OPENDCS
Addendum for USDA SNOTEL

Revision 6

October, 2021

Prepared For

Technology Solutions Provider, Inc. (TSPi)

By

Cove Software, LLC

Table of Contents

1 Introduction

This document is prepared for Technology Solutions Provider, Inc (TSPi) under subcontractor agreement dated Feb 3, 2021.

This document supplements the other PDF documents found in the 'doc' subdirectory after installing OpenDCS. The purpose here is to describe the specific configurations and enhancements for using the software as part of the USDA WCIS/SNOTEL system.

OpenDCS will be used to:

- Acquire raw DCP (Data Collection Platform) data transmitted over GOES (Geosynchronous Operational Environmental Satellite).
    - Acquire from public servers operated by NOAA (National Oceanic and Atmospheric Administration) and USGS (US Geological Survey).
    - Acquire from a to-be-purchased HRIT (High Rate Information Transfer) satellite receiver system to receive data directly from GOES.
- Provide a short term (90 days typical) archive of raw, unprocessed GOES DCP message data. The "LRGS" module within OpenDCS provides this functionality.
- Decode raw message data into a specific CSV (Comma-Separated Value) file format and place it into a directory for ingest into WCIS.

Because WCIS is already maintaining a database of platform meta-data that includes SNOTEL station IDs, names, and sensor information, it was not desirable to duplicate this information in the OpenDCS "DECODES" database. We have therefore prepared custom software that bypasses the normal decoding functions of OpenDCS and instead uses a flat-file of meta data provided by WCIS.

This document will describe the specific DECODES configuration and database elements that are needed, and the file formats for meta data, control, and output. It is assumed that readers know the basics of OpenDCS, LRGS, and DECODES.

1.1 Document Revisions

Revision 2 May 2021:

- Document rewritten for the custom SNOTEL daemon

Revision 3 June 2021:

- Updates to section 3.4 on History Retrievals to clarify the interpretation of begin and end times.
- Section 3.3 was rewritten to reflect the new data formats supported for SNOTEL.
- New configuration variable outputTmp.

Revision 4 August 2021:

- Minor grammatical corrections and improvements for clarity.
- Section 3.4 on History Directives has been improved for clarity.

Revision 5 August 2021:

- Section 3.3.2 added a description of the reasonableness-check used for format B.

Revision 6 October 2021:

- Improved description of real-time and history retrievals in section 2.
- Improved description of Output Files in section 2.
- Improved description of Log and Status Files in section 2.

2 SNOTEL Daemon Process

A new background process has been added to OpenDCS for this application. Install OpenDCS in the normal manner as described in the installation guide.

The new process is started with the commands:

```
cd $DCSTOOL_HOME
bin/decj decodes.snotel.SnotelDaemon -k snoteldaemon.lock OtherOptions &
```

The "-k snoteldaemon.lock" argument tells the application to use a lock file with the given name. The lock file provides two functions:

- It prevents two instances of the daemon from running at the same time. If you try to start a second, and a currently lock file exists, it will exit with an error message.
- Provides a way to shut down the daemon gracefully: simply remove the lock file.

OtherOptions may include:

- -l *logfilename* (default is "snoteldaemon.log in current directory)
- -d *level* (where level is 1, 2, or 3 from least to most verbose debug level)

The daemon will continually monitor three directories for directives from the control-M web application:

- Control-M Config – contains configuration directives that can modify the current running configuration.
- Control-M RealTime – Updates to the real-time specification platform list are dropped here. The application will pick them up and use them at the next scheduled real-time retrieval.
- Control-M History – History directives are dropped here. The application will process them immediately.

The location of these three directories can be controlled via the configuration file described below.

Real-Time retrievals are run continuously or periodically and provide the normal mechanism for retrieving data as it comes available. There are two configuration variables that control real-time retrievals:

- retrievalFreq – (The variable name is an unfortunate misnomer – it sets an *interval,* not a frequency). Set as an integer number of minutes (default = 60). Real-time retrievals will be started at this interval. Set to 0 (zero) to have real-time retrievals start once and then never terminate.
- realtimeSince – This is a string that determines how much backlog of data to retrieve each time that the real-time retrieval is started. The default is "now – 1 hour 10 minutes". So, if you change retrievalFreq, you should probably modify realtimeSince to a corresponding interval. Note that the meaning of this parameter is consistent even if retrievalFreq is set to 0 (zero). In this case, the one-and-only time that retrieval is started, it will retrieve this amount of backlog.

History Retrievals provide a way to retrieve data that may have been missed by the real-time process. There are a number of scenarios for this:

- A new station was added to the network 10 days ago and you want to process data since the station became operational.
- Something was wrong with the station's configuration and you want to reprocess old data.
- The data acquisition system was down for five hours. It is restarted now, and you want to recover dropped during the outage.

History retrievals are triggered when a file is dropped into the Control-M History directory. The file may contain one or more history directives. Each directive contains a platform to retrieve from and a time range.

Output Files

Output files will be placed into the output directory and will be time tagged as:

- hs-YYYYMMDD-HHMMSS.csv – history retrieval
- rt-YYYYMMDD-HHMMSS.csv – realtime retrieval

As the filename extension implies, these files are CSV (Comma-Separated Value) files. The format of each line is:

```
msgdate,msgtime,stationID,data-date,data-time,chan1,chan2, … chanN
```

Where,

- msgdate and data-date are in the format MM/DD/YYYY
- msgtime and data-time are in the format HH:MM:SS
- msgdate and msgtime denote the date and time that the DCP message was generated by the station and transferred over the GOES satellite network.
- Data-date and data-time denote the date and time of the data samples to follow on this line of data
- stationID is the numeric SNOTEL ID assigned to the station.

Log and Status Files

As the daemon operates, it writes log messages into "snoteldaemon.log" (or other file if you specified one on the command line). This file can be used for troubleshooting and monitoring the running application with the command:

```
tail -f snoteldaemon.log
```

The name of the log file may be changed with the "-l *logfile*" argument on the command line.

Log files are allowed to grow to 10 MB in size. When this is reached, the file is renamed with a ".old" extension, and a new log file is started.

The daemon also continually updates a status file called “snotel.stat”. This file holds name=value pairs, and is intended to allow the process to pickup where it left off when it is started. A typical status file looks like the following:

```
#Snotel Status as of Tue Aug 31 09:40:44 EDT 2021
#Tue Aug 31 09:40:44 EDT 2021
configLMT=1630349301000
realtimeLMT=1629211870000
lastHistoryRun=0
lastRealtimeRun=1630410302636
historyLMT=1623174388000
```

The time values are stored as milliseconds since the epoch of Jan 1, 1970 UTC.

2.1 Configuration

The *master* configuration file for the Snotel Daemon is called “snotel.conf” and will reside in the $DCSTOOL_USERDIR directory. If this is a single-user installation, DCSTOOL_USERDIR and DCSTOOL_HOME will both be set to the directory where you installed OpenDCS.

The “snotel.conf” file is a Java properties file with name=value pairs, one per line. The table below lists the valid parameter names and describes how each is used by the code.

You may leave the settings with a default blank. lrgsUser and lrgsPassword must be provided.

Note that the Control-M Interface described below provides a way to set individual parameters by dropping a file in the controlmConfig directory. The files dropped in the control-M directory are distinct from the master config.

For example, a user might drop a file into the control-M config dir containing just lrgsUser and lrgsPassword. The code will modify the master config file with these parameters, leaving the others alone.

Name	*Data Type*	*Default*	*Description*
controlmConfigDir	String	$DCSTOOL_USERDIR/ controlmConfig	The directory where the software will monitor for configuration directives from the Control-M file interface.
controlmRealtimeDir	String	$DCSTOOL_USERDIR/ controlmRealtime	The directory where the software will monitor for new realtime station list files from the Control-M file interface.
controlmHistoryDir	String	$DCSTOOL_USERDIR/ controlmHistory	The directory where the software will monitor for history retrieval requests from the Control-M file interface.
moveToArchive	Boolean	false	If true, after processing each control-M file, move it to a subdirectory named 'archive' under the directory in which it was found.
fileBufferTime	integer	5	Number of seconds to buffer multiple messages into the same output file.
lrgs1	host:port	nlrgs1.noaa.gov:16003	Host name – colon – port number for the first (preferred) LRGS from which to receive data
lrgs2	host:port	cdabackup.wcda.noaa.gov :16003	Second (1st backup) LRGS
lrgs3	host:port	lrgseddn1.cr.usgs.gov :16003	Third (2nd backup) LRGS
lrgs4	host:port	nlrgs2.noaa.gov:16003	Fourth (3nd backup) LRGS
lrgsUser	String	None – must be supplied	The user name for connecting to the LRGS systems.
lrgsPassword	String	None – must be supplied	The password for connecting to the LRGS systesm.
outputDir	String	$DCSTOOL_USERDIR/ snotel-out	The directory to place decoded output data files.
realtimeSince	String	now – 1 hour 10 minutes	Each time the real time retrieval is run, how much data to retrieve. Usually set to the retrieval frequence plus a small amount.
outputTZ	String	GMT-08:00	Time Zone for output data. This also controls the interpretation of dates in a history retrieval file.
outputTmp	String	$DCSTOOL_USERDIR/tmp	Output files are built here and then moved to outputDir when complete.
retrievalFreq	Integer	60	Number of minutes. Execute the regular retrievals at this interval. Set to zero for real-time, i.e. stay connected to LRGS and process new messages as they arrive.

3 Control Interface

TSPi is writing code that will control OpenDCS by dropping files into three directories that OpenDCS will monitor. It is understood that OpenDCS access to these directories may be read-only. OpenDCS will keep track of the last-modify-time (LMT) each time it processes a file in these directories. The LMT will be used to avoid re-processing old files. It is assumed that Control-M will eventually delete old/obsolete files.

These 3 directories that are controlled by control-M are:

- Config – files dropped here contain various configuration directives
- History – files dropped here are directives telling OpenDCS to do a retrieval of historical data.
- RealTime – Only the newest (greatest LMT) file in this directory is used. It will be copied and will replace the station list being used by the real-time routing spec.

3.1 Don't Process Old Control-M Files

As files are processed from the control-m directories, the daemon keeps track of the last modify time (LMT) of each file seen. It will only process files that are newer than the last one seen. The LMT for each type of file (config, realtime, history) is kept in memory as the daemon runs. When the daemon exits, it stores the LMTs in the "snotel.stat" file so that when it is restarted, it will not reprocess files remaining in the directory.

If you want the software to process an old file, you have two options:

- Use the Unix 'touch' command on the file. This update's the file's LMT.
- Stop the daemon and edit the "snotel.stat" file. If you delete the stored LMT, then when you restart the daemon it will process all files in that directory.

The configuration variable 'moveToArchive' is Boolean (true/false). If you set it to true, then after each file is processed, it will be moved to a subdirectory under the CONTROL-M directory named 'archive'.

- Setting moveToArchive to true also disables the LMT processing. All files in the directories will be processed.

3.2 Configuration Updates

CONTROL-M can modify the daemon's configuration by dropping files into the CONTROL-M Configuration directory. These dropped files may contain any of the configuration variables defined above.

NOTE: These files dropped into the CONTROL-M Config Directory are Separate from the master configuration file described above. The daemon will read the files dropped into the control-m directory, incorporate them into the master, and then update the master configuration file.

For example, if you just want to set the LRGS user and password, drop a file into the config directory with the contents:

```
lrgsUser=joe_user
lrgsPassword=SomethingVerySecure
```

3.3 Real-Time Spec File Updates

A.k.a "REGULAR STATION LIST" files.

When the daemon sees a new real-time spec file, it copies it into $DCSTOOL_USERDIR in the name specified by the configuration. This *replaces* the previous regular station list completely.

A Regular Station List file is a CSV (Comma Separated Value) file with the following fields:

- SNOTEL Station ID (integer)
- Station Name (free-form string, but may not contain comma)
- GOES DCP Address (8 hex digits)
- Data Format Type (A or B – see below)

3.3.1 Format A

Data Format 'A' is for legacy stations. All the data is to be considered for the same hour. No limit on number of channels.

Format A messages may be in ASCII, but the norm is pseudo-binary. The code will scan the message to determine if it is ASCII or pseudo-binary.

For pseudo-binary, each value is a 3-character Campbell Pseudo-binary number. See Campbell Scientific's documentation for details on how floating-point numbers are encoded into 3 printable characters.

For ASCII, values are ASCII numbers separated by commas.

3.3.2 Format B

Data Format 'B' is for new stations. The first two 3-character pseudo-binary numbers are integers representing:

- The number of hours covered by the message
- The number of channels in each hour.

Each hour then forms a block of data containing:

- 3-character pseudo-binary integer day-of-year
- 3-character pseudo-binary integer representing HHMM. Example: the number 429 represents hour 4, minute 29.
- *N* 3-character pseudo-binary floating-point values representing the data channels.

Processing stops when the specified number of hours and channels are reached or when the end of the message is reached.

Aug 30, 2021: A reasonableness-check has been added for format B as follows:

- The number of hours must be in the range [1 … 5]
- The number of hours multiplied by the number of channels must be in the range [1 … 375]

If either of these checks fails, the code will attempt to decode using format A.

3.4 History Directives

History directives are CSV files with the following columns:

- Station number
- Station name – may contain spaces but no commas.
- GOES DCP Address
- Start date in format MM/DD/YYYY
- End date in format MM/DD/YYYY

Multiple directives may be placed in a single file. They will be processed in order.

Example of history directive file:

```
385,CARROT BASIN,DA50627A,A,05/21/2021,05/22/2021
393,CHALK CREEK #2,DA558648,A,05/21/2021,05/22/2021
399,CLEAR CREEK #1,DA533BDA,A,05/21/2021,05/22/2021
402,CLOUD PEAK RESERVOIR,DA55132A,A,05/21/2021,05/22/2021
```

The configuration variable ‘outputTZ’ is used to interpret the dates. The end date is inclusive (i.e. the entire day will be retrieved.) These start/end times are applied to DCP message times when retrieving data from an LRGS.

Example:

- If start is 05/21/2021, then all messages since midnight on 5/21/2021 in the timezone specified by outputTZ will be retrieved.
- If end is 5/22/2021, then all messages until 5/21/2022 23:59:59 in the timezone specified by outputTZ will be retrieved.