IDL_HS Installation Guide

James Tappin

20th February 2002

This document is intended to facilitate the effective installation of IDL_HS version 4 on various systems.

One big change from version 3 is that it is now possible to install it in one place and run on multiple systems. It is also possible for the administrator to set default options appropriate to the site.

Note: *all the notes that follow are appropriate to Unix/Linux systems; for VMS systems, you may need to do a bit of fiddling to get things right (I've forgotten most of the VMS DCL I ever knew).*

Windows and MacOS are not supported. If you want to attempt a port, by all means do so but I can't help[1].

1 General

You will need the following components, the explicitly-named files are to be found on `ftp://ftp.sr.bham.ac.uk/pub/ulysses` .

1. The idl_hs distribution (`idl_hs-4.vv.vv.tar.gz`).
2. A recent version of IDL. Development has been done on version 5.2 and 5.4, however it is possible that earlier version 5 releases will work. Version 4 and before do not support objects and pointers so they won't work.
3. The hsio distribution (`hsio-3.1.2.tar.gz`) 3.1.1 will work but you will need to sort out the file names manually.
4. The ulio distribution (`ulio-0.2.0.tar.gz`). This is a new item, it is a library of routines to read ULEIS HDF files. (Version 0.2 fixes a problem which caused the HDF subsystem to run out of space after 32 files had been opened).
5. Version 4.1 of the NCSA HDF libraries, you will need the source code.

The baseline installation requires the following steps:

1. Unpack the tar file in a suitable directory. If your tar command does not support the `z` option (i.e. if it's not GNU tar), then you will need to uncompress it with `gunzip` first.

[1] 2-reasons, firstly I don't know anything about those systems and second I don't have time to figure out what needs to be done.

2. Ensure that everything has the right permissions (it should have, but if there are problems then use

   ```
   chmod -R a+rX
   ```

   in the idl_hs directory.

3. Build the hsio sharable library, this is needs the hsio version 3 sources and install it in the `lib` subdirectory. With version 3.1.1 of hsio you will need to rename the `hsioshr.so` file to `libhs_<sys>.so` if you want to use the defaults in the hsio routines. More detailed instructions for this are given below. If you already have a working installation of an earlier release of IDL_HS version 4, then you can simply copy the sharables from that installation to the new one.

4. Build the ulio sharable library. Instructions below. As with hsio, you can just copy from an earlier installation.

5. Run the `install.sh` script in the top-level directory. This will attempt to guess a number of settings and then ask for verifications/corrections, it will then edit a number of script templates to create a working installation.

6. Edit the resource file `Data/idl_hs.rc` to suit your site, or copy from an earlier installation. See the user's guide for a full list of possible resources. In particular you will need to change the various data search paths, probably the default plot path and the spool command. The `rc_maker` procedure run as `rc_maker,/system` from within IDL_HS will guide you through setting up the resource file.

7. If you use any non-standard names for your HISCALE or EPAM files you will need to edit `Data/ulanames.dat` to set the name prefixes correctly.

8. Make a soft link from the `idl_hs` script in the `bin` subdirectory to a directory that is in users' paths.

2 Installing HSIO

Installing HSIO is relatively simple. If you have version 3.1.2 then just follow the instructions in the documentation that comes with it. The stuff about libc versions on Linux systems should now not be a problem as current versions of IDL use libc 6, and idl_hs_4 needs version 5 of IDL (if you are still using a libc 5 based release of Linux then you need to upgrade).

If you want to support multiple platforms, then you will need to check that you have a suitable setting of `OS_TAIL` in `Mk.shr`

3 Installing ULIO

The installation of ULIO requires some care, partly because it's newer so I've not got it down to a fine art yet.

1. Install HDF

   (a) Obtain the HDF source code from NCSA—a binary distribution will probably not work.

   (b) Modify the config files for your system to include the `-fPIC` option in the compile stage (otherwise you won't be able to include the library into a sharable—at least this is the case on Solaris).

(c) Follow the rest of the installation instructions for your system.

2. Unpack the ulio file.
3. Edit the Makefile:
   (a) Change the setting of `HDFDIR` to wherever the HDF stuff was installed (the libraries should be in `${HDFDIR}/lib` and the include files in `${HDFDIR}/include`.
   (b) Change `IDLDIR` to point to the top-level directory of the idl_hs distribution.
   (c) For Solaris or Linux uncomment the relevant sets of options for compilation and comment out the ones you don't need.
   For other systems you'll need to make up a set using the `Mk.shr` file from HSIO and the examples in the IDL distribution as a guide.
4. `make`
5. `make install`

With a bit of luck you'll be home.

4 Setting up for multiple platforms

As of version 4 it is relatively easy to modify IDL_HS so that a single tree can be used for multiple platforms.

To do this you must do 2 things:

- Install the hsio and ulio sharable libraries for each system, using different names, (e.g. `libhs_sun.so` and `libuleis_sun.so` for Solaris, `libhs_lnx.so` and `libuleis_lnx.so` for Linux etc.), this is now the default behaviour.
- Edit the `idl_hs` script in the `bin` subdirectory and modify the cases to suit the systems you have and the names you used.
  Example for Solaris, Linux and OSF1:

```
case ‘uname -s‘ in
SunOS) LIBHS=libhs_sun.so ; LIBULEIS=libuleis_sun.so ;;
Linux) LIBHS=libhs_lnx.so ; LIBULEIS=libuleis_lnx.so ;;
OSF1) LIBHS=libhs_osf.so ; LIBULEIS=libuleis_osf.so ;;
esac
export LIBHS LIBULEIS
```

If you have only one platform then you can remove the whole platform case statement from the script, and set the `OS_TAIL` value to an empty variable in the makefiles for the libraries.

If you have a different path for your IDL system libraries on different platforms then you'll need to move the setting of SYS_IDL_DIR into this case statement.

5 The Resource File

The IDL_HS resource file is the main tool for setting up IDL_HS for your system. The system-wide file is in `idl_hs.rc` in the `Data` subdirectory. In addition users may set up personal files in their home directories (the personal file is called `.idl_hsrc` on Unix/Linux and `idl_hs.rc` on VMS). Settings made in the personal file will override those in the system-wide file (however, see the note on paths).

The search paths for the various classes of data are proper search paths consisting of colon-separated lists of directories on Unix/Linux or comma-separated lists on VMS. For efficiency it is a good idea to put the directories containing averaged data before the raw-data directories as the search for matching filenames is sequential. If no value is specified for a path, then the user's current directory is used. For any PATH setting made in a resource file, prepending the list with a + causes the list to be appended to any previous path, thus:

```
HS_RATE_PATH = /datan3/ulysses/cuaf:/datanca/ulysses/ula
```

and

```
HS_RATE_PATH = /datan3/ulysses/cuaf
HS_RATE_PATH = +/datanca/ulysses/ula
```

are equivalent.