The Abdus Salam
International Centre for Theoretical Physics

United Nations
Educational, Scientific
and Cultural Organization

International Atomic
Energy Agency

310/1749-27

ICTP-COST-USNSWP-CAWSES-INAF-INFN
International Advanced School
on
Space Weather
2-19 May 2006

EGSO
An Introduction to Solar Soft

Robert BENTLEY
UCL Department of Space and Climate Physics
Mullard Space Science Laboratory
Hombury St. Mary
Dorking
Surrey RH5 6NT
U.K.

These lecture notes are intended only for distribution to participants

An introduction to *SolarSoft*

R.D.Bentley, UCL-MSSL

International Advanced School on Space Weather
Trieste, September 2000

Outline

- What Solarsoft is, its purpose and aims
- Structure of the SolarSoft Tree (SSW)
- The role of the SolarSoft DataBase (SSWDB)
- How the SolarSoft environment is configured and invoked

- Generalized data analysis techniques
- How and where to get help
- Practical session - ask if there are particular things you want to know!!

What is SolarSoft

- ***SolarSoft*** is a set of integrated software libraries, data bases, and system utilities which provides a "common" programming and data analysis environment for the Solar Physics community.

- The **SolarSoftWare (SSW)** system is built from libraries developed for Yohkoh, SoHO and TRACE, and the SDAC and Astronomy libraries. It draws upon contributions from many members of those projects.

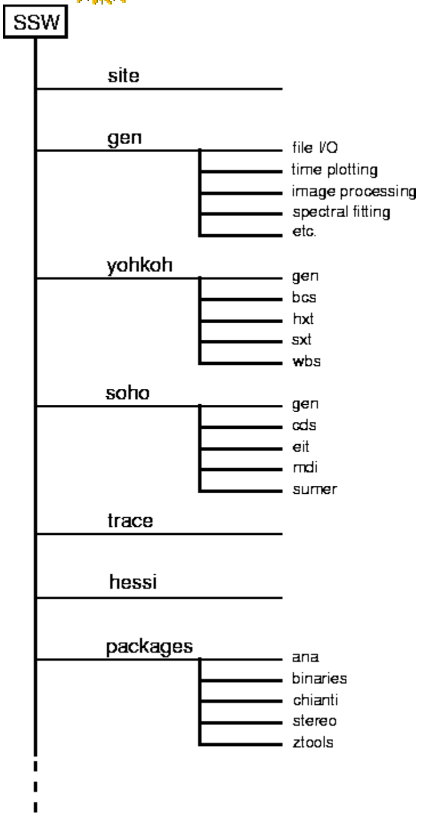
- **SSW** is primarily an **IDL** based system, although some instrument teams can and do integrate executables written in other languages.

- The **SSW** environment provides a consistent look and feel at various CoI institutions (Yohkoh, SoHO, etc.). This facilitates the sharing and exchange data and minimize the time taken to "come up-to-speed" when doing research away from your home institute.

Aims of SolarSoft

- **Promote the use of standards that will facilitate coordinated data analysis**
  - Suggested standards make many powerful routines immediately available for the analysis of new data sets.
- **Provide a hardware and site independent analysis system**
  - Fundamental shared software is written in a largely hardware-independent and site-configuration independent form. The **SSW** setup utilities support local configuration files that tailor the user's environment.
- **Provide a large reusable software library**
  - The software library represents an evolutionary system tracing back to SMM, through Yohkoh and SOHO.
- **Provide easy access to supporting ancillary data bases**
  - Access to supporting data sets from ground-based and satellite-based solar observatories are provided in consistent formats to facilitate coordinated analysis.
- **Provide integrated access to other IDL packages**
  - When IDL libraries are identified that would provide a useful extension to **SSW** capabilities, they are integrated into the tree.

The SolarSoft Tree


SSW
site
gen
file I/O
time plotting
image processing
spectral fitting
etc.
yohkoh
gen
bcs
hxt
sxt
wbs
soho
gen
cds
eit
mdi
sumer
trace
hessi
packages
ana
binaries
chianti
stereo
ztools

- The ***SolarSoft*** tree consists of many branches with a similar structure
- Branch for each mission, etc
  - SOHO, TRACE…
  - Also branches for ground based

  Observations: optical & radio
- ***$SSW/gen*** contains a reusable library of many general purpose routines
- Configuration of ***SolarSoft*** to your site is with file under the ***$SSW/site*** directory

Re-usable Software Library

- The re-usable software library represents an evolutionary system dating back to SMM. It developed through Yohkoh and SOHO, and now incorporates routines related to TRACE and other solar observatories.
- Amongst the many common tasks already included are file I/O, system access, IDL structure manipulation, data display, etc.. Spacecraft and instrument specific routines are dealt with in other, separate branches.
- ***Existing library capabilities include:***
  - Time series analysis, time conversions, time series plotting (UTPLOT)
  - Spectral fitting
  - Image and Image cube manipulation and display (including movies)
  - Solar image data routines (limb fitting, grid overlay, co-ordinate transformations, co-alignment...)
  - File I/O (generic binary, ASCII, FITS...)
  - IDL data manipulation (structure, string, array, mathematics...)
  - WWW related routines

Other Branches of the Tree

Other branches support, or are planned for the following instruments

- **Current and Past Spacecraft:**
  - SoHO: CDS EIT MDI SUMER LASCO (1996)
  - Yohkoh: BCS HXT SXT WBS (1991)
  - TRACE: Trace (1998)
  - KORONAS-ReSIK (2001)
  - HESSI (2001)
  - GOES-SXI (2001)
  - Others: GOES, CGRO, Spartan, SMM
  - Virtual Observatories (EGSO, VSO, CoSEC)
- **Future Spacecraft:**
  - Solar-B (2006)
  - STEREO (2006)
  - SDO (2008)
- **Packages:**
  - chianti, xpex, xray, etc.
  - ANA, binaries

Structure of the branches

- **The branches are similar in form:**
  - gen (includes the re-usable library)
  - missions with single or multiple instruments (e.g. TRACE; SoHO)
    - *mission/gen*
    - *mission/instrument* (e.g. soho/cds)
  - packages
  - **site**
- **Each branch has *standard* directories:**
  - doc
  - response (calibration information)
  - **idl** (can include many sub-directories)
  - **setup** (defines configuration for a branch)

- The **IDL path** is assembled from the sub-directories in the **idl** directories of the selected branches
- The **setup directories** contain files key to the function of ***SolarSoft*** - these define the location required instrument and ancillary data.
  - The files have names like: setup.*mission*_env and setup.*instrument*_env

Configuring SSW

- The setup files contain environment variables in the Unix C-shell format:

```
setenv  DIR_SXT_SDC $SSWDB/ydb/sdc
setenv  DIR_SXT_CAL $SSW/sxt/response
```

- The setup files at ssw, mission and instrument level should define almost everything needed to configure the SSW environment for an installation.
- If any instrument or ancillary files are stored in non-standard places, include setup files with the appropriate name in the **site** setup directory. ***Do not make modifications within the tree.***
- The setup files are executed in strict hierarchy:
  - $SSW/site/setup/setup.ssw_paths
  - $SSW/gen/setup/setup.mission.env
  - $SSW/site/setup/setup.mission.env
  - $SSW/mission/xxx/setup.xxx
  - $SSW/site/setup/setup.xxx

This is not as complex as it seems. It is not a fatal mistake if an environment variables are defined in the wrong place, so long as the overall order of execution has the desired effect. Also, a ***simple installation may require few changes***

The SSWDB Tree

- The ***SolarSoftWare Database*** tree (**SSWDB**) contains ancillary data provided by the experiment teams. It has a similar outline structure to the **SSW** tree – there are branches for each mission, and sub-branches for some instruments.

- The **Yohkoh** branch ($ydb)
  - Combines all datasets at mission level.
  - Some data sets are general (e.g spacecraft attitude)
  - Others are instrument specific (e.g. SXT CCD data)
  - ***Catalogues*** in form of instruments observing logs
  - Includes supporting datasets from other sources
- The **SoHO** branch is split by instrument
  - Some instruments store more than others
  - Catalogs for CDS and EIT, MDI must be generated
- The **TRACE** branch ($tdb)
  - Includes a ***catalogue***

Installation of SSW and SSWDB

- **The master copies of *SolarSoft* and SSWDB are maintained at the Goddard Space Flight Center (GSFC).** Software and ancillary data from instrument groups around the world are are mirrored into the SSW and SSWDB trees for general distribution.
- ***SolarSoft* can be installed using a web page in the SSWDOC pages**. This generates an installation package that is mailed to the user. On execution this will create directories, and copy and unpack tar files containing the requested software branches.
  - **Only install the branches that are needed**
  - Always get gen and packages/binaries
  - For some instrument, other things are installed automatically
    - HESSI always gets packages/spex and /xray
  - ***Updates by mirror*** (Perl script running FTP)

- **Required parts of the *SolarSoft DataBase* can also be installed using a web page in the SSWDOC pages**. Usually, you only need ancillary files for a short interval either side of a dataset under study. (Note: the is interface is still under development)

Invoking SolarSoft

- The environment variable **SSW must have been defined**. Installed tree is then open to you.
- Environment variable **SSW_INSTR** defines the selection of instrument branches that will be used to create the desired SolarSoft environment.
- If you are using ancillary data, **SSWDB** needs to have been defined.

- In Unix and Linux, this is defined ***outside*** IDL with the command setssw. IDL is then started with the command sswidl.Thus:

```
setssw sxt eit cds
sswidl
```

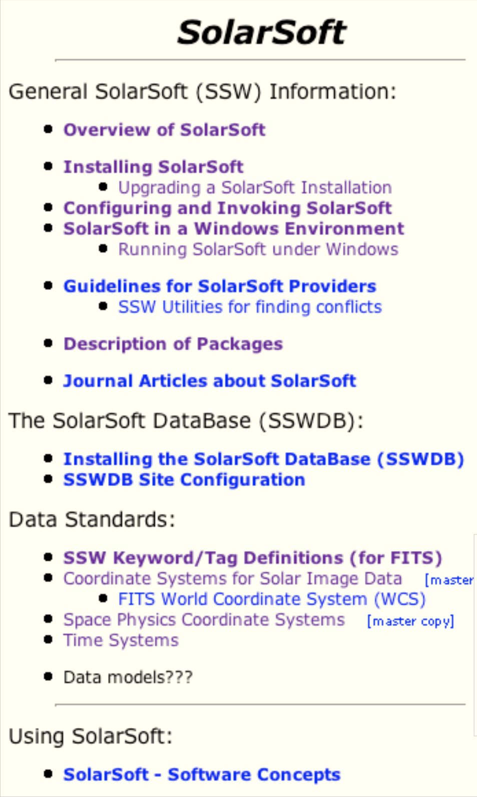
- For Windows, the selection is done ***within*** IDL, using the command setssw_windows (e.g. setssw_windows,/sxt). IDL is invoked using a .bat file that defines SSW before it calls IDL.

- The **IDL_STARTUP** routines allow the user to run some procedures when IDL is invoked.

How to get help - I

- There are many documents on the Web that can provide assistance for SolarSoft
- An integrated set of documentation, SSWDOC, including SolarSoft documents and user guides, is provided. To access it, open **`index.html`** in the sswdoc tree

How to get help - I



SolarSoft

General SolarSoft (SSW) Information:

- Overview of SolarSoft
- Installing SolarSoft
  - Upgrading a SolarSoft Installation
- Configuring and Invoking SolarSoft
- SolarSoft in a Windows Environment
  - Running SolarSoft under Windows
- Guidelines for SolarSoft Providers
  - SSW Utilities for finding conflicts
- Description of Packages
- Journal Articles about SolarSoft

The SolarSoft DataBase (SSWDB):

- Installing the SolarSoft DataBase (SSWDB)
- SSWDB Site Configuration

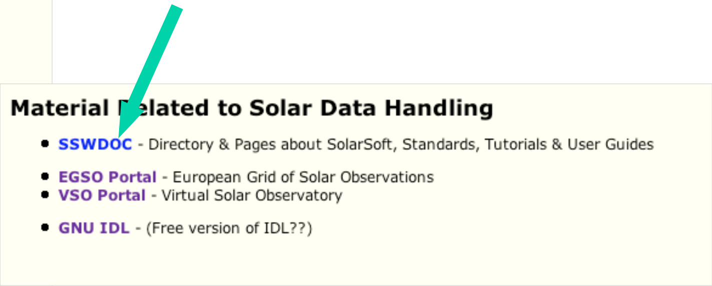
Data Standards:

- SSW Keyword/Tag Definitions (for FITS)
- Coordinate Systems for Solar Image Data [master
  - FITS World Coordinate System (WCS)
- Space Physics Coordinate Systems [master copy]
- Time Systems
- Data models???

Using SolarSoft:

- SolarSoft - Software Concepts

- There are many documents on the Web that can provide assistance for SolarSoft
- An integrated documentation set, SSWDOC, is provided – this includes SolarSoft documents and user guides. To access it, open **`index.html`** in the sswdoc tree and then select **`sswdoc`**



Material Related to Solar Data Handling

- SSWDOC - Directory & Pages about SolarSoft, Standards, Tutorials & User Guides
- EGSO Portal - European Grid of Solar Observations
- VSO Portal - Virtual Solar Observatory
- GNU IDL - (Free version of IDL??)

Data Analysis with SolarSoft

- **How to look at data in a generic manner**
  - Not all instruments are the same
  - Structure of analysis software define by the instrument teams
  - Routines in the gen branch try to cope with the differences in structure of the data and naming conventions
  - ***Lack of rigid rules is one of the strengths of SolarSoft; also a problem***

- **Combining data from several instruments**
  - Overlay images
  - Compare time series of data or derived parameters

Steps in Processing Data

- For each instrument the steps involved in accessing the data are very similar:
  - Locate the data and/ or **catalogue** for the instrument for the required time interval
  - **Access the catalogue** or data roadmap (index) to find the modes run during this interval
  - Make a selection of the modes
  - Read the data and index information for the selected modes

- Review the data by displaying it – *images, spectra, light-curves*

- **Calibrate the data**
- **Derive physical parameters**

- **Compare with data sets** from other instruments

Locate the instrument data

- Whereas the SSW and SSWDB trees combine elements from many missions and instruments, the instrument data or observations are normally stored on an instrument by instrument basis.

- This is necessary because of the shear bulk of the data, e.g. Yohkoh produces 500 Mbyte/week, and TRACE produces >3 Gbytes/week.

- The data is again addressed through environment variables. For example

– Yohkoh:	YOHKOH_DATA
– TRACE:	TRACE_I1_DIR
– MDI:	MDI_MAGS
– EIT:	EIT_LZ
– CDS:	CDS_FITS_ROOT (at MSSL)

- The data is often organized under such environment variables in sub-directories divided by weeks, months, or some other useful quantity.
- Search software identifies correct files by date range

Data and index structures

- Once you have located you data, it needs to be read in

- Many different **file formats** are used - the ***format is defined by the experiment team who also provide the read routines***
- The size of some datasets makes ***some form of selection essential***, and may necessitate special processing in some cases. In some cases this is defined by the instrument team.

- The read routines often returns a **index structure**. If so, there are many generic routines available that allow selection of the data by:
  - **mode** (e.g. channel, filter)
  - **time** (absolute time or cadence)
  - **pointing**
  - **match the times of other data**

- If you want to look at the contents of a structure

```
help,/st,structure_name
```

Selecting a data subset

- The routine **struct_where** is a generalized filter that works on index or catalogue structures. The selection is made according to filters supplied in a configuration file or string array. For example:

```
ss = struct_where(index,conf_file=file)
ss = struct_where(index,test_array=array)
```

- Filters are of the form:

```
<TAG> <OPERATOR> <VALUE>
```

- For example, a configuration file to select **TRACE** data might contain:

```
NAXIS1 = 512,1024
WAVE_LEN = 171,195,284            XCEN=600.~800.
IMG_MIN > 1.
IMG_AVG > 100 && IMG_MAX < 4096
```

- To select a series of images at a pre-defined cadence, use **grid_data**. To select images at a particular time, use **tim2dset**. For example:

```
ss = grid_data(index, min=5)
 and
ss = tim2dset(index, tref)
```

Some useful Library routines

Too many routines go into detail - look at the help pages in SSWDOC for guidance on techniques to analyse a particular dataset.

To look at the time history of data, look at:

utplot

If you are interested in comparing images from different instrument, over and expended interval, look at:

Map Objects

See the Zarro description

ssw_track_fov

track field of view over time

How to get help - II

Within **SSWIDL**, several useful procedures are available to provide help:

xdoc, *’routine_name’*
doc_library, *’routine_name’*
allow the user to search for routines and examine them (or just the document header)

chkarg, *’routine_name’*
returns the call parameters of a routine

sswloc, *’search_string’*
*Lists **names** of routines in the **SolarSoft tree** that match the search string. Note: sswloc also work at the Unix shell level as “sswloc ‘search_string’”*

pr_path
returns the current IDL path
pr_env [,/*search_string*]
allows the user to examine what environment variables defined

More information and Tutorials

- SolarSoft - Software Concepts

- Tutorials:
  - Finding/reading FITS data, plotting maps, etc.
  - Techniques for dealing with 3D datasets
  - EIT, MDI, Mapping and SSW Techniques
  - TRACE, SXT, MDI, and SSW Techniques