# Terrenus Earth Sciences HRPT Software: User Manual

**Version 2.2**
**Revised June 13, 2012**

Prepared by:
Peter Hollemans, Terrenus Earth Sciences
Web site: www.terrenus.ca

## Copyright Notice and Statement for Terrenus HRPT Software



## Obtaining a Copy

To download a copy of the Terrenus Earth Sciences HRPT software, visit the software page at:

`http://www.terrenus.ca`

## Providing Feedback

Email general questions, bug reports and source code modifications to the author at `peter@terrenus.ca`. Detailed bug reports are preferred, including such information as the operating system, the software version, command line parameters, sample data files, and so on.

Specific software support and added functionality requests **will not be accepted without a Terrenus Earth Sciences software support agreement**. Contact `peter@terrenus.ca` to discuss specific software support requests.

# Contents

# Preface

## Typographic Conventions

In this manual, we use the following conventions for the font and color of text. References within the document are in the standard font, but are red to emphasize that they are active links, for example a reference to a section: "see Appendix A for information on data formats". External references are a `typewriter` font in magenta, such as a web site address: "`http://www.google.com` is a great search engine". Typed commands, file names, and verbatim character strings are in a `typewriter` font. Java class names are in *italics*. Replacement parameters are in CAPITALS.


## Acknowledgments

We would like to acknowledge the following parties for support in creating the Terrenus HRPT software (in no particular order):

- Katherine Kidwell of NOAA as editor of the NOAA KLM User's Guide, available at `http://www2.ncdc.noaa.gov/docs/klm`. This guide is invaluable as a reference on NOAA polar orbiter satellites and sensors.
- Dr. Xiaofeng Li of NOAA/NESDIS for providing descriptions of the Quorum HRPT capture format and inspiration for writing the software.
- Allan B. Bundens of Quorum Communications, Inc. for discussions on the Quorum HRPT capture format.
- Howard Carney of NOAA/NESDIS for discussions on orbital modeling code, Earth coordinate systems, coordinate transformations, and view vector intersection computations.
- The NCSA Hierarchical Data Format (HDF) project at `http://hdf.ncsa.uiuc.edu`. The HDF format software and documentation are a model of usability.
- Pedro J. Fernandez for the development of the SGP4 orbital model in Java based on the code from Dr. T.S. Kelso (`http://celestrak.com/software/satellite/PassFinder.zip`).
- JAMA, the linear algebra package for Java from `http://math.nist.gov/javanumerics/jama`.
- The CelesTrak website at `http://www.celestrak.com` for information on orbital modeling and two-line element data.

- Mohammed Kamal, Guido Cervone, Jacek Radzikowski of George Mason University and Sergey Diakov and Sergey Katamanov of the Far Eastern Branch Russian Academy of Sciences for help with FY-1D data decoding and calibration.

- Bolli Palmason for bug reports and testing.

# Chapter 1

# Introduction

The Terrenus Earth Sciences HRPT software is used for processing raw capture station NOAA HRPT and China Meteorological Administration (CMA) CHRPT data. HRPT and CHRPT data (hereafter referred to collectively as "HRPT") may be captured from NOAA and CMA polar orbiting weather satellites using relatively low-cost receiving hardware available from a number of commercial vendors. The HRPT software includes a Java API and ingestor program that converts the AVHRR or MVISR sensor data in the HRPT telemetry stream to calibrated, earth-located data in HDF format. The resulting HDF files are compatible with the CoastWatch Utilities distributed from the NOAA CoastWatch website (`http://coastwatch.noaa.gov`) as well as other commercial packages such as IDL/ENVI (`http://www.rsinc.com`). Various capture station HRPT formats are supported, including Quorum, Global Imaging, and SeaSpace format.

This manual describes how to install and use the software, and a discussion of the Java code design for interested developers. The software has been placed under the GNU General Public license in the hope that it will be useful to researchers and others and that it will be improved through collaboration among developers and users. Feedback on the code should be sent to the author at `peter@terrenus.ca`, but software support questions ("How can I write a script to update the orbital element files automatically?") and added functionality requests ("Can you make it read HRPT format XYZ?") will not be handled without a Terrenus Earth Sciences software support agreement.

## 1.1 Requirements

The HRPT software is currently available for Windows, Linux (Intel), and Mac OS X (Universal). Even though the compiled Java code is operating system independent, the HDF format writing routines are compiled from C code and are specific to the operating system. Send email to the author for other operating systems, as it may be possible to provide a custom version.

Before installing the software, Java 1.5.0 or higher is required. Java packages are available from the Sun web site, for example `http://java.sun.com/getjava`.

## 1.2 Installation

### 1.2.1 On Windows

To install on Windows, download and run the Windows installation package. The installation will place an entry in the Windows Start Menu for the user manual and uninstall program. The `hrptingest.exe` program located in the `bin` directory may only be run from the command line via the Windows Command Prompt.

### 1.2.2 On Linux

To install on Linux, download and untar the Linux installation package:

```
tar -zxf hrpt_2_0_linux.tar.gz
```

This will create a directory named `hrpt`. To run the hrptingest program, type `hrpt/bin/hrptingest`.

### 1.2.3 On Mac OS X

To install on Mac OS X, download and double-click the Mac disk image (dmg), and run the installation program. Mac users must run the ingestor from the Terminal program, for example type `/Applications/Terrenus\ HRPT/bin/hrptingest`.

# Chapter 2

# Using the Software

The following sections describe how to run the HRPT ingestor to create calibrated radiometer data in HDF format from raw HRPT data. The ingestor uses an orbital model to compute the position and velocity of the satellite through time, and requires special two-line orbital element data files, discussed in section 2.2. The ingestor also requires current AVHRR or MVISR sensor calibration data, discussed in section 2.3.

## 2.1 Running the Ingestor

### Synopsis

```
hrptingest [OPTIONS] input output
```

**Options:**

```
-h, --help
-i, --informat=TYPE
-f, --outformat=TYPE
-m, --minvalid=COUNT
-o, --origin=STRING
-V, --version
-y, --year=YYYY
-d, --day=DDD
-c, --creationdate
```

### Description

The hrptingest program converts the high-resolution radiometer sensor data in a NOAA HRPT or CMA CHRPT raw capture station data file into calibrated, earth-located data in HDF. Currently, only NOAA KLM and CMA FY1 satellites are supported, including NOAA-15, -16, -17, -18, -19, and FY-1D.

AVHRR or MVISR sensor data is quality checked and re-calibrated every 5 lines. In some cases the output data may contain missing scan lines where the corresponding telemetry frame data was corrupt. The output HDF or NetCDF file contains all AVHRR or MVISR channels calibrated to scientific units, as well as earth locations and ancillary angles. The file contains HDF SDS or NetCDF variable data and file/variable metadata conforming to the CoastWatch HDF Metadata Specification, version 3.4 (http://coastwatch.noaa.gov/cw_formats.html). The SDS variables written are as follows:

| | |
|---|---|
| `avhrr_ch??` | AVHRR channel data, visible albedo data in percent and thermal radiance data in Celsius (black body temperatures), 16-bit signed integer values, scaling factor 0.01. |
| `mvisr_ch??` | MVISR channel data, same conventions as AVHRR channels. |
| `latitude` | Geodetic latitude in degrees, 32-bit floating point data. |
| `longitude` | Geographic longitude in degrees, 32-bit floating point data. |
| `sat_zenith` | Satellite zenith angle in degrees, 16-bit signed integer data, scaling factor 0.01. |
| `sun_zenith` | Solar zenith angle in degrees, 16-bit signed integer data, scaling factor 0.01. |
| `rel_azimuth` | Relative azimuth angle in degrees, 16-bit signed integer data, scaling factor 0.01. |

## Parameters

**Main parameters:**

`input` The input data file name.

`output` The output data file name.

**Options:**

`-h, --help` Prints a brief help message.

`-i, --informat=TYPE` The input format. The supported formats are:

- `auto` - Automatic format detection (default)
- `quorum-hrpt` - Quorum QTrack HRPT
- `quorum-chrpt` - Quorum QTrack CHRPT
- `global` - Global Imaging
- `seaspace` - SeaSpace TeraScan
- `seaspace-swap` - SeaSpace TeraScan byte-swapped
- `raw-hrpt` - Raw NOAA HRPT
- `raw-chrpt` - Raw CMA CHRPT
- `1a5-hrpt` - NOAA 1A5 HRPT
- `1a5-chrpt` - CMA 1A5 CHRPT

`-f, --outformat=TYPE` The output format. The supported formats are:

- `auto` - Automatic format detection from file extension (default)

- `hdf` - Hierarchical Data Format (HDF)
- `nc` - Network Common Data Form (NetCDF)

`-m, --minvalid=COUNT` Specifies the minimum number of consecutive valid data lines at the start and end of the data. Data beyond this boundary is discarded. The default is to write all data lines to the output file. Invalid data lines are detected by examining the frame synchronization words and TIP data parity.

`-o, --origin=STRING` Specifies the data origin string for the HDF 'origin' attribute. The default is to write 'unknown' for the origin, which is not very informative.

`-V, --version` Prints the software version number.

`-y, --year=YYYY` Specifies the four digit year of data capture. By default, the current year is assumed, unless the year can be determined from the input data contents. Currently, only the Quorum and 1A5 formats contain the year.

`-d, --day=DDD` Specifies the three digit day of data capture (January 1 = 001). By default, the current day is assumed, unless the day can be determined from the input data contents. Currently, only raw CHRPT requires a day to be specified.

`-c, --creationdate` Specifies that the values for year and day of data capture (see -y and -d above) should be obtained from the file creation date. By default the current year and day are assumed unless the date can be determined from the input data contents.

## Exit status

0 on success, > 0 on failure. Possible causes of errors:

- Invalid command line option.
- Invalid input or output file names.
- Unrecognized format of input data.
- Unknown satellite code in input file.
- No applicable orbital elements available.

## Examples

The following shows the ingest of HRPT data from a Quorum data file:

```
phollema@bean<Data> hrptingest -V
[hrpt 2.0] hrptingest
Java 1.5.0_07 on Mac OS X 10.4.9 i386
phollema@bean<Data> hrptingest 2006-07-14-1905-n18.QuorumHRPT 2006-07-14-1905-n18.hdf
hrptingest: Opened Quorum QTrack HRPT telemetry data file with 4426 frames
hrptingest: Data capture year is 2006
```

```
hrptingest: Detected spacecraft noaa-18 launched 2005/05/20 JD 140
hrptingest: Data starts at 2006/07/14 JD 195 19:05:25.245 UTC
hrptingest: Data ends at   2006/07/14 JD 195 19:17:42.745 UTC
hrptingest: Decoding data from frames 0 through 4425
hrptingest: Writing avhrr radiometer channel and angle data
```

The CoastWatch Utilities can then be used to query and analyze the data:

```
phollema@bean<Data> cwinfo 2006-07-14-1905-n18.hdf
Contents of file 2006-07-14-1905-n18.hdf

Global information:
  Satellite:           noaa-18
  Sensor:              avhrr
  Date:                2006/07/14 JD 195
  Time:                19:05:25 UTC
  Scene time:          day
  Projection type:     swath
  Origin:              unknown
  Format:              CoastWatch HDF version 3.4

Variable information:
  Variable       Type    Dimensions  Units         Scale     Offset
  avhrr_ch1      short   4426x2048   percent       0.01      0
  avhrr_ch2      short   4426x2048   percent       0.01      0
  avhrr_ch3      short   4426x2048   celsius       0.01      0
  avhrr_ch4      short   4426x2048   celsius       0.01      0
  avhrr_ch5      short   4426x2048   celsius       0.01      0
  avhrr_ch3a     short   4426x2048   percent       0.01      0
  latitude       float   4426x2048   degrees       -         -
  longitude      float   4426x2048   degrees       -         -
  sat_zenith     short   4426x2048   degrees       0.01      0
  sun_zenith     short   4426x2048   degrees       0.01      0
  rel_azimuth    short   4426x2048   degrees       0.01      0

phollema@bean<Data> cwstats --sample 0.01 2006-07-14-1905-n18.hdf
Variable       Count     Valid     Min         Max         Mean        Stdev
avhrr_ch1      90815     90815     1.93        106.77      15.970976   16.357994
avhrr_ch2      90815     90815     0.95        123.13      21.458364   15.816578
avhrr_ch3      90815     90815     -23.78      64.31       27.48028    13.601457
avhrr_ch4      90815     90815     -80.25      51.45       12.980522   20.530457
avhrr_ch5      90815     90815     -79.74      46.62       10.031554   20.063936
avhrr_ch3a     90815     0         NaN         NaN         NaN         NaN
latitude       90815     90815     13.350684   59.875427   37.012815   12.258589
longitude      90815     90815     -119.264236 -68.644142 -88.972349 8.879522
sat_zenith     90815     90815     0.21        69.11       32.621807   19.494969
sun_zenith     90815     90815     5.53        46          24.720583   7.579942
rel_azimuth    90815     90815     0           179.99      84.804809   60.840523
```

## 2.2 Orbital Element Updates

Two-line orbital element data are used by the Terrenus HRPT software to generate latitude, longitude and other angle data for each sensor scan line. The elements are used in conjunction with a Java implementation of the NORAD SGP4 orbital model to predict the position and velocity vectors of the satellite at a given date and time. Generally, the accuracy of the orbital prediction is best when the prediction time is close to the orbital element epoch time. For that reason, the Terrenus HRPT software will only accept orbital elements whose epoch time is within a three day window of the satellite capture time.

When run in a near-real time mode (ie: data is processed upon capture), the ingestor automatically obtains the required orbital element data over the network from the CelesTrak web site using the URL `http://www.celestrak.com/NORAD/elements/noaa.txt`, or from the telemetry data file in the case of Quorum format. However, when run in a reprocessing mode (ie: data is months or years old), the ingestor must rely on stored orbital element data in the `resources/terrenus/orbit` subdirectory of the software installation. The default installation contains current orbital element data obtained from the CelesTrak website for NOAA-15, -16, -17, -18, -19 and FY-1D up to the software release date. For other HRPT data, orbital elements must be manually added to the resource files.

As an example, a two-line orbital element set for NOAA-18 on November 24, 2005 looks as follows:

```
1 28654U 05018A 05328.40630561 .00000392 00000-0 24229-3 0 2496
2 28654 98.7651 271.3254 0015174 81.8167 278.4732 14.10905129 26504
```

To add the data to the resource files, open the `resources/terrenus/orbit/noaa18.txt` file in a text editor and add the two new element lines to the end of the file. Then save the file and run the hrptingest program as normal. For NOAA-18 HRPT data between November 21 and 27, the newly added elements for November 24 will be used unless more accurate elements can be found elsewhere. See the CelesTrak web site at `http://www.celestrak.com` for more information about orbital elements and prediction models.

## 2.3 Radiometer Calibration Updates

### 2.3.1 AVHRR

The AVHRR instrument records up to three visible wavelength channels and three thermal wavelength channels. The thermal channels (3b, 4, and 5) are calibrated against an on-board warm body target, but the three visible channels (1, 2, and 3a) have no on-board calibration target and must be calibrated manually by NOAA researchers. The visible channel calibration coefficients are updated periodically and notices are posted to the NOAA/NESDIS web site at `http://www.osdpd.noaa.gov/ml/ppp/notices.html`. These notices should be periodically checked, and if any updates are available, the new coefficients copied into the Terrenus HRPT software resources. Out of date coefficients can have a negative impact on any subsequent processing of the data, such as cloud masking or NDVI computations.

The default installation of the software contains AVHRR visible channel calibration coefficients that are current as of the software release date. The coefficients are stored in Java property files in the `resources/terrenus/orbit` subdirectory of the software installation. Each satellite has its own property file which contains prominent notices denoting the section for visible coefficients and when they were last updated. For example, to update the visible channel coefficients for NOAA-18 AVHRR channel 1, open the `resources/terrenus/orbit/noaa18.properties` file and edit the values for `avhrr.ch1.s1` (low slope), `avhrr.ch1.i1` (low intercept), `avhrr.ch1.s2` (high slope), `avhrr.ch1.i2` (high intercept), and `avhrr.ch1.cut` (cutoff count). Similarly, update the coefficients for other AVHRR channels then save the file and run the hrptingest program as normal.

### 2.3.2 MVISR

The Chinese MVISR instrument has 7 visible channels (1, 2, 6, 7, 8, 9, 10) and three thermal channels (3, 4, 5). MVISR channel data is calibrated similarly to AVHRR – visible channels must be calibrated manually by CMA researchers, and thermal channels are calibrated against an on-board warm body target. Visible channel updates are not posted regularly but my be obtained from CMA personnel. The Terrenus HRPT software includes visible calibration coefficients received by personal communication with Guido Cervone and Jacek Radzikowski of George Mason University on January 22, 2007. Thermal channel coefficients were obtained from "A.I. Aleksanin, C.E. Diakov, S.N. Katamanov, Y.V. Naumkin. Technology processing data of polar-orbital satellites FY-1C/1D for physical ocean field monitoring. Journal of Underwater Investigation and Robotics, Institute of Marine Technology Problems, 2006, 2, p. 82-91." MVISR calibration coefficients are stored in the `resources/terrenus/orbit/fy1d.properties` file, and may be edited similarly to the AVHRR calibration. The main difference in coefficients is that the MVISR uses only a single slope and intercept for visible channel calibration where as AVHRR uses a dual slope and intercept.

# Chapter 3

# For the Developer

This chapter gives details for developers who are interested in using the Terrenus HRPT software Java packages for custom development. For example different sensor data could be extracted from the HRPT stream, or the orbital model could be used for different satellites.

## 3.1 Installation Layout

The software installation contains a number of files and subdirectories as follows:

| | |
|---|---|
| `bin` | Binary directory containing hrptingest program. |
| `build.xml` | XML build file for Apache Ant. |
| `doc` | Software API, license, and user manual. |
| `ext` | External code libraries. |
| `hrpt.install4j` | Install4j packager configuration file. |
| `hrpt.properties` | Apache Ant properties file. |
| `lib` | Java .jar files and shared libraries. |
| `resources` | Orbital element and AVHRR calibration data. |
| `src` | Java source code. |

The main files of interest to developers are:

| | |
|---|---|
| `lib/hrpt.jar` | Terrenus HRPT compiled Java classes. |
| `lib/Jama-1.0.1.jar` | Java matrix support for AVHRR sensor model. |
| `lib/sgp.jar` | SGP4 orbital prediction model. |
| `doc/api/index.html` | Java API for Terrenus HRPT classes. |

## 3.2 Java Packages

There are three major Java class packages in the Terrenus HRPT software:

*terrenus.telemetry* The telemetry package contains classes for reading and decoding raw spacecraft telemetry data. *TelemetryStream* is an interface that specifies all telemetry data will be delivered as frames of byte blocks, while the implementing classes (*QuorumHRPTStream*, *GlobalHRPTStream*, and so on) handle the specifics of delivering the data from the contents of different capture station format files. The *TelemetryStreamFactory* class opens a telemetry data file and automatically detects the data format. A *TelemetryDecoder* uses a stream to deliver spacecraft information, telemetry frame errors, and instrument data.

*terrenus.instrument* The instrument package contains classes for processing data from various spacecraft instruments, including AVHRR and MVISR. The *RadiometerData* interface allows access to raw and calibrated multichannel radiometer data, as well as sample location and angle data.

*terrenus.orbit* The orbital model package contains classes for managing two-line orbital element data, tracking spacecrafts over time, and computing Earth ellipsoid intersections. The *OrbitalElementManager* class acts as a general purpose factory for finding and returning orbital element data for a specific spacecraft and date. Of main interest are the *SpacecraftLocator* class which constructs using a two-line element set and returns spacecraft position and velocity vectors at a given date and time, and the *SensorModel* class which models a spacecraft image sensor and computes the intersection between sensor view vectors with the surface of the Earth. The *Spacecraft* interface and implementing classes *AbstractSpacecraft*, *NOAAKLMSpacecraft*, and *FY1Spacecraft* simplify access to spacecraft orbital information. An example of using these classes may be found in the *terrenus.instrument.LineScannerLocator* class.

## 3.3 How to Compile

Compiling the Terrenus HRPT software itself may be accomplished using the included Apache Ant build file. Typing `ant -p` shows a list of compile targets. The default target is `classes`, which compiles the Java source files in `src` to the `classes` directory and creates the `lib/hrpt.jar` file. The API is rebuilt in `doc/api` using the `javadoc` target. The `dist` target uses the install4j software from ej-technologies (`http://www.ej-technologies.com`) to create Window, Linux, and Mac OS X installation packages using the the included install4j configuration file. install4j is commercial software and requires a license.

To use the HRPT API, developers should add `lib/hrpt.jar` to the class path, and optionally `lib/sgp.jar` for orbital model support and `lib/Jama-1.0.1.jar` for sensor model support. Note that excluding the *terrenus.instrument.CWHDFDataWriter* class, the Terrenus HRPT software is entirely platform independent (ie: no native code calls).

## 3.4 Code Testing

A number of the Java classes have built-in test programs to aid in code testing. To run a test program, type:

```
ant test -Dclass="classname" -Dargs="arguments"
```

For example:

```
phollema@Beanlet<hrpt-2.0> ant test
  -Dclass="terrenus.orbit.AbstractSpacecraft" -Dargs="28654"
Buildfile: build.xml

classes:

test:
     [java] Name:           noaa-18
     [java] Launch date:    2005/05/20 00:00:00 +0000
     [java] Current date:   2007/03/23 23:05:25 +0000
     [java] Position (ECF): -0.57, -0.26, 0.95 (norm)
     [java] Position (GEO): 56.72 N, -155.30 E (deg)
     [java] Velocity (ECF): 4.55, 4.37, 3.92 (km/s)
     [java] Solar zenith:   55.98 (deg)

BUILD SUCCESSFUL
Total time: 8 seconds
```

The test program classes and their arguments are as follows:

`terrenus.instrument.AVHRRCalibrator fileName`
Calibrates AVHRR data and prints calibration records.

`terrenus.instrument.AVHRRLocator fileName`
Computes AVHRR sample earth locations using two different methods and compares the results.

`terrenus.instrument.MVISRCalibrator fileName`
Calibrates MVISR data and prints calibration records.

`terrenus.instrument.MVISRData fileName startFrame endFrame startSample endSample`
Extracts and prints MVISR raw channel data.

`terrenus.orbit.AbstractSpacecraft noradId`
Uses a NORAD spacecraft ID to retrieve current orbital elements and print the current spacecraft position and velocity.

`terrenus.orbit.ArchiveElementSource url noradId msecs`
Uses a NORAD spacecraft ID to find and print archive-style orbital elements from a URL for a time offset some number of milliseconds +/- from the present.

`terrenus.orbit.CurrentElementSource url noradId`
Uses a NORAD spacecraft ID to find and print current-style orbital elements from a URL for a time within 5 days of the present.

`terrenus.orbit.FY1Spacecraft`
Prints the current position and velocity of CMA FY1 spacecrafts.

`terrenus.orbit.NOAAKLMSpacecraft`
Prints the current position and velocity of NOAA KLM spacecrafts.

`terrenus.orbit.SensorModel`
Tests the class using theoretical view vectors.

`terrenus.telemetry.AbstractTelemetryStream fileName startFrame endFrame blocks`
Reads telemetry data and prints out the first few blocks of 16-bit values from each frame.

`terrenus.telemetry.QuorumFile fileName`
Reads a Quorum HRPT file and prints out various file information including the ephemeris data.

`terrenus.telemetry.TelemetryStreamFactory fileName`
Opens a telemetry stream data file and prints frames, start time, end time, satellite, and instrument properties.

`terrenus.telemetry.HRPTDecoder fileName`
Opens a telemetry data stream file and prints out the quality indicators for each frame.

# Appendix A

# HRPT Formats

This section describes the raw capture station NOAA HRPT and CMA CHRPT formats supported by the Terrenus HRPT software. HRPT data is broadcast by NOAA satellites in a well-known format, documented at http://www2.ncdc.noaa.gov/docs/klm/html/c4/sec4-1.htm. CHRPT is broadcast by CMA satellites and is documented at http://www.sat.dundee.ac.uk/hrptformat.html. Capture station hardware manufacturers often store the raw data on disk in an enhanced format that includes headers, special data start, stop, and resync records, ephemeris data, and so on. The supported variants of "broadcast HRPT" are as follows:

### Quorum QTrack HRPT/CHRPT

- Little endian byte order
- Possible header of unknown size
- HRPT pseudo frames signaled by a 0x8000 start word
- 22180 bytes per HRPT minor frame or 44360 bytes per CHRPT frame
- 10-bit words are stored as least significant 10 bits in each 16-bit word

### CMA 1A5 HRPT/CHRPT

- Little endian byte order
- 21980 byte file header for HRPT, or 44360 byte file header for CHRPT
- 21980 bytes per HRPT minor frame composed of 1500 byte frame header and 20480 byte channel data for HRPT, or 44360 bytes per CHRPT frame composed of 3200 byte frame header and 40960 byte channel data
- 10-bit words are stored as least significant 10 bits in each 16-bit word

## Global Imaging HRPT

- Big endian byte order
- 240 byte file header
- 22180 bytes per HRPT minor frame
- 10-bit words are stored as least significant 10 bits in each 16-bit word

## SeaSpace TeraScan HRPT

- Either big endian or little endian byte order
- 22528 bytes per HRPT minor frame
- 10-bit words are stored as least significant 10 bits in each 16-bit word

## Raw HRPT

- Big endian byte order
- 22180 bytes per HRPT minor frame
- 10-bit words are stored as least significant 10 bits in each 16-bit word

## Raw CHRPT

- Big endian byte order
- 44360 bytes per CHRPT frame
- 10-bit words are stored as least significant 10 bits in each 16-bit word

# Appendix B

# Acronyms

**API** Application Programming Interface

**AVHRR** Advanced Very High Resolution Radiometer

**CHRPT** Chinese High Resolution Picture Transmission

**CMA** China Meteorological Administration

**HDF** Hierarchical Data Format

**HRPT** High Resolution Picture Transmission

**IDL** Interactive Data Language

**MVISR** Multispectral Visible and IR Scan Radiometer

**NDVI** Normalized Difference Vegetation Index

**NESDIS** National Environmental Satellite, Data, and Information Service

**NOAA** National Oceanic and Atmospheric Administration

**NORAD** North American Aerospace Defense Command

**SDS** Scientific Data Set

# Appendix C

# History of Changes

## Version 1.0, 2005/12/05

- Initial release

## Version 1.1, 2006/10/18

- Corrected ellipsoid equation coefficient. This was causing the ellipsoid intersection calculations to compute inaccurate earth locations at large satellite zenith angles.

- Changed from geocentric to geodetic satellite subpoint. The KLM user's guide specifies that the attitude control system orients the satellite such that the scan direction axis points at the geodetic satellite subpoint. Previously, the software used the geocentric subpoint which is up to 2.5 km from the geodetic subpoint at 45° latitude.

- Added a package for Mac OS X.

- Updated the visible calibration coefficients from the NOAA 1B notices website: `http://www.osdpd.noaa.gov/PSB/PPP/notices/notices.html`.

## Version 1.2, 2006/12/04

- Corrected problem with satellite view vector computation. This was causing the ingest to stall indefinitely with certain datasets.

- Added iteration loop limit in geocentric to geodetic latitude computation.

## Version 2.0, 2007/04/06

- Completely re-wrote the low level data reading and radiometer calibration code to be more generic, under the concept of telemetry streams and instruments. Separated out the *terrenus.hrpt* package into *terrenus.telemetry* and *terrenus.instrument*. Moved the main ingestor program to the *terrenus.tools* package.
- Added handling for CHRPT data and the MVISR sensor on the FY-1D satellite.
- Added new capabilities to the hrptingest program: automatic file format detection, version option, minimum valid start/end lines option, 1A5 file format support.
- Updated the AVHRR visible calibration coefficients from the NOAA 1B notices website: `http://www.osdpd.noaa.gov/PSB/PPP/notices/notices.html`.
- Added orbital element data for all supported satellites up to March 24, 2007.
- Added missing HDF DLL files for Windows installation package.
- Moved the location of satellite properties and orbital elements files to `resources/terrenus/orbit`.
- Added new sections to the user manual: acronyms, developer code testing programs, CHPRT and MVISR sections.
- As a result of code modification, the AVHRR instrument now always produces data for channel 3a (`avhrr_ch3a`) *and* channel 3b (`avhrr_ch3`) regardless of which channel is active. Special missing data values are written to the output file for whichever channel is not active.
- Updated code to take advantage of new Java 1.5 language features.

## Version 2.1, 2010/12/18

- Added NetCDF output format option to hrptingest. This is useful so that hrptingest can be run entirely in Java for OS-independence – no binary libraries are used during execution when in NetCDF output mode.
- Added NOAA-19 support.
- Updated the AVHRR visible calibration coefficients from the NOAA 1B notices website: `http://www.osdpd.noaa.gov/ml/ppp/notices.html`.
- Added orbital element data for all supported satellites up to December 18, 2010.
- Added support for raw CHRPT format ingest.
- Added options to hrptingest for manually setting the day, and for setting the year/day from the file creation date. This is primarily for raw CHRPT which has no year or day available in the dataset.

## Version 2.2, 2012/06/13

- Corrected problem with satellite position during March in leap years.
- Corrected problem with NOAA-18 missing lines.
- Added orbital element data for all supported satellites up to software release date.
- Corrected problem with missing line data in NetCDF output files.
- Added support for TeraScan CHRPT format ingest.
- Added modified SGP4 orbital model Java source files to distribution.
- Added testing code for terrenus.telemetry.HRPTDecoder.
- Updated the AVHRR visible calibration coefficients from the NOAA 1B notices website: `http://www.osdpd.noaa.gov/ml/ppp/notices.html`.
- Added Xcode project file for ease of development on Mac.

# Appendix D

# Known Issues

- HDF format output is only possible using a Java VM that supports 32-bit mode, because the HDF native libraries are compiled for a 32-bit processor. Using a 64-bit JVM will cause hrptingest to issue an error message about not being able to find a suitable architecture. To sidestep this issue, use NetCDF format output, or install a Java VM with 32-bit support (some VMs will have an option to run in 32-bit mode).
- Automatic format detection does not function correctly in all circumstances. It may be necessary to force hrptingest to assume an input format in some cases.
- Some input formats require that the year and/or day of the year be explicitly given on the command line. This is only required when the input data format does not contain the full date information. If the year and/or day of the year is not specified for file formats that do not contain it, the ingestor assumes the current year and day as determined by the clock on the computer. If this assumption is incorrect, or if the date parameters passed on the commmand line are incorrect, the computed earth location data will be incorrect. The only way to know if this has occurred is to inspect the earth location data by hand.
- For developers, there is no coherent testing method, for example one that tests the ingest of all supported HRPT input formats.