BEAM

Blacknest Data System (BDS)

BdsTimestampUpdate Program – 2.2.7 - 2021-08-25

1. Introduction

The bdsTimestampUpdate program is designed to update the sensor data block timestamps on a previously imported set of data. This may be required when it is determined that the GPS timestamp generation system has failed on an instrument.

In order to run the program needs the startTime, endTime and list of Network:Station:Channel:Source data selectors. It then needs the time offset supplied as a signed floating point number in seconds or as a time to modify the data start time to. The time offset has a resolution of 1 microsecond. By default the program will list the files it will modify and check it can do so. To actually perform the operation the "-doIt" flag has to be given.

There is also the "-trial" flag that will actually update the database metadata in a transaction and process all of the data files to temporary files to check fully at all works. The database transaction is aborted leaving the database as it was. Note the temporary files are left in a "tmp" directory within each affected DataStore directory. If a final -doIt command is run, these temporary files will be removed.

The program accesses the BdsServer via its API to perform the necessary Metadata queries and updates. It also needs direct access to the actual BDS format data files so it can update their contents. This means the program will have to be run as the "bds" user and on the BDS server host.

2. Usage

The bdsTimestampUpdate program can be run by any user that has permissions to access the BDS system, but only the "bds" user will be able to actually update the timestamps and then only when run on the BDS server itself. The program accepts the following command line options:

-help	Help on command line parameters
-v	Verbose operation
-host <hostname>	BDS Server host name
-user <user:password>	The BDS user id and password
-startTime <time>	The StartTime. "YYYY-MM-DDTHH:MM:SS.000000"
-endTime <time>	The EndTime
-select <network:station:channel:source>	Select a data set. Can have multiple -select's
-timeOffset <seconds>	Offset the timestamps of all blocks within the selection by the number of seconds given. The seconds is given as a floating point number and has a resolution of 1 microsecond.
-timeTo <time>	Offset the timestamps of all blocks within the selection by the using the offset calculated from startTime to timeTo.
-trial	Actually perform the file updates to temporary files and database changes, but don't commit the changes.
-doIt	Actually perform the update. By default the program will just list what it will do and check that it can do this.

-f <filename>	If this option is provided the program will just process the given file and create an updated file in the filename given by the -o parameter.
-o <filename>	This defines the output filename for the “-f” raw file processing mode.

The bdsTimestampUpdate program will read the BDS_HOST environment variable at start-up. This variable, if set, defines the default BdsServer host name to contact. The default is “localhost” if this is not set.
The “-startTime” and “-endTime” options define a period of time. The times for these arguments should be given in ISO 8601 date time format. Examples of acceptable values include: “2008-11-03T10:00:00.000000”, “2008-11-03”, “2008-11-03T10:00:00”. Multiple “-select” options are allowed. Each of these define a Network, Station, Channel and Source set. A null entry for any of these parameters is taken as meaning any value. You can also use regular expressions to define the fields.
Generally the bdsTimestampUpdate program outputs the results on the stdout stream using a comma separated value (CSV) scheme.

By default the program will only list what it will do. The -doIt option tells it to actually perform the update. When performing the actual update the program will first try and update all of the database Metadata in a single transaction and then process the files to a specially created “tmp” directory in each of the BDS data store’s affected directories. Only when all of this succeeds will the database transaction be committed and the newly created files in the tmp directories moved over the existing BDS data files. The old files will be moved to the Deleted directory.

The program will only work with BDS data files in the channel multiplexed format. It will abort with an error if asked to update a sample multiplexed file.

The program updates the sensor data files selected, the DataChannel Metadata start and end times and the DataFiles start, endTimes and url. Thus it changes the times for file name on the associated DataFile metadata and the file names of the BDS sensor data files themselves. It also does not update any associated sensor data file internal string based info that might mention a time or any of the BDS Metadata’s Notes that might be set to a particular data range.

As well as updating the sensor data on a BDS server, the program can simply update a sensor data file in BDS format creating a new BDS format file with the timestamps changed for the channels defined. The -f <filename> with the -o <filename> command line options provide this.

3. Return Value

The program will return a status value of 0 if all was Ok. It will return a non zero value on error, the BDS error number, together with a message output on stderr.

4. Further Information

For further information please look at the BDS system documentation at:
https://portal.beam.ltd.uk/support/blacknest.