HIRDLS

SW-HIR-2009

Originator: Charles Cavanaugh

Date: 10 Dec. 2012

Subject/Title:

The User's Guide of the HIRDLS Level 0 – Level 1 Processor

Description/Summary/Contents:

The purpose of this document is to describe the building, running and validating of the High Resolution Dynamics Limb Sounder (HIRDLS) Level 0 to Level 1 Processing System. The goal of this document is to detail the description sufficiently, such that the user requires no more guidance than that what is listed in this document.

Keywords:

Purpose of this Document:

Oxford University
Atmospheric, Oceanic &
Planetary Physics
Parks Road
OXFORD OXI 3PU
United Kingdom

University of Colorado, Boulder
Center for Limb Atmospheric Sounding
3450 Mitchell Lane, Bldg. FL-0
Boulder, CO 80301

EOS

The User's Guide of the HIRDLS Level 0 – Level 1 Processor

Charles Cavanaugh

Table of Contents

Table of Contents		i
List of Figures and Tables.		ii
Section 1	Document Purpose and Goal	1
Section 2	References	1
Section 3	Creating the System	1
Section 3.1	Extracting the System	1
Section 3.2	Platform Dependencies	1
Section 3.3	External Dependencies	2
Section 3.4	Compiling the System	2
Section 4	Editing the System	2
Section 5	Running the System	3
Section 5.1	Building the Run	3
Section 5.2	Resource Dependencies	3
Section 5.3	Ancillary Files	3
Section 5.4	Invoking the System	4
Section 6	Validating the System	4
Section 6.1	Toolkit Log Files	4
Section 6.2	System Log Files	5
Section 6.3	Metadata	6
Section 6.4	Graphical & System Tools	6

List of Figures and Tables

Figure 1 Extracted System 1
Figure 2 External Scripts 2
Figure 3 Sandbox Listing 3
Figure 4 Ancillary File Listing 4
Figure 5 Usfmask Parameter Value Listing 5
Figure 6 L1Report.txt Listing 7

1 Document Purpose and Goal

The purpose of this document is to describe the building, running and validating of the High Resolution Dynamics Limb Sounder (HIRDLS) Level 0 to Level 1 Processing System. The goal of this document is to detail the description sufficiently, such that the user requires no more guidance than that what is listed in this document.

2 References

The HIRDLS Level 0 (L0) to Level 1 (L1) Processing System, hereby known as L1 Processor, has had its requirements, architecture and design already detailed in previous documents. It is not necessary to read those documents before using L1 Processor, as those documents detail the construction of the system. However, it would be helpful to read those documents before using L1 Processor, as those documents also contain useful information on system input and output files, and also contain information on the purpose of the system.

3 Creating the System

L1 Processor is included in the DAP (Delivered Algorithm Package) as a gzipped file, called "L1Processor.gz". This section will detail how to create the system.

3.1 Extracting the System

To build L1 Processor, copy the gzipped L1 Processor file to the directory in which you want the system to be built, and then extract the files (thereby building the system) by using the Unix "tar" utility. This will create a directory that includes all the source code, make files, and control files that you will need. Other ancillary files (e.g., Aura ephemeris files), however, will be necessary to run L1 Processor, and those are described in Section 5.3. Figure 1 shows a practical example of the steps in this section.

```
hal:/hal/cavanaug/L1Processor> ls
-rw-r--r-- 1 cavanaug hirdls 3810779 Sep 24 11:11 L1Processor.gz
hal:/hal/cavanaug/L1Processor> tar xf L1Processor.gz
hal:/hal/cavanaug/L1Processor> ls
-rw-r--r-- 1 cavanaug hirdls 3810779 Sep 24 11:11 L1Processor.gz
drwxr-xr-x 4 cavanaug hirdls      39 Apr  4 15:47 v8.0.11/
```

Figure 1 Extracted System

3.2 Platform Dependencies

L1 Processor, overall, was developed and implemented to be Unix platform independent and ANSI compatible. L1 Processor uses ASCII files and HDF5 files during processing, and both of these file structures are platform independent. L1 Processor also uses binary files provided by ESDIS, and obviously these are not platform independent, so care must be taken when accessing these files. There is no "platform" switch when building L1 Processor, so it is possible that the source code may need to be altered to accommodate a change in machine endianness.

3.3 External Dependencies

In order to run, L1 Processor needs access to the SDP Toolkit. L1 Processor uses this toolkit to perform such tasks as time conversion and geo-location. As of this document version, the current toolkit version is 5.2.18. L1 Processor should be built with this version, as it is not guaranteed that L1 Processor will work correctly with previous versions. Also, the SDP Toolkit libraries that L1 Processor accesses must be compiled with the same compiler as used on L1 Processor (see Section 3.4). The SDP Toolkit also provides environment-creating scripts that must be run before compiling or running L1 Processor. These scripts are platform-dependent, and can be found in the toolkit directory, in the appropriate platform "bin" subdirectory. Figure 2 shows a listing that could be used, depending on the platform.

```
/usr/local/TOOLKIT5.2.18gnu/bin/linux/pgs-env.ksh
/usr/local/TOOLKIT5.2.18gnu/bin/linux/pgs-dev-env.ksh
/usr/local/TOOLKIT5.2.18gnu/hdfeos/bin/linux/hdfeos_env.ksh
```

Figure 2 External Scripts

3.4 Compiling the System

To create the L1 Processor binary executable, go to the "driver" subdirectory of the "C++/packages" subdirectory of the build you created. In that subdirectory are 5 Makefiles that control compilation of L1 Processor. At most, only two of them, "Makefile.exec" and "Makefile.global", should need editing.

"Makefile.global" contains compilation flag definitions, and these can be platform-specific. The default listings of "CFLAGS" and "EXECCFLAGS" are minimal and should work on all platforms, but you are certainly welcome to add your favorite compilation flags. The default for "FASTCFLAGS" turns optimization on, to level 2. Again, this should work for most cases. The defaults for "WARNCFLAGS" and "DEBUGCFLAGS" should also work for most cases. This file also contains a soft reference to the compiler to use to build L1 Processor. The default is GNU's g++ compiler.

"Makefile.exec" contains the library listing to use for compilation. If these libraries are different on your machine, you will need to adjust this listing. And like "Makefile.global", "Makefile.exec" contains the same compiler soft link.

"Makefile.system" should not need editing, but contains the three included compilation targets: "systemfast", "systemdebug" and "systemwarn". These targets control which compilation flags are used during compilation. The default way to build L1 Processor is to use the "systemfast" target. If you want to be able to access L1 Processor with a debugger during a run, use the "systemdebug" target. If you want to see compilation warnings during compile, use the "systemwarn" target. It is recommended that for any extensive run jobs, you use the "systemfast" target to generate L1 Processor. To actually build L1 Processor, invoke the make command with one of these three targets, i.e., at the Unix prompt, enter "make systemfast" (without the parentheses). Upon successful completion, the system executable, called "L1X", will have been created in the "run/sandbox" subdirectory, as shown in Figure 3.

4 Editing the System

It is strongly recommended that you successfully complete Section 3 before you do any editing, to make sure that any issues that might arise are not due to faulty instructions in this user's guide. If you edit the source code, you will of course need to recompile the system. If you edit any of the input files included in the DAP, there is no need to recompile the system, unless of course you change the internal structure of a file. Any changes to the system should be documented in the "history" ancillary file.

```
hal:/hal/cavanaug/L1Processor/v8.0.11/run/sandbox> ls
-rw-r--r-- 1 cavanaug hirdls     876 Apr  4 15:44 ExclusionInstrument.txt
-rw-r--r-- 1 cavanaug hirdls     803 Apr  4 15:44 ExclusionRandom.txt
-rw-r--r-- 1 cavanaug hirdls    1158 Apr  4 15:44 ExclusionSpacecraft.txt
-rw-r--r-- 1 cavanaug hirdls    7877 Apr  4 15:44 HIRDLS1.mcf
-rw-r--r-- 1 cavanaug hirdls  693520 Apr  4 15:44 HIRFOV_OOF.he5
-rw-r--r-- 1 cavanaug hirdls   16172 Apr 12 13:26 history
-rwxr-xr-x 1 cavanaug hirdls 9968985 Sep 14 15:44 L1X*
-rw-r--r-- 1 cavanaug hirdls   26261 Apr  4 15:44 L1X_pcf.txt
-rwxr-xr-x 1 cavanaug hirdls    2952 Apr  4 15:44 L1X.py*
-rw-r--r-- 1 cavanaug hirdls    1891 Apr  4 15:44 Mnemonics286.txt
-rw-r--r-- 1 cavanaug hirdls    3676 Apr  4 15:44 Mnemonics288.txt
-rw-r--r-- 1 cavanaug hirdls    2316 Apr  4 15:44 Mnemonics289.txt
-rw-r--r-- 1 cavanaug hirdls   12485 Apr  4 15:44 proc.py
-rw-r--r-- 1 cavanaug hirdls   10626 Apr  4 15:44 RadiometricFlux.txt
-rw-r--r-- 1 cavanaug hirdls    1269 Apr  4 15:44 RadiometricParameters.txt
```

Figure 3 Sandbox Listing

5 Running the System

To run L1 Processor, it is assumed you have followed the steps in Section 3 or Section 4, and have a compiled version of the system. Congratulations! That was the difficult step. This section will detail how to run the system

5.1 Building the Run

To build the run, you will first need to create a directory in which you want the run to happen. As detailed in the next section, you will need to consider how much free space you have in that directory. Once you have that directory established or newly created, copy all the files from the "run/sandbox" subdirectory, as listed in Figure 3, into the run directory. Also copy the "usfmask" file from the "run" subdirectory (one level up from the "sandbox" subdirectory).

5.2 Resource Dependencies

Running L1 Processor requires at least 6.5 Gbytes of available memory and up to 800 Mbytes of available storage space (depending on the extent of the processing request). L1 Processor requires only one thread.

5.3 Ancillary Files

Not included in L1 Processor's build are the input HIRDLS L0, Aura attitude and Aura ephemeris files necessary for a successful run. The specific files necessary are dependent on the processing request, and a DAP that included all potential combinations would be impossibly wieldy. HIRDLS has received these ancillary files from ESDIS, and it is recommended that you do too. Also note that these files are all in binary format and therefore are platform dependent.

For a standard day's run, you will need the L0 file for the respective day, and the L0 file for the day preceding the respective day. You will also need the same setup for the ephemeris files. The attitude files, unlike the ephemeris and L0 which are 24-hour files, are 2-hour files, and therefore you will need 14 of them to process a standard day. 12 of the files are for the respective processing day, 1 is for the last ephemeris file for the preceding day, and 1 is for the first ephemeris file for the

subsequent day. Figure 4 shows a typical listing of these files (note that the processing day request is 2006d138, or May 18, 2006). Section 5.4 details how L1 Processor is notified of the input L0, attitude and ephemeris files.

```
hal:/hal/cavanaug/L1Processor/v8.0.11/run/sandbox> ls
-rw-rw-r-- 1 cavanaug hirdls 748808320 May 24 14:55 HIR0SCI_FIXED_v1.3-aIrix-c1_2006d137.PDS
-rw-rw-r-- 1 cavanaug hirdls 748808320 May 24 14:55 HIR0SCI_FIXED_v1.3-aIrix-c1_2006d138.PDS
-rw-rw-r-- 1 cavanaug hirdls   5531104 May 24 14:55 AUREPHMN_vna-ana-c1_2006d137.eph
-rw-rw-r-- 1 cavanaug hirdls   5531104 May 24 14:55 AUREPHMN_vna-ana-c1_2006d138.eph
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d137h22-2006d137h23.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h00-2006d138h01.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h02-2006d138h03.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h04-2006d138h05.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h06-2006d138h07.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h08-2006d138h09.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h10-2006d138h11.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h12-2006d138h13.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h14-2006d138h15.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h16-2006d138h17.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h18-2006d138h19.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h20-2006d138h21.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d138h22-2006d138h23.att
-rw-rw-r-- 1 cavanaug hirdls    462848 May 24 14:55 AURATTN_vna-ana-c1_2006d139h00-2006d139h01.att
```

Figure 4 Ancillary File Listing

5.4 Invoking the System

L1 Processor is invoked by calling the python script "L1X.py" with the input "usfmask" text file. This input text file, an example of which is shown in Figure 5, contains parameters read in by L1 Processor. These parameters are then used to fill in the run file needed by the SDP Toolkit. The values of the parameters in the "Processor Specifics" and "Script Specifics" sections should remain as they are, except for "L1XVERSION", whose value should be edited, if necessary, to reflect the current version. In the "Input Files" section, only the values of the parameters "L1XLEVEL0", "L1XEPHEMERIS" and "L1XATTITUDE" would need editing, as these values change with each day's run. Use the sample listing in Figure 5 as a template for these values. All the values for the parameters in the "Output Files" section can be changed to whatever you like, with the "HIRDLS1" parameter value reflecting the name of the HIRDLS1 file that L1 Processor will create. The values of the parameters in the "Processing Time Range" section will also need to be changed to the start and stop time, respectively, of the data to process. The listing in Figure 5 denotes one entire day of processing. To reiterate, once the "usfmask" file contains valid values, invoke L1 Processor by entering, at the Unix prompt, "L1X.py usfmask" (without the parentheses).

6 Validating the System

To validate the system, it is assumed you have followed the steps in Section 5, and have an output HIRDLS1 file. Congratulations! You are 2/3 the way to having a useful HIRDLS Level 1 file. This section will detail how to validate the system.

6.1 Toolkit Log Files

In the directory in which L1 Processor was run, you should have 3 ASCII toolkit log files. These are named "LogReport", "LogStatus" and "LogUser". The "LogStatus" file is the useful one. This is the file into which SDP Toolkit writes its errors, warnings and reports. There is an endless permutation of these three types of messages, and it is impossible to detail them all. But if the only message, besides the ubiquitous "W A R N I N G" message about Toolkit version mismatch, is "No reference was found matching Product ID (47000) and Version number (3).", then L1 Processor has run successfully, in the view of SDP Toolkit.

```
# usfmask
# Mask file used by run script to fill L1X process control file
# Charles Cavanaugh

# Processor Specifics
L1XEXEC    = L1X
L1XVERSION = v8.0.11

# Script Specifics
L1XPCF_MASK = L1X_pcf.txt
L1XPCF      = L1X.pcf

# Input Files
HIRDLS1MCF      = HIRDLS1.mcf
L1XMNEMONICS286 = Mnemonics286.txt
L1XMNEMONICS288 = Mnemonics288.txt
L1XMNEMONICS289 = Mnemonics289.txt
L1XEXCLINST     = ExclusionInstrument.txt
L1XEXCLSC       = ExclusionSpacecraft.txt
L1XEXCLRANDOM   = ExclusionRandom.txt
L1XRADFLUX      = RadiometricFlux.txt
L1XRADPARAMS    = RadiometricParameters.txt
L1XRADOOF       = HIRFOV_OOF.he5
L1XLEVEL0       = HIR0SCI_FIXED_v1.5-aItanium-Suse-c1_2006d137.PDS,HIR0SCI_FIXED_v1.5-aItanium-Suse-
c1_2006d138.PDS
L1XEPHEMERIS    = AUREPHMN_vna-ana-c1_2006d137.eph,AUREPHMN_vna-ana-c1_2006d138.eph
L1XATTITUDE     = AURATTN_vna-ana-c1_2006d137h22-2006d137h23.att,AURATTN_vna-ana-c1_2006d138h00-
2006d138h01.att,AURATTN_vna-ana-c1_2006d138h02-2006d138h03.att,AURATTN_vna-ana-c1_2006d138h04-
2006d138h05.att,AURATTN_vna-ana-c1_2006d138h06-2006d138h07.att,AURATTN_vna-ana-c1_2006d138h08-
2006d138h09.att,AURATTN_vna-ana-c1_2006d138h10-2006d138h11.att,AURATTN_vna-ana-c1_2006d138h12-
2006d138h13.att,AURATTN_vna-ana-c1_2006d138h14-2006d138h15.att,AURATTN_vna-ana-c1_2006d138h16-
2006d138h17.att,AURATTN_vna-ana-c1_2006d138h18-2006d138h19.att,AURATTN_vna-ana-c1_2006d138h20-
2006d138h21.att,AURATTN_vna-ana-c1_2006d138h22-2006d138h23.att,AURATTN_vna-ana-c1_2006d139h00-
2006d139h01.att

# Output Files
HIRDLS1        = HIRDLS1_8011_2006d138.hdf
L1XREPORT      = L1Report.txt
L1XDIAGNOSTICS = L1XDiagnostics.txt
L1XSCREENOUT   = L1X_stdout

# Processing Time Range
PROCESSINGSTARTTIME = 2006-05-18T00:00:00.000Z
PROCESSINGSTOPTIME  = 2006-05-19T00:00:00.000Z
```

Figure 5 Usfmask Parameter Value Listing

6.2 System Log Files

Upon finishing, L1 Processor generates an ASCII file called "L1Report.txt", which is a status report of the L1 Processor run. L1 Processor was designed to always create this file, no matter the status of the run. The only time this file won't be generated is if L1 Processor dumps core. If this is the case, you need to go back to Section 4 and review why this happened.

Figure 6 details the formatted listing of "L1Report.txt", which are occurences that L1 Processor deems worthy of special note. Figure 6 shows a typical, non-eventful run of L1 Processor. Of special note is the last line. If the last line does not start with "Normal Termination", then L1 Processor did not normally terminate. If this is the case, the message string after the colon will give a succinct reason why L1 Processor terminated abnormally, and this can be used as a starting to point to determine what went wrong.

The first 8 lines of "L1Report.txt" should look very much like what is shown in Figure 6. The number of packets read, which will then be reflected in the azimuth and elevation encoder block lines, could vary. If the following 29 lines do not have zero

occurrences each, something went very wrong with L1 Processor, or more specifically, something was very wrong with the input to L1 Processor.

It is possible that the following 3 lines ("L0 packets with mif/rev …", "Revs with unacceptable time gap", "Revs with geolocation error") could have more than zero occurrences each, as these did indeed happen on some days when HIRDLS had sync issues. If these issues happened during nominal HIRDLS scans, those scans will not be assigned "scan numbers", and therefore will not be further processed by HIRDLS processing software.

There were many days when a celestial body was in HIRDLS' field of view (FOV), and the light reflected off one of these bodies was strong enough to corrupt the limb radiances generated by HIRDLS. Occurrences of these corruptions are shown in the next 4 lines. The Moon, Mars, Venus and Jupiter are the only celestial bodies listed, as these are the only ones whose reflected light could corrupt HIRDLS radiances. As with those issues listed in the preceding paragraph, if any of these corruptions happened during nominal scans, those scans with not be further processed.

Any occurrences of "Instances with unrecognized scan table" will also be handled as described in the two preceding paragraphs.

Occurrences of "Scans excluded" are directly related to the listing in the three "Exclusion" files in the L1 Processor run directory. These three ASCII files contain listing of times that HIRDLS is to ignore for various reasons, including spacecraft maneuvers and known instrument anomalies. Again, any nominal scans affected will not be further processed.

The "Scans removed" line will contain a count of the number of scans that were adversely affected by any of the above situations. It is possible to have occurrences of celestial body corruption, unrecognized scan tables or scan exclusions that do not happen during nominal scans, and therefore no scans will be removed. This is an important note.

6.3 Metadata

A normally-terminated run of L1 Processor will generate a metadata file for the output HIRDLS1 file, and the name of the file will be the same as the HIRDLS1 file, but with ".met" attached at the end of the name. The information in this file is used for ingestion of the HIRDLS1 file into the ESDIS system, and does not contain much useful information. The parameters that include "DATE" and/or "TIME" could be examined to check veracity of the run.

6.4 Graphical & System Tools

There are numerous graphical and/or system tools that will closely examine the fields inside the HIRDLS1 file. Included in this list are "h5ls" and "h5dump". The HIRDLS program relied heavily on Matlab and IDL programs to examine the contents of all the generated HIRDLS data products. These are not included in the DAP, as these were generated with minimal documentation and minimal notation, and were intended for in-house use only. If you are interested in using any of these, please contact your HIRDLS representative.

```
          2  L0 files opened
    1800020  L0 packets read
          0  L0 azimuth primary encoder block used
    1800020  L0 azimuth secondary encoder block used
          0  L0 elevation primary var encoder block used
          0  L0 elevation primary 2 encoder block used
    1800020  L0 elevation secondary var encoder block used
          0  L0 elevation secondary 2 encoder block used
          0  L0 azimuth encoder blocks all invalid
          0  L0 elevation encoder blocks all invalid
          0  L0 science housekeeping blocks invalid
          0  L0 radiance blocks invalid
          0  L0 time blocks invalid
          0  L0 packets skipped because rdsr not 1, 2, 3 or 6
          0  L0 radiance blocks with data size disparity
          0  L0 packets with radiance data quality error set
          0  L0 packets with channel 01 radiance not present
          0  L0 packets with channel 02 radiance not present
          0  L0 packets with channel 03 radiance not present
          0  L0 packets with channel 04 radiance not present
          0  L0 packets with channel 05 radiance not present
          0  L0 packets with channel 06 radiance not present
          0  L0 packets with channel 07 radiance not present
          0  L0 packets with channel 08 radiance not present
          0  L0 packets with channel 09 radiance not present
          0  L0 packets with channel 10 radiance not present
          0  L0 packets with channel 11 radiance not present
          0  L0 packets with channel 12 radiance not present
          0  L0 packets with channel 13 radiance not present
          0  L0 packets with channel 14 radiance not present
          0  L0 packets with channel 15 radiance not present
          0  L0 packets with channel 16 radiance not present
          0  L0 packets with channel 17 radiance not present
          0  L0 packets with channel 18 radiance not present
          0  L0 packets with channel 19 radiance not present
          0  L0 packets with channel 20 radiance not present
          0  L0 packets with channel 21 radiance not present
          0  L0 packets with mif/rev out-of-sequence
          0  Revs with unacceptable time gap
          0  Revs with geolocation error
          0  Revs with moon in FOV
          0  Revs with mars in FOV
          0  Revs with venus in FOV
          0  Revs with jupiter in FOV
          0  Instances of unrecognized scan table
          0  Scans excluded
          0  Scans removed
     112520  Mafs written to L1 file
Normal Termination : No anomalies during processing
```

Figure 6 L1Report.txt Listing