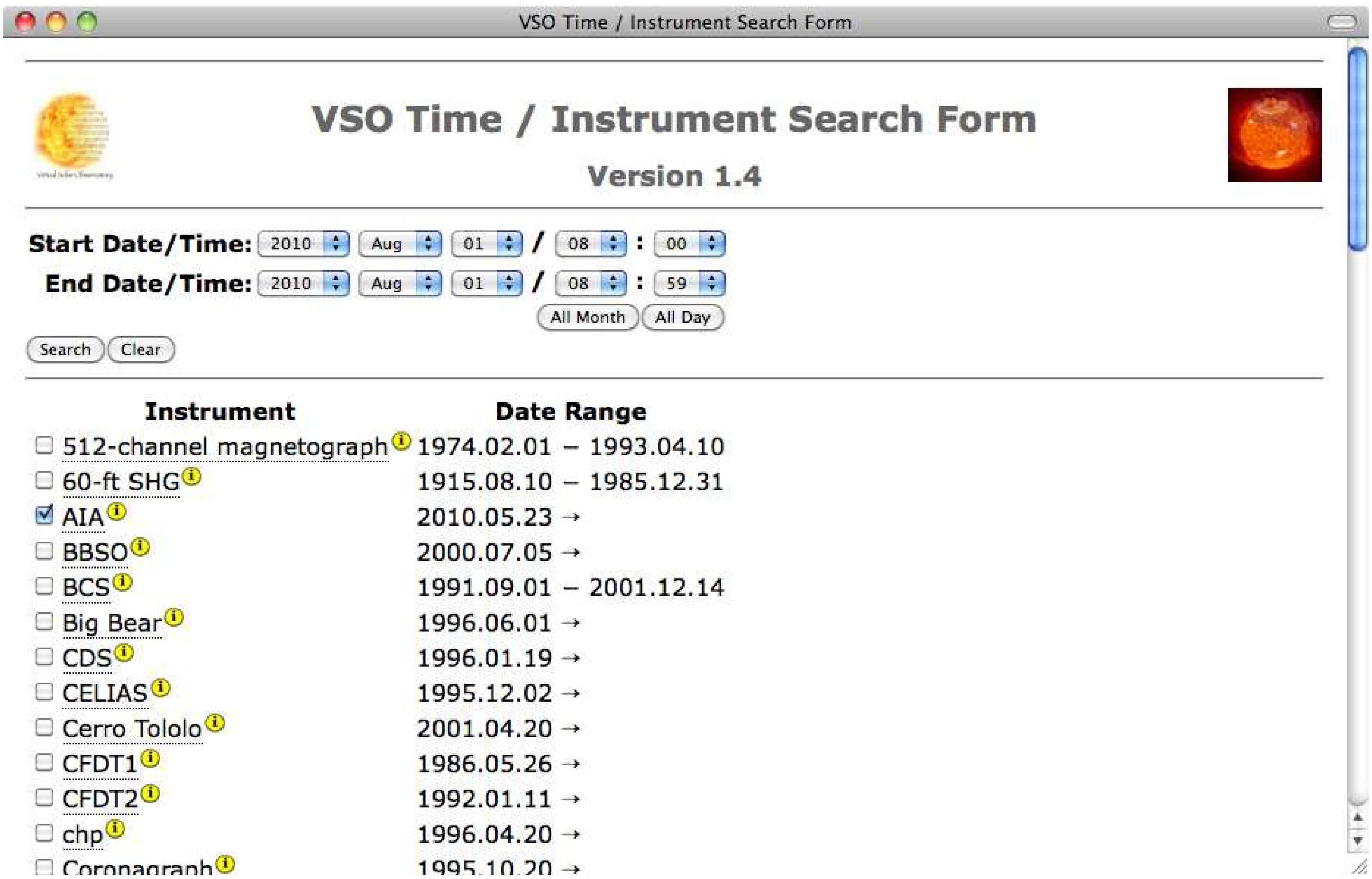


Figure 21: The VSO search form, with the user about to request one hour's worth of AIA data from 2010 Aug. 1.

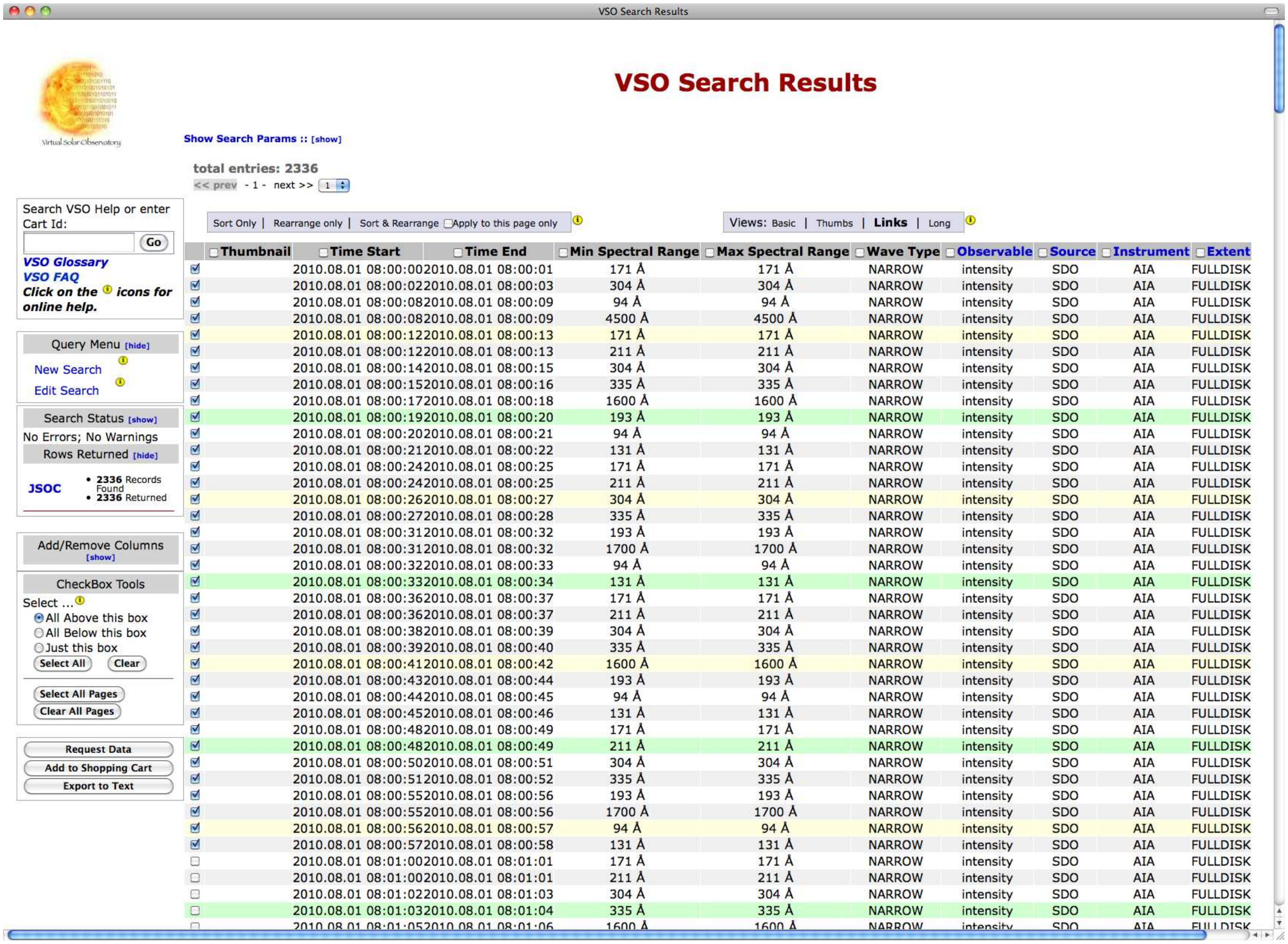


Figure 22: The results of the VSO query from Figure 21, showing the top portion of the listing of 2336 records. One minute's worth of AIA data have been selected for export.

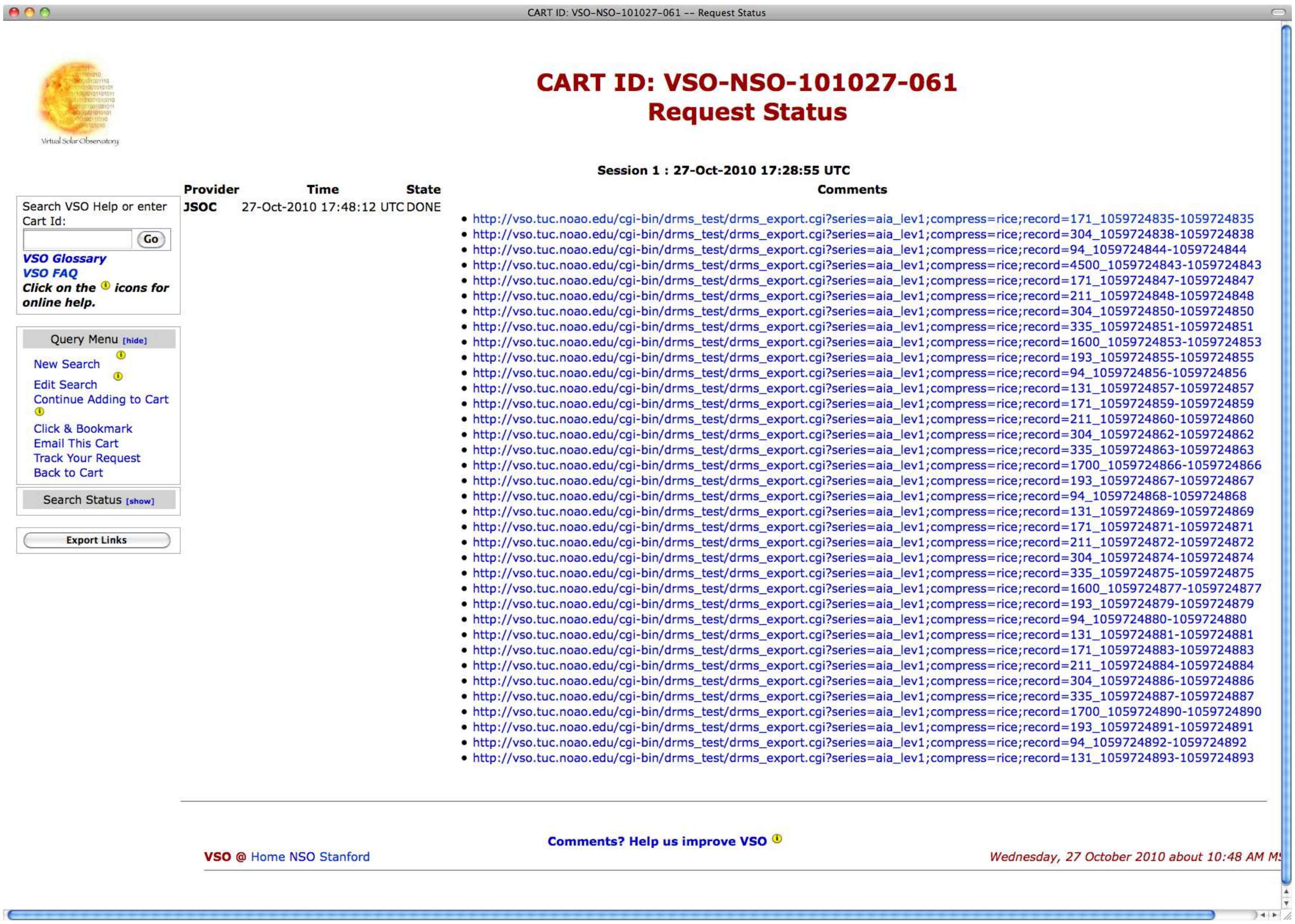


Figure 23: The VSO download page corresponding to the set of records checked off in Figure 22.

5 SolarSoft and SDO

The SolarSoftWare (SSW) system, or "SolarSoft" for short, is a software distribution system that originally stemmed from the need to coordinate solar mission data analysis across several institutions, going back as far as the Yohkoh mission (which launched in 1991). It is comprised of integrated software libraries, databases, and system utilities that provide a coordinated data analysis environment for solar physics research. It is largely hardware- and site-independent, with mirroring capabilities to keep installations at various sites consistent. Although most of the software distributed via SSW is in ITT Visual Information Solutions' Interactive Data Language (IDL) — the *lingua franca* of the solar physics community — SSW can easily accommodate software in other languages.

In the following sections, we will describe how to install or upgrade SSW for use with SDO data (Section 5.1), and the how to use SSW to interface with the HEK (Section 5.2) and to get HMI or AIA image data from the JSOC (Section 5.3). Also demonstrated are the SSW interfaces to the cutout service (Section 5.4), the VSO (Section 5.5), and other miscellaneous items (Section 5.6). Access to EVE data via SSW is currently under development.

5.1 Installing or Upgrading SolarSoft

5.1.1 Doing a Clean Install

To do a clean install of SSW, follow these steps:

- Navigate to the main SSW installation webpage. This is the webpage containing instructions for installing SSW on different operating systems.
- Next, click on the "SSW INSTALLATION FORM" link. The installation form allows the user to select the top-level path in which the software will reside, along with which packages are to be downloaded. For SDO-related software, select **AIA**, **HMI**, **EVE**, and **ONTOLOGY**. The **ONTOLOGY** package includes software that interacts with the JSOC, and the others contain software specific to the respective SDO instruments. If you think that you will be using the SSW interface to the VSO (see Section 5.5), then the **VSO** package should also be selected.
- After completing the installation form, click on the "Generate Installation Script" button near the bottom of the page. The website will then give you an indication of how much disk space your SSW installation will occupy, and provide you with a link to your personalized installation script or batch file.
- Next, save the installation script or batch file to disk by right-clicking on the link underneath the installation table. Run the script or batch file on your machine. On Windows systems, the batch file will be contained in an archive, which needs to be unzipped. Running the script or batch file will cause the SSW packages to be downloaded.

At this point, SSW and its constituent packages (should) have been downloaded to your system. On Unix and Unix-like systems, a few more steps are required for IDL to have access to this software, as described on the SSW setup webpage. First, it is necessary to define the **SSW** environment variable so that it points to the top-level path of the SSW tree:

```
setenv SSW [whatever you specified on the form]
```

SSW also needs to know which packages are available. This is achieved by defining the **SSW_INSTR** environment variable so that it includes the packages that were downloaded:

```
setenv SSW_INSTR "ontology aia hmi eve vso [etc.]"
```

Then, after sourcing the setup file,

```
source $SSW/gen/setup/setup.ssw [/loud]
```

one simply types **sswidl** (an alias that was defined in the setup file) at the Unix prompt to start up IDL. IDL combined with the SolarSoft libraries and databases is frequently referred to as "SSWIDL".

5.1.2 Adding Packages to an Existing Installation

If you already have an existing SSW installation and you simply want to add packages, then at a SSWIDL prompt issue a command of the form:

```
ssw_upgrade,/spawn,/loud,/newpackage1 [,/newpackage2] [,/...]
```

For example, adding the **`AIA`** and **`ONTOLOGY`** packages is achieved by entering the following (note that some systems also require the **`/passive_ftp`** flag):

```
ssw_upgrade,/spawn,/loud,/aia,/ontology [,/passive_ftp]
```

Afterward, be sure to add the new packages to your list of instruments in the **`SSW_INSTR`** environment variable. More information regarding SSW upgrades is available on the SSW upgrade webpage.

5.2 Querying the HER

Querying the HER (see Section 3.1) using SSW is achieved by using the IDL code available through the **`ONTOLOGY`** package. This package contains basic routines that interface with the HEK using the HEK API.

For SSWIDL queries to the HER, SolarSoft provides two routines: **`ssw_her_make_query.pro`**, which generates a string of parameters in used in the subsequent CGI query, and **`ssw_her_query.pro`**, which issues an input verbatim query to the HEK CGI interface and translates the output to an IDL structure vector of event matches. In essence, these routines provide the translation between the SSW-style index metadata structures and the XML-style **`<param>=<value>`** syntax used within the HEK.

The following example demonstrates how to run the query using **`ssw_her_make_query.pro`**:

```
;  copy and paste these commands into your SSWIDL session:
t0='2010-01-01'  &  t1='2010-12-31'
query = ssw_her_make_query(t0,t1,/fl)
print,query

IDL> t0='2010-01-01'  &  t1='2010-12-31'
IDL> query=ssw_her_make_query(t0,t1,/fl)
IDL> print,query
event_starttime=2010-01-01T00:00:00&event_endtime=2010-12-31T00:00:00&event_coordsys=helioprojec
```

Here, the **`/fl`** switch is set to indicate that we are querying for HER entries having the flare event class, since "**`fl`**" is the two-letter code for "flare" (as shown in the full list of event classes).

To execute the query, the **`ssw_her_query.pro`** routine is used:

```
;  copy and paste these commands into your SSWIDL session:
t0='2010-01-01'  &  t1='2010-12-31'
query=ssw_her_make_query(t0,t1,/fl)
events=ssw_her_query(query)
help,events,/str
help,events.fl

IDL> t0='2010-01-01'  &  t1='2010-12-31'
IDL> query=ssw_her_make_query(t0,t1,/fl)
IDL> events=ssw_her_query(query)
IDL> help,events
EVENTS          STRUCT    = -> <Anonymous> Array[1]
```

```
IDL> help,events,/str
** Structure <1c4dc0c8>, 1 tags, length=182400, data length=182400, refs=1:
   FL               STRUCT    -> <Anonymous> Array[100]
IDL> help,events.fl
<Expression>    STRUCT    = -> <Anonymous> Array[100]
IDL> help,events.fl,/str
** Structure <2838808>, 114 tags, length=1824, data length=1824, refs=2:
   EVENT_PIXELUNIT STRING    ''
   BOUND_CCSTARTC2 STRING    ''
   EVENT_DESCRIPTION
                   STRING    ''
   EVENT_NPIXELS   STRING    ''
   FRM_HUMANFLAG   STRING    'false'
   REVISION        STRING    '1'
   EVENT_EXPIRES   STRING    ''
   KB_ARCHIVID     STRING    'ivo://helio-informatics.org/FL_SECstandard_20100613_150757_2010010
  [...]
```

The output of the **ssw_her_query.pro** routine is an array of event structures (one mer match), in which are stored the metadata describing each event. The 100 matches reflect the current default match limit. This can be changed by setting the **RESULT_LIMIT** keyword in the call to **ssw_her_query.pro**.

Users can add filters that screen for, or screen out, events that satisfy a set of user-defined conditions by defining the **SEARCH_ARRAY** keyword in the call to **ssw_her_make_query** as a set of additional criteria. Using the same example as in the iSolSearch example of Section 3.2, suppose one wants to find all flares having GOES Class M2 or greater that occurred in the year 2010. The resulting query would filter the results based on the **FL_GOESCls** (flare GOES class) attribute, and would be constructed and executed as follows:

```
;  copy and paste these commands into your SSWIDL session:
t0='2010-01-01'  &  t1='2010-12-31'
query=ssw_her_make_query(t0,t1,/fl,search_array=['FL_GOESCls>=M2.0'])
events=ssw_her_query(query)
help,events,/str
```

```
IDL> t0='2010-01-01'  &  t1='2010-12-31'
IDL> query=ssw_her_make_query(t0,t1,/fl,search_array=['FL_GOESCls>=M2.0'])
IDL> events=ssw_her_query(query)
IDL> help,events,/str
** Structure <1c48a7c8>, 1 tags, length=32832, data length=32832, refs=1:
   FL               STRUCT    -> <Anonymous> Array[18]
```

Again, at the time of this writing, the query returned 18 events. Valid relational operators for the **SEARCH_ARRAY** keyword include **=**, **>=**, **<=**, **<>**, **>**, and **<**.

Within the SSWIDL environment, the **struct4event.pro** routine is available for determining the current list of attributes for each event class (basically by returning a blank event entry):

```
;  copy and paste these commands into your SSWIDL session:
flaretags=struct4event('FL')
help,flaretags.required,/str
help,flaretags.optional,/str
```

```
IDL> flaretags=struct4event('FL')
IDL> help,flaretags.required,/str
** Structure <480d208>, 31 tags, length=392, data length=388, refs=2:
   EVENT_TYPE       STRING    'FL: Flare'
   KB_ARCHIVDATE    STRING    'Reserved for KB archivist: KB entry date'
   KB_ARCHIVID      STRING    'Reserved for KB archivist: KB entry identifier'
   KB_ARCHIVIST     STRING    'Reserved for KB archivist: KB entry made by'
   KB_ARCHIVURL     STRING    'Reserved for KB archivist: URL to suppl. info.'
   EVENT_COORDSYS   STRING    'UTC-HPC-TOPO'
   EVENT_COORDUNIT STRING    'blank'
   EVENT_ENDTIME    STRING    '1492-10-12 00:00:00'
   EVENT_STARTTIME STRING    '1492-10-12 00:00:00'
   [...]
IDL> help,flaretags.optional,/str
** Structure <4802a08>, 55 tags, length=632, data length=586, refs=2:
   EVENT_PROBABILITY
                    FLOAT                Inf
   EVENT_EXPIRES    STRING    '1492-10-12 00:00:00'
   EVENT_COORD3     FLOAT                Inf
   EVENT_MAPURL     STRING    'blank'
   EVENT_MASKURL    STRING    'blank'
   EVENT_CLIPPEDSPATIAL
                    STRING    'blank'
   EVENT_CLIPPEDTEMPORAL
                    STRING    'blank'
   EVENT_TESTFLAG   STRING    'blank'
   [...]
```

5.3 Retrieving Data from the JSOC

The SSWIDL routines that interface with the JSOC (see Section 4.1) all have the form **ssw_jsoc*.pro**, and are located in the **ONTOLOGY** package in the SSW installation. These routines have many more parameters and keywords than are outlined here, and we urge users to look at the source-code preamble for more detailed lists of options. The source code can be viewed either by using the **xdoc.pro** routine from within SSWIDL, by using the SSW code search webpage to search for a routine of interest, or by directly viewing the source code once the local path is known (say, after using the SSWIDL routine **which.pro**, which returns the path of any routine in a user's IDL path). Additionally, readers should be familiar with terms describing the JSOC data management system, such as *dataseries*, *record*, and *segment*, as described in Section 4.1.

There are three basic steps that enable users to download data from the JSOC directly into SSWIDL, as discussed in the following subsections. The steps illustrated here are directly analogous to the steps needed to use the *lookdata* tool demonstrated in Section 4.2.

5.3.1 Getting Dataseries Names

The first step is to get the JSOC dataseries name using the **FILTER** keyword of **ssw_jsoc.pro**:

```
result=ssw_jsoc(filter='[filter string]')
```

This function returns a structure, with each element in the structure corresponding to a different dataseries. All of the dataseries names contain the substring provided in the **FILTER** keyword on input. The **NAMES** tag of this structure contains the dataseries name, the list of prime keys, and a one-line description of each dataseries. For example, to find all JSOC dataseries containing the substring **aia**, enter the following commands into SSWIDL:

```
;  copy and paste these commands into your SSWIDL session:
result=ssw_jsoc(filter='aia')
help,result,/str
help,result.names(0),/str
```

```
IDL> result=ssw_jsoc(filter='aia')
IDL> help,result,/str
** Structure <1438fc8>, 3 tags, length=208, data length=196, refs=1:
   STATUS          INT               0
   NAMES           STRUCT    -> <Anonymous> Array[4]
   N               INT               4
IDL> help,result.names(0),/str
** Structure <1436c48>, 3 tags, length=48, data length=48, refs=2:
   NAME            STRING    'aia_test.lev1'
   PRIMEKEYS       STRING    'T_OBSFSN'
   NOTE            STRING    'AIA level 1 test'
```

Note that this result should mimic the output displayed in Step 2 of the *lookdata* tool (see Section 4.2.1).

> TECHNICAL NOTE: Prepending the caret (circumflex) character ˆ to the search substring in the **FILTER** keyword will return only those dataseries whose names that start with the search string.

5.3.2 Selecting Records

Once the dataseries name is known, the next step is to get a listing of available records from the JSOC. This is achieved using the **ssw_jsoc_time2data.pro** routine:

```
ssw_jsoc_time2data,t0,t1,index,ds='[dataseries name]'
```

If images from the specified dataseries exist in the date/time range specified by the **T0** and **T1** arguments, the procedure will return (in the **INDEX** variable) a listing of these image headers. The structure tags mimic the list of keywords shown by the *lookdata* tool in Step 4 (see Section 4.2.2). For example:

```
;  copy and paste these commands into your SSWIDL session:
ssw_jsoc_time2data,'01-nov-2010 01:00','01-nov-2010 02:00',index,ds='aia_test.lev1'
help,index
help,index,/str
```

```
IDL> ssw_jsoc_time2data,'01-nov-2010 01:00','01-nov-2010 02:00',index,ds='aia_test.lev1'
 ----------------------------------
| /SERIES_STRUCT, ds=aia_test.lev1 |
 ----------------------------------
 -------------------------------------
| Negative time in Interval structure |
 -------------------------------------
 ----------------------------------------------------------------
| ssw_jsoc_time2query output: aia_test.lev1[2010-11-01T01:00:00Z/60m] |
```

```
 --------------------------------------------------------------------
 ------------------------------
| selecting segment> image_lev1 |
 ------------------------------
 -----------------------------------------------------------------
| /rs_list, ds=aia_test.lev1[2010-11-01T01:00:00Z/60m]{image_lev1} |
 -----------------------------------------------------------------
IDL> help,index
INDEX           STRUCT    = -> <Anonymous> Array[2135]
IDL> help,index,/str
** Structure <19e2408>, 165 tags, length=1024, data length=975, refs=1:
   BLD_VERS        STRING    'V5R10'
   T_OBS_STEP      DOUBLE           1.0000000
   T_OBS_EPOCH     STRING    '1977.01.01_00:00:00_TAI'
   T_OBS_ROUND     LONG                    1
   ORIGIN          STRING    'SDO\/JSOC-SDP'
   DATE            STRING    '2010-11-03T00:59:54Z'
   TELESCOP        STRING    'SDO/AIA'
   INSTRUME        STRING    'AIA_3'
   DATE__OBS       STRING    '2010-11-01T01:00:00.34Z'
   T_OBS           STRING    '2010-11-01T01:00:01.34Z'
   [...]
```

Note that in this one hour, there are 2,135 AIA images, each of which is about 12MB prior to being uncompressed.

> TECHNICAL NOTE: The "Negative time in Interval structure" message from **ssw_jsoc_time2data.pro** arises from the fact that some records in the dataseries contain anomalous values for the relevant observation date keyword (such as values in the year 4712 BC — a "negative time" value). Such records lie outside the date range of almost all useful queries, and consequently the results produced by **ssw_jsoc_time2data.pro** are typically not affected in an adverse manner. See the related technical note in Section 4.2.2 for more information.

5.3.3 Exporting Data

Downloading the image data (instead of just the headers, as in Item 2 above), is achieved using the same **ssw_jsoc_time2data.pro** procedure but with an additional argument:

```
ssw_jsoc_time2data,t0,t1,index,data,ds='[dataseries name]'
```

On output, the newly added **DATA** argument will contain the image data. What is actually being downloaded are Rice-compressed images, but **ssw_jsoc_time2data.pro** automatically uncompresses the data using **read_sdo.pro** (as mentioned in Section 4.5) prior to exiting the procedure. There will be a one-to-one correspondence between the images in the **DATA** array and the header information in the **INDEX** array.

The **ssw_jsoc_time2data.pro** procedure accepts many keywords. For AIA data, the **WAVES** keyword can be used to select particular wavelength channels of interest. The **MAX_FILES** keyword limits the number of files that are downloaded. *It is often prudent to use the **MAX_FILES** keyword unless you know how much data you are requesting.* Each full-resolution, full-disk AIA or HMI image is approximately 12MB when compressed, expanding to 64MB when uncompressed.

To keep your working directory clean, make use of both the **COMP_DELETE** switch, which will delete the compressed files after they have been downloaded (they are stored in directories whose names begin with **SUM**, followed by a number), and/or the **UNCOMP_DELETE** switch, which will delete the uncompressed files after they have been read into IDL.

Here is an example illustrating how one would download a sample of 4 AIA images from one minute of mission time:

```
;  copy and paste these commands into your SSWIDL session:
t0='01-nov-2010 01:00'
t1='01-nov-2010 01:01'
ssw_jsoc_time2data,t0,t1,index,data,ds='aia_test.lev1',waves='171,304',$
  max_files=4,/uncomp_delete,/comp_delete
```

```
IDL> t0='01-nov-2010 01:00'
IDL> t1='01-nov-2010 01:01'
IDL> ssw_jsoc_time2data,t0,t1,index,data,ds='aia_test.lev1',waves='171,304',$
  max_files=4,/uncomp_delete,/comp_delete
 ----------------------------------
| /SERIES_STRUCT, ds=aia_test.lev1 |
 ----------------------------------
 -------------------------------------
| Negative time in Interval structure |
 -------------------------------------
 ---------------------------------------------------------------------
| ssw_jsoc_time2query output: aia_test.lev1[2010-11-01T01:00:00Z/60s] |
 ---------------------------------------------------------------------
 -------------------------------
| selecting segment> image_lev1 |
 -------------------------------
 ------------------------------------------------------------------
| /rs_list, ds=aia_test.lev1[2010-11-01T01:00:00Z/60s]{image_lev1} |
 ------------------------------------------------------------------
FILTER>> ssnew=where(gt_tagval(index(ss),/wavelnth) eq 171 OR gt_tagval(index(ss),/wavelnth) eq 3
FILTER>> ssnew=where(strmatch(gt_tagval(index(ss),/img_type),'LIGHT',fold_case=fold_case) eq 1,ss
 -----------------------------------------------------------------
| /export, ds=aia_test.lev1[2010-11-01T01:00:00Z/60s]{image_lev1} |
 -----------------------------------------------------------------
STR             UNDEFINED = <Undefined>
Use: status = tag_exist(str, tag)
str = structure variable
tag = string variable
 -------------------
| segment subselect |
 -------------------
 --------------------------
| Throtteling to MAX_FILES |
 --------------------------
 ---------------------------------------------------------------------------
| Getting> http://jsoc.stanford.edu//SUM9/D101248958/S00000/image_lev1.fits |
 ---------------------------------------------------------------------------
 ----------------------------------------------------------------------------
| Getting> http://jsoc.stanford.edu//SUM10/D101248961/S00000/image_lev1.fits |
```

```
 -----------------------------------------------------------------------------
 -----------------------------------------------------------------------------
| Getting> http://jsoc.stanford.edu//SUM11/D101248962/S00000/image_lev1.fits |
 -----------------------------------------------------------------------------
 -----------------------------------------------------------------------------
| Getting> http://jsoc.stanford.edu//SUM14/D101248965/S00000/image_lev1.fits |
 -----------------------------------------------------------------------------
 ----------------------------------
| Tile compressed data handling... |
 ----------------------------------
 -------------------------------------------
| Uncompressing to> /Users/derosa/temp/test |
 -------------------------------------------
 -----------------------------------------------------------------------------------------------
| /Users/derosa/temp/test//SUM9/D101248958/S00000/image_lev1.fits -> /Users/derosa/temp/test/AIA2
 -----------------------------------------------------------------------------------------------
 ---------------------------------------------
| struct2fitshead - using procedure: sxaddpar |
 ---------------------------------------------
 -----------------------------------------------------------------------------------------------
| /Users/derosa/temp/test//SUM10/D101248961/S00000/image_lev1.fits -> /Users/derosa/temp/test/AI/
 -----------------------------------------------------------------------------------------------
 -----------------------------------------------------------------------------------------------
| /Users/derosa/temp/test//SUM11/D101248962/S00000/image_lev1.fits -> /Users/derosa/temp/test/AI/
 -----------------------------------------------------------------------------------------------
 -----------------------------------------------------------------------------------------------
| /Users/derosa/temp/test//SUM14/D101248965/S00000/image_lev1.fits -> /Users/derosa/temp/test/AI/
 -----------------------------------------------------------------------------------------------
 -------------------------------------------
| Removing uncompressed versions on request |
 -------------------------------------------
 -------------------------------------
| Removing compressed files on demand |
 -------------------------------------
```

After downloading and uncompressing all of the requested data, the standardized AIA color tables can be loaded via the **aia_lct.pro** routine (see Section 5.6.1), and a (downsampled) image displayed on screen as follows:

```
;  copy and paste these commands into your SSWIDL session:
aia_lct,wave=171,/load
window,0,xsiz=512,ysiz=512
tvscl,rebin(data(*,*,1),[512,512])
```

TECHNICAL NOTE: If header information appears to exist, but upon requesting the image data the **ssw_jsoc_time2data.pro** routine produces an error message stating "There are no files in this RecordSet", this usually indicates that the requested image files exist but have expired from the LMSAL cache and are thus not presently available. Currently, the online cache is large enough for only about two months of AIA data. Once image files roll off the cache, they are not available using **ssw_jsoc_time2data.pro**. As of Dec. 2010, this situation is in the process of being rectified by way of expanding the size of the cache so as to accommodate all Level 1 AIA image data for the full duration of the mission.

5.4 Requesting a Cutout using SSW

The online cutout service described in Section 4.3 can also be accessed from SSWIDL using the **ssw_cutout_service.pro** procedure:

```
ssw_cutout_service,t0,t1,ref_helio='[centroid coordinates]',fovx=[width],fovy=[height],$
  waves=[wavelengths],instr='[instrument]',max_frames=[maximum frames],email='xxx@yyy.zzz'
```

With this routine, the user specifies IDL keywords corresponding to the same parameters as for the online form. For example, if a user wanted a cutout from the 171Å and 304Å channels of AIA, the request might look similar to this sequence:

```
t0='01-jul-2010 01:00'
t1='01-jul-2010 03:00'
ssw_cutout_service,t0,t1,fovx=256,fovy=256,ref_helio='N35W20',$
  waves='171,304',max_frames=30, instrument='aia',email='xxx@yyy.zzz'
```

Alternatively, one can use the cutout service to downsample full-disk images to a more manageable size via use of the **/FULL_DISK** switch (which supersedes the **REF_HELIO** keyword), with a query similar to the following:

```
t0='01-jul-2010 01:00'
t1='01-jul-2010 03:00'
ssw_cutout_service,t0,t1,fovx=256,fovy=256,/full_disk,$
  waves='171,304',max_frames=30, instrument='aia',email='xxx@yyy.zzz'
```

When a cutout request is finished, an e-mail containing a link to the data will be sent to the e-mail address provided in the **EMAIL** keyword.

TECHNICAL NOTE: If the webpage contained in the e-mail message indicates a status of "No Data Found", this may mean that the requested image files exist, but are not available online. At present, the online cache is large enough for only about two months of AIA data. Once image files roll off the cache, they are not available using the cutout service. As of Dec. 2010, this situation is in the process of being rectified by way of expanding the size of the cache so as to accommodate all Level 1 AIA image data for the full duration of the mission.

5.5 Using the SSW Interface to the VSO

Queries to the VSO can either be submitted online (see Section 4.4) or through SSWIDL. Using SSWIDL, the basic sequence of query→select→download described earlier is implemented using IDL routines in the **VSO** package from SSW.

To run a VSO query, use the **vso_search.pro** routine. The routine has several keywords that may be useful for constructing queries, including **PROVIDER**, **INSTRUMENT**, **PHYSOBS**, and **WAVE**, that match the checkboxes on the online VSO entry page (see Figure 20 of Section 4.4). For example, duplicating in SSWIDL the query of Figure 21 for one hour's worth of AIA data can be achieved as follows:

```
;  copy and paste these commands into your SSWIDL session:
result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia')
help,result
help,result,/str
```

```
IDL> result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia')
Records Returned : JSOC : 2336/2336
Records Returned : JSOC : 0/0
IDL> help,result
RESULT          STRUCT    = -> VSORECORD Array[2336]
IDL> help,result,/str
** Structure VSORECORD, 13 tags, length=256, data length=252:
   TIME            STRUCT    -> VSOTIME Array[1]
   EXTENT          STRUCT    -> VSOEXTENT Array[1]
   WAVE            STRUCT    -> VSOWAVE Array[1]
   DETECTOR        STRING    ''
   INSTRUMENT      STRING    'AIA'
   SOURCE          STRING    'SDO'
   PROVIDER        STRING    'JSOC'
   INFO            STRING    'AIA level 1,  4096x4096'
   PHYSOBS         STRING    'intensity'
   FILEID          STRING    'aia_lev1:171:1059724835'
   SIZE            FLOAT           66200.0
   URL             STRING    ''
   GETINFO         STRING    ''
```

This search returns 2336 records corresponding to images from all of the AIA channels. Some details about each record is stored in the array of structures returned by the **vso_search.pro** routine, with each element in the array corresponding to a valid record. To narrow this list down by wavelength, one adds the **WAVE** keyword:

```
;  copy and paste these commands into your SSWIDL session:
result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia',wave=171)
help,result
```

```
IDL> result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia',wave=171)
Records Returned : JSOC : 293/293
Records Returned : JSOC : 0/0
IDL> help,result
RESULT          STRUCT    = -> VSORECORD Array[293]
```

This query results in a more manageable 293 records. Additionally, this query can be further refined by specifying a sampling cadence (in seconds) in the **SAMPLE** keyword:

```
;  copy and paste these commands into your SSWIDL session:
result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia',wave=171,sample=60)
help,result
```

```
IDL> result=vso_search('2010-08-01 8:00','2010-08-01 8:59',inst='aia',wave=171,sample=60)
Records Returned : JSOC : 60/60
Records Returned : JSOC : 0/0
IDL> help,result
RESULT          STRUCT    = -> VSORECORD Array[60]
```

The above query returns 60 AIA images from the 171Å channel from the requested hour.

To retrieve records, one simply passes an array of structures returned from **vso_search.pro** to the **vso_get.pro** routine:

```
;  copy and paste these commands into your SSWIDL session:
result=vso_search('2010-08-01 8:00','2010-08-01 8:01',inst='aia',wave=171,sample=60)
log=vso_get(result,out_dir='data',/rice)
```

```
IDL> log=vso_get(result,out_dir='data',/rice)
% VSO_GET: This will download 2 file(s)
2 : http://vso.tuc.noao.edu/cgi-bin/drms_test/drms_export.cgi?series=aia_lev1;record=171_10597248
% RDWRT_BUFF: Please wait. Downloading...
% File: /cgi-bin/drms_test/drms_export.cgi?series=aia_lev1;record=171_1059724895-1059724895
% Size: 67124160 bytes
% From: vso.tuc.noao.edu
% To: data
% HTTP::COPY: 67124160 bytes of 67124160 total bytes copied in    65.23 seconds
% HTTP::COPY: Wrote 67124160 bytes to file data/aia_test.lev1.171A_2010-08-01T08_01_00.34Z.image_
1 : http://vso.tuc.noao.edu/cgi-bin/drms_test/drms_export.cgi?series=aia_lev1;record=171_10597248
% RDWRT_BUFF: Please wait. Downloading...
% File: /cgi-bin/drms_test/drms_export.cgi?series=aia_lev1;record=171_1059724835-1059724835
% Size: 67124160 bytes
% From: vso.tuc.noao.edu
% To: data
% HTTP::COPY: 67124160 bytes of 67124160 total bytes copied in    66.43 seconds
% HTTP::COPY: Wrote 67124160 bytes to file data/aia_test.lev1.171A_2010-08-01T08_00_00.34Z.image_
Downloading completed
```

In this example, the **OUT_DIR** keyword is used to specify the output directory, and the **RICE** switch is used to indicate that the data are to be downloaded in a Rice-compressed format. Unlike the **ssw_jsoc_time2data.pro** routine described in the previous section, the **vso_get.pro** does not load the data into an IDL variable. Instead it only downloads the requested FITS files to the specified directory. For Rice-compressed files, the **read_sdo.pro** routine can be used to read in the data that has been downloaded into an IDL variable. Other (non-IDL) software that can handle compressed image files from SDO are listed in Section 4.5.

5.6 Miscellaneous SSW Items

5.6.1 Loading AIA Standard Color Tables

The standardized AIA color tables can be loaded into variables via the **aia_lct.pro** routine (using the 171Å channel as an example)

```
aia_lct,wave=171,red,green,blue
```

or made active using the **/LOAD** switch

```
aia_lct,wave=171,/load
```

Valid arguments for the **WAVE** keyword are **94**, **131**, **171**, **193**, **211**, **304**, **335**, **1600**, **1700**, or **4500**.

5.6.2 Getting AIA Filter Response Data

The filter responses for the EUV channels can be loaded using the **aia_get_response.pro** procedure. The default behavior is to return the response curves as a function of wavelength:

```
;  copy and paste these commands into your SSWIDL session:
result=aia_get_response()
help,result,/str
```

```
IDL> result=aia_get_response()
IDL> help,result,/str
** Structure <2850c08>, 14 tags, length=352536, data length=352504, refs=1:
   NAME            STRING    'AIA'
   DATE            STRING    '20100804_232510'
   CHANNELS        STRING    Array[7]
   WAVE            FLOAT     Array[4001]
   ALL             FLOAT     Array[7, 4001]
   PLATESCALE      FLOAT       8.46158e-12
   UNITS           STRING    'cm2'
   A94             STRUCT    -> <Anonymous> Array[1]
   A131            STRUCT    -> <Anonymous> Array[1]
   A171            STRUCT    -> <Anonymous> Array[1]
   A193            STRUCT    -> <Anonymous> Array[1]
   A211            STRUCT    -> <Anonymous> Array[1]
   A304            STRUCT    -> <Anonymous> Array[1]
   A335            STRUCT    -> <Anonymous> Array[1]
```

The response curves can also be computed as a function of (log) temperature by setting the **/TEMP** switch:

```
;  copy and paste these commands into your SSWIDL session:
result=aia_get_response(/temp)
help,result,/str
```

```
IDL> result=aia_get_response(/temp)
 -------------------------------------------------------
| Generating temperature response function from         |
| /ssw/sdo/aia/response/aia_preflight_all_fullinst.genx |
| /ssw/sdo/aia/response/aia_preflight_fullemiss.genx    |
 -------------------------------------------------------
IDL> help,result,/str
** Structure <19fa008>, 12 tags, length=14944, data length=14912, refs=1:
   NAME            STRING    'AIA'
   LOGTE           FLOAT     Array[101]
   CHANNELS        STRING    Array[7]
   UNITS           STRING    'phot cm^5 s^-1 pix^-1'
   ALL             DOUBLE    Array[101, 7]
   A94             STRUCT    -> <Anonymous> Array[1]
   A131            STRUCT    -> <Anonymous> Array[1]
   A171            STRUCT    -> <Anonymous> Array[1]
   A193            STRUCT    -> <Anonymous> Array[1]
   A211            STRUCT    -> <Anonymous> Array[1]
   A304            STRUCT    -> <Anonymous> Array[1]
   A335            STRUCT    -> <Anonymous> Array[1]
```

When the **/DN** keyword switch is set, the effective area function also includes the DN/photon conversion:

```
;  copy and paste these commands into your SSWIDL session:
```

```
result=aia_get_response(/dn)
help,result,/str
```

```
IDL> result=aia_get_response(/dn)
IDL> help,result,/str
** Structure <5803008>, 14 tags, length=352536, data length=352504, refs=1:
   NAME            STRING    'AIA'
   DATE            STRING    '20100804_232510'
   CHANNELS        STRING    Array[7]
   WAVE            FLOAT     Array[4001]
   ALL             FLOAT     Array[7, 4001]
   PLATESCALE      FLOAT       8.46158e-12
   UNITS           STRING    'cm^2 DN phot^-1'
   A94             STRUCT    -> <Anonymous> Array[1]
   A131            STRUCT    -> <Anonymous> Array[1]
   A171            STRUCT    -> <Anonymous> Array[1]
   A193            STRUCT    -> <Anonymous> Array[1]
   A211            STRUCT    -> <Anonymous> Array[1]
   A304            STRUCT    -> <Anonymous> Array[1]
   A335            STRUCT    -> <Anonymous> Array[1]
```

More examples using **aia_get_response.pro** can be found in your SSW distribution at **$SSW/sdo/aia/response/README.txt**, or online here.

5.6.3 Plotting an AIA Light Curve

Because the image headers are segregated from the images in the JSOC, any quantity that can be derived directly from the keywords can be computed quickly. For example, to plot an AIA light curve, one can use data contained in the **DATAMEAN** and **DATE_OBS** keywords from the image headers:

```
;  copy and paste these commands into your SSWIDL session:
ssw_jsoc_time2data,reltime(days=-2),reltime(/now),index,ds='aia_test.synoptic2',$
  waves='171',key='t_obs,date__obs,wavelnth,wave_str,datamean'
utplot,index.date_obs,index.datamean
```

In this example, a light curve from the 171Å channel for the past two days is plotted using the SSW procedure **utplot.pro**. The **reltime.pro** function (also included in the SSW distribution) is used to set the desired date range.

6 Frequently Asked Questions

- **Browsing and finding data:**
  - **How do I browse through the data?** You can easily browse and download summary images and movies at any of the places mentioned in Section 2.
  - **How do I get a more detailed look instead of just a "browse"?** One option is Helioviewer (see Section 2.4), which enables users to overlay images in multiple layers from multiple data sources. Another option is to query the HCR (see Section 3.1) to see whether data exists for the feature or event of interest.
  - **How do I find data for a specific feature or event, but I haven't determined (or don't know) the dates/times?** Many events and features are cataloged in the HEK (see Section 3.1), which can be queried using iSolSearch (see Section 3.2).
- **Accessing the data:**
  - **How do I get science-grade data for HMI and/or AIA?** First you should determine whether you need the volume of data you are requesting. (If you only need a subfield, or are willing to subsample the data either in space or time, see the next two questions.) Once you have determined that your data volume constitutes a reasonable size, HMI and AIA data are available through the JSOC, either by the *lookdata* web interface (see Section 4.2), the cutout service (see Sections 4.3 or 5.4), or the VSO (see Sections 4.4 or 5.5).
  - **Do I really have to learn about the inner workings of the JSOC in order to get AIA or HMI data?** No, but it helps. Refer to the brief description in Section 4.1, and the links therein for an accessible primer.
  - **How do I request a subfield (localized patch) of the full-resolution image data?** For AIA data, the easiest way to do this is to use the cutout service (see Section 4.3). If your desired subfield is related to that of a HEK event (say, as found using iSolSearch [see Section 3.2]), be aware that each HEK event entry has a link to the cutout service, as well as links to related observations that may point to relevant datasets that have already been created and thus are readily available.
  - **How do I subsample the data in either space or time?** One option is the VSO, which provides a great deal of flexibility in accessing solar data (see Sections 4.4 or 5.5) at various temporal cadences. Another option is to use the "full-disk" and "Max frames" options of the online cutout service form (see Section 4.3), which allows users to readily subsample in both space and/or time.
  - **What about EVE data?** EVE data is available from the EVE data access webpage. Access to EVE data via SolarSoft is currently under development.
- **Analyzing the data:**
  - **Do I need to prepare the data in any way before analyzing it (a la `eit_prep.pro` or `trace_prep.pro`)?** Hopefully not. If you are using Level 1.5 AIA data, the standard and best-available gain (flat-fielding), filter, vignette, and bad-pixel/cosmic-ray corrections will have already been applied. Furthermore, the images will have been rescaled to a standard 0.6″ plate scale, and will have been rotated so that solar north is up in the image. For Level 1.5 HMI data based on filtergrams from the Doppler camera (including Dopplergrams, continuum intensity images, and line-of-sight magnetograms), similar "always-needed" corrections will have already been applied as well.
  - **Are the AIA color tables that are used on the browsing webpages available?** Yes, see Section 5.6.1.
  - **Where can I get a list of HMI or AIA keywords?** Each instrument team has produced a document listing the various keywords. Because many keywords are shared between HMI and AIA, there is probably much overlap between these documents: JSOC keywords for metadata (PDF file from the JSOC wiki) and AIA/SDO FITS keyword list (PDF file from SDOdocs at LMSAL).
  - **Where can I get AIA filter response data?** See Section 5.6.2.

7 Lists of Useful Links

- Observatory and instrument home pages:
  - SDO homepage at Goddard
  - AIA homepage at LMSAL and SDOdocs: AIA documentation
  - EVE homepage at CU/LASP
  - HMI homepage at Stanford
- Data portals:
  - Sungate
  - VSO entry page and search form
  - JSOC data portal and
  - EVE science data access
  - NOAA SWPC data center
- SolarSoft links:
  - SSW installation webpage
  - SSW upgrade webpage
  - SSW codesearch form (for fast doc-header searches of entire SolarSoft library)

8 Contributors to this Guide

- Editors: Marc DeRosa, Greg Slater
- Browse webpages: Dean Pesnell, Kevin Addison ("The Sun Now"), Linus Chang ("The Sun Today"), Peter Gallagher (SolarMonitor), Jack Ireland, Daniel Mueller (Helioviewer)
- JSOC information and demos: Phil Scherrer, Yang Liu, Rock Bush, John Serafin, Jesper Schou
- HEK information and demos: Mark Cheung, Neal Hurlburt, Linus Chang, Ryan Timmons, Ankur Somani, Scott Green
- SSW information and demos: Sam Freeland
- VSO information and demos: Joe Hourclé
- Sage advice: Karel Schrijver, Barbara Thompson, Karin Muglach, and many others...

PDF version of this guide