Data Playback Utilities

Version 3.8

Reference Guide

Nanometrics Inc.
Kanata, Ontario
Canada

Data Playback Utilities Version 3.8 Reference Guide

This product uses the Java™ 2 Runtime Environment (J2RE) Standard Edition, which includes code licensed from RSA Security, Inc. Some portions licensed from IBM are available at http://oss.software.ibm.com/icu4j/

Nanometrics, Inc.
250 Herzberg Road
Kanata, Ontario, Canada K2K 2A1
Tel (613)592-6776
Fax (613)592-5929
Email info@nanometrics.ca

Part number 14548R1 2002-10-30

Name

DECIM

Version

2.3

Synopsis

decim [-i input_file] [-o output_path] [-f filter_file]

Options

-i input_file	Input file path of the Y data files. It may contain wildcard characters, e. g., Y1998*, and it can contain absolute or relative directory path to the files. For example, the argument can be c:\data\Y1998*, or ..\data\Y1998*.
output_directory	Output directory path in which the program places the decimated Y data files (e.g. c:\data\newyfiles).
-f filter_file	Path and file name of the FIR filter. For example, c:\data\filter\fir1.dat.

Files

FIR filter files

DECIM requires filter information when decimating a time series into a new one with different sampling rate. The basic format of the file is:

1. (int) number of stages
2. (int) number of coefficients, and (int)decimation for stage i
3. (float) coefficients

Lines 2 and 3 are the values defining each stage, and therefore, they repeat depending on the number of stages given in line 1. A sample FIR filter file, d4fir.dat, which can be used to decimate data by 4, is provided in the user directory of the release.

Environment

DECIM runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.7 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\decim.fm
Date last revised: 2002-10-30

Name

EXTRACT CLIENT

Version

1.2

Description

Extract Client retrieves time series data from the ringbuffer files on the central acquisition computer. This is a firewall-compliant remote extraction service; Extract Client sends the extract request to the Extract Server program (running on the central acquisition machine) via TCP socket. The Extract Server extracts the requested data from the ringbuffers using Extractp, then compresses the files and returns them via FTP to the Extract Client machine. The current version uses zip for file compression.

Overview

Extract Client is run from the Windows command line using the following syntax:

extractclient host port [-r] type start -d d1 [-i i1|-c c1] -o o1 [options]

host Name of the machine where Extract Server runs.

Format: www.xxx.yyy.zzz
e.g. 199.71.138.01 or nmx.nanometrics.ca

port Port number of Extract Server

-r Optional callback flag.

When used, this flag indicates that Extract Client should use function in callback mode.

start Start time of data.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1998-10-17-23-11-17

type Type of data to extract

-x for x-files

-y for y-files

-d for decimated y-files

-s for soh files

-d Duration of data to extract in units of seconds.

-i Optional input ringbuffer(s) name.

e.g. -i “STN01.BHZ”

Extract data from the ringbuffer STN01.BHZ on the Extract Server machine. Extract Server searches the directories indicated in its configuration file for ringbuffers of that name. The ringbuffer name MUST BE ENCLOSED IN QUOTES.

Wildcards are allowed in the ringbuffer's filename as a way of extracting data from more than one ringbuffer or being more general about the ringbuffer names.

e.g. -i “*.BHZ”

This option cannot be used in conjunction with the -c option.

-c Input channel list file name.

e.g. -c “EXTRACT.LST”

Extract data from all the ringbuffers indicated in the file EXTRACT.LST. Extract Server searches the directories listed in its configuration file for a channel list file of that name. The channel list file name MUST BE ENCLOSED IN QUOTES.

The format of the channel list file is a file containing a list of filenames corresponding to the different ringbuffer channels, separated by either blanks or new lines.

See the -i option for how to specify a ringbuffer filename.

An example of a channel list file is as follows:

RINGBUFF\ROTT.BH*

D:\DSS\RINGBUFF\R*.BHZ

This option cannot be used in conjunction with the -i option.

-o Output Directory name.

e.g. -o user\extract

Return the extracted files via FTP to the directory with the FTP alias “user\extract”. If the specified directory is not accessible via FTP, the compressed output files will not be delivered. Extract Client will not be able to detect that this error had occurred.

FTP aliases are specified in the configuration of the FTP server. The Microsoft FTP Service requires the administrator to indicate which directories on the hard drive are accessible for FTP reads or writes. Any directory added to the list of accessible directories is given an alias. For instance, the administrator may to choose to alias the directory “d:\user\extract” as “extract”. Then to have Extract Client return it’s output to the directory “d:\user\extract\tuesday”, the user must specify “extract\tuesday” as the output directory.

options Optional arguments for specific extract types

For type -d:	-f FilterFileName (required)
For type -s:	[-l logFile | -f | -n | -u | -m | -w | -b | -c | -g | -t | -v | -x | -r | -h | -k | -i | -p | -a | -q | -3 | -4] (optional). See sohextrp.exe for details.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\extractclient.fm
Date last revised: 2002-10-30

Name

EXTRACTP

Version

4.15

Description

EXTRACTP retrieves time series data from the ringbuffer files and creates one file (or more) per component in the case of multi-component extracts. The output file is in compressed X5 file format. The text below is derived from the extractp on-line help commands. EXTRACTP operates in one of five modes as listed below.

Overview of Modes

extractp -m | -a | -d | -s | -e

-a Extract All required switch
-m Manual mode required switch
-d Debug mode required switch
-s Summary mode required switch
-e Event mode required switch

-a Extract All mode.

Extract All mode is the simplest method for extracting data from the ringbuffers. In this mode, EXTRACTP extracts all data from the specified ringbuffers into X-files. It is not necessary to specify a start time and duration. To extract a subset of the data, use the -m option.

For help on syntax, type	"EXTRACTP -a -h"
For full help, type	"EXTRACTP -a -h full"
For help on a specific argument, type	"EXTRACTP -a -h arg"

-m Manual mode.

Manual mode is the normal method for extracting data from the compressed ringbuffers. The command line arguments specify the start and the end times of the request and EXTRACTP extracts the data in between and writes the data to an X-file. Manual mode always includes the last partial block. It rounds upwards the final block number requested.

For help on syntax, type	"EXTRACTP -m -h"
For full help, type	"EXTRACTP -m -h full"
For help on a specific argument, type	"EXTRACTP -m -h arg"

-d Debug mode.

In debug mode EXTRACTP dumps key fields in the header of each compressed data block in the ringbuffer to a file in ASCII format. This information consists of the start time, sample count, sample rate, and header index number for all blocks in the ringbuffer.

For help on syntax, type	"EXTRACTP -d -h"
For full help, type	"EXTRACTP -d -h full"
For help on a specific argument, type	"EXTRACTP -d -h arg"

-s Summary mode.

In summary mode EXTRACTP provides a summary after searching through a ringbuffer. The summary contains time duration, number of samples and time duration of each continuous data section, time gap between this data section and the previous one.

For help on syntax, type	"EXTRACTP -s -h"

For full help, type	"EXTRACTP -s -h full"
For help on a specific argument, type	"EXTRACTP -s -h arg"

-e Event mode.

In event mode EXTRACTP retrieves the data from the ringbuffer files based on the time and duration listed in the given event file, e.g., naqs.elf. It writes the time series data of the events in X5 files, and optionally, put them under subdirectories named after event time.

For help on syntax, type	"EXTRACTP -e -h"
For full help, type	"EXTRACTP -e -h full"
For help on a specific argument, type	"EXTRACTP -e -h arg"

Usage

When specifying pathnames, a leading slash is interpreted as an absolute pathname while absence of a leading slash implies a relative pathname or a filename. Note that the slash used in a Windows system is a backslash while that in a Solaris system is a forward slash.

Manual Mode

-m Manual mode.

extractp -m Arg2 [-e e1 | -d d1] [-i i1|-c cl] [-o o1] [-x x1] [-f] [-t t1] [-p p]

Arg1	-m required switch
Arg2	StartTime required argument (string)
-e	EndTime optional switch
e1	EndTime required switch argument (string)
-d	Duration optional switch
d1	Duration required switch argument (integer)
-i	Input optional switch
i1	RngBufPathname required switch argument (string)
-c	Channel list optional switch
c1	Channel list file required switch argument (string)
-o	Output optional switch
o1	OutputPath required switch argument (string)
-x	Extension optional switch
x1	ExtNum required switch argument (integer)
-f	FullName optional switch
-t	Tolerance optional switch
t1	Tolerance optional switch argument (floating point)
-p	Output X-file in pieces switch
p1	Time duration in minutes of each piece in an X-file

Arg2 Start time of data.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1999-10-17-23-11-17

-e End time of data.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1999-10-17-23-16-00

-d Duration of data to extract in units of seconds.

-i Input ringbuffer(s) pathname.

e.g. -i D:\DSSOS2\RINGBUFF\OTTSPZ.RBF

Extract data from the ringbuffer D:\DSSOS2\RINGBUFF\OTTSPZ.RBF.

The ringbuffer pathname must be the ringbuffer's absolute pathname or pathname relative to the current working directory.

Wildcards are allowed in the ringbuffer's filename as a way of extracting data from more than one ringbuffer or being more general about the ringbuffer names.

The default is "RINGBUFF*".

This option cannot be used in conjunction with the -c option.

-c Input channel list file pathname.

e.g. -c D:\DSSOS2\EXTRACT.LST

Extract data from all the ringbuffers indicated in the file D:\DSSOS2\EXTRACT.LST. The channel list file pathname must be the file's absolute pathname or pathname relative to the current working directory.

The format of the channel list file is a file containing a list of filenames corresponding to the different ringbuffer channels, separated by either blanks or new lines.

See the -i option for how to specify a ringbuffer filename.

An example of a channel list file is as follows:

```
RINGBUFF\OTTSPZ.RBF
D:\DSSOS2\RINGBUFF\NMX*.RBF
```

This option cannot be used in conjunction with the -i option.

-o Output path of extracted data.

In manual and block mode the output is X-files. In debug mode the output is ASCII debug files.

e.g. -o D:\DAN\XDATA\

Direct all output to the directory D:\DAN\XDATA. The default is XDATA\.

-x Append a user supplied extension to all extracted files.

e.g. -x 017

This appends the extension ".017" to all extracted files. The default is an internally generated number that is incremented each time EXTRACT is called. This only applies when using the default short filenames option. See also -f

-f Use full filenames.

Extracted X-files are named using the following convention:

Xchannel_yyyymmdd.hhmmss

The default X-file naming convention omits the date and time and names the files using the following convention:

Xchannel.nnn

This option requires the output drive (in Windows) to support long file names. See also -x.

-t Tolerance in seconds

e.g. -t 0.05

This is the time tolerance between two compressed data blocks. For every compressed data block extracted from the ringbuffer EXTRACTP calculates the time gap between its start time and the time of the last sample of the previous block plus 1/sample-rate second. If the time gap matches or exceeds the given tolerance, a new X-file is created. The default is 0.010 seconds.

-p Set the time duration (in minutes) of each piece in a X- file.

e.g. -p 30 5

The two numbers after the -p switch define the time duration of each trace and the amount of overlap in the two consequative traces, respectively. If the requested time duration is longer than the value given after -p, traces with similar length are extracted to different sub directories. The name of the sub directory is yyyymmddhhmm. For example, if total time requested is from 1999-06-20_02:15:30.00 to 1999-06-20_03:40:00.00 and p is 30, the sub directories are 19990620200215, 19990620200230, 19990620200300, and 19990620200330. The trace under each subdirectory contains data blocks of time duration equal to or longer than 30 minutes, depending on how data blocks are created. On the other hand, if the second time parameter is not zero (5 minutes in this case), the traces under each subdirectory contains data blocks of time duration equal to or longer than 35 minutes.

-v Set the verbosity of the messages given by the program

e.g. -v 63488

The following verbosity masks are defined:

CRITICAL	0x8000	32768	system call failed
FATAL	0x4000	16384	kill the program
ERROR	0x2000	8192	stop this function
WARNING	0x1000	4096	continue but may be wrong
NOTICE	0x0800	2048	no problem but of interest
VERBOSE	0x0400	1024	Verbose messages
DEBUG	0x0200	512	Debug messages

The code to be specified with the -v switch is the sum of the codes of different levels you want to specify. For example if you want to set extractp to display all messages with ERROR or NOTICE or DEBUG levels the number you should use is: 8192+2048+512=10752. A messages is displayed if the result of ANDing the message verbosity with the verbosity mask is non-zero. The default mask is:

CRITICAL | FATAL | ERROR | WARNING | NOTICE = 63488.

If the verbosity mask is not specified by the user the program will use the default one.

Usage

The new X-file name depends on whether the -f option is used. If the -f option is used (long filenames) the new X-file name is Xchannel_yyyymmdd.hhmmss where the date/time is now the date/time of the first sample in the new file. If -f is not specified the new X-file is named by incrementing the extension number.

See Also

The section “Usage” on page 6.

Extract All Mode

Extract All mode is identical to Manual mode except that the StartTime, EndTime, and Duration parameters are not required.

Debug Mode

-d Debug mode.

extractp -d [-i i1|-c c1] [-o o1]

Arg1	-d required switch
-i	Input optional switch
i1	RngBufPathname required switch argument (string)
-c	Channel list optional switch
c1	Channel list file required switch argument (string)
-o	Output optional switch
o1	OutputDir required switch argument (string)

-i Input ringbuffer(s) pathname.

e.g. -i D:\DSSOS2\RINGBUFF\OTTSPZ.RBF

Extract data from the ringbuffer D:\DSSOS2\RINGBUFF\OTTSPZ.RBF.

The ringbuffer pathname must be the ringbuffer's absolute pathname or pathname relative to the current working directory.

Wildcards are allowed in the ringbuffer's filename as a way of extracting data from more than one ringbuffer or being more general about the ringbuffer names.

The default is "RINGBUFF*".

This option cannot be used in conjunction with the -c option.

-c Input channel list file pathname.

e.g. -c D:\DSSOS2\EXTRACT.LST

Extract data from all the ringbuffers indicated in the file D:\DSSOS2\EXTRACT.LST. The channel list file pathname must be the file's absolute pathname or pathname relative to the current working directory.

The format of the channel list file is a file containing a list of filenames corresponding to the different ringbuffer channels, separated by either blanks or new lines.

See the -i option for how to specify a ringbuffer filename. An example of a channel list file is as follows:

```
RINGBUFF\OTTSPZ.RBF
D:\DSSOS2\RINGBUFF\NMX*.RBF
```

This option cannot be used in conjunction with the -i option.

-o Output path of extracted data.

In manual and block mode the output is X-files; in debug mode the output is ASCII debug files.

e.g. -o D:\DAN\XDATA\

Direct all output to the directory D:\DAN\XDATA\. The default is XDATA\.

-v Set the verbosity of the messages given by the program

e.g. -v 63488

The following verbosity masks are defined:

CRITICAL	0x8000	32768	system call failed
FATAL	0x4000	16384	kill the program
ERROR	0x2000	8192	stop this function

WARNING	0x1000	4096	continue but may be wrong
NOTICE	0x0800	2048	no problem but of interest
VERBOSE	0x0400	1024	Verbose messages
DEBUG	0x0200	512	Debug messages

The code to be specified with the -v switch is the sum of the codes of different levels you want to specify. For example if you want to set extractp to display all messages with ERROR or NOTICE or DEBUG levels the number you should use is: 8192+2048+512=10752. A messages is displayed if the result of ANDing the message verbosity with the verbosity mask is non-zero. The default mask is:

CRITICAL | FATAL | ERROR | WARNING | NOTICE = 63488.

If the verbosity mask is not specified by the user the program will use the default one.

See Also

The section "Usage" on page 6.

Summary Mode

-s Summary mode.

extractp -s [-i i1 | -c c1] [-o o1] [-t t1]

Arg1	-s required argument (string)
Arg2	StartTime optional argument (string)
-d	Duration optional switch
d1	Duration required switch argument (integer)
-i	Input optional switch
i1	RngBufPathname required switch argument (string)
-c	Channel list optional switch
c1	Channel list file required switch argument (string)
-v	Verbosity optional switch
v1	Verbosity mask optional switch argument (integer)
-t	Tolerance optional switch
t1	Tolerance optional switch argument (floating point)

Arg2 Optional start time.

e.g. 2000-02-07-12-00-00

Optionally specify the start time for the summary in format yyyy-mm-dd-hh-mm-ss.

If this field is omitted, the summary starts at the beginning of the ringbuffer.

-d Duration.

e.g. -d 3600

Optionally specify the duration in seconds for the summary report. If this field is omitted, the report will cover from the start time to the end of the ringbuffer.

-i Input ringbuffer(s) pathname.

e.g. -i D:\DSSOS2\RINGBUFF\OTTSPZ.RBF

Extract data from the ringbuffer D:\DSSOS2\RINGBUFF\OTTSPZ.RBF.

The ringbuffer pathname must be the ringbuffer's absolute pathname or pathname relative to the current working directory.

Wildcards are allowed in the ringbuffer's filename as a way of extracting data from more than one

ringbuffer or being more general about the ringbuffer names.

The default is "RINGBUFF*".

This option cannot be used in conjunction with the -c option.

-c Input channel list file pathname.

e.g. -c D:\DSSOS2\EXTRACT.LST

Extract data from all the ringbuffers indicated in the file D:\DSSOS2\EXTRACT.LST. The channel list file pathname must be the file's absolute pathname or pathname relative to the current working directory.

The format of the channel list file is a file containing a list of filenames corresponding to the different ringbuffer channels, separated by either blanks or new lines.

See the -i option for how to specify a ringbuffer filename.

An example of a channel list file is as follows:

RINGBUFF\OTTSPZ.RBF
D:\DSSOS2\RINGBUFF\NMX*.RBF

This option cannot be used in conjunction with the -i option.

-t Tolerance in seconds

e.g. -t 0.05

This is the time tolerance between two compressed data blocks. For every compressed data block extracted from the ringbuffer EXTRACTP calculates the time gap between its start time and the time of the last sample of the previous block plus 1/sample-rate second. If the time gap matches or exceeds the given tolerance, a new X-file is created. The default is 0.010 seconds.

-v Set the verbosity of the messages given by the program

e.g. -v 63488

The following verbosity masks are defined:

CRITICAL	0x8000	32768	system call failed
FATAL	0x4000	16384	kill the program
ERROR	0x2000	8192	stop this function
WARNING	0x1000	4096	continue but may be wrong
NOTICE	0x0800	2048	no problem but of interest
VERBOSE	0x0400	1024	Verbose messages
DEBUG	0x0200	512	Debug messages

The code to be specified with the -v switch is the sum of the codes of different levels you want to specify. For example if you want to set extractp to display all messages with ERROR or NOTICE or DEBUG levels the number you should use is: 8192+2048+512=10752. A messages is displayed if the result of ANDing the message verbosity with the verbosity mask is non-zero. The default mask is:

CRITICAL | FATAL | ERROR | WARNING | NOTICE = 63488.

If the verbosity mask is not specified by the user the program will use the default one.

Event Mode

-e Event mode.

extractp -e [-l l1] [-s s1] [-e e1] [-i i1|-c cl]] [-o o1] [-x x1] [-f] [-d] [-t t1]

Arg1	-e required switch
-l	Event list optional switch
l1	Event list pathname required switch argument (string)
-s	Pre event optional switch
s1	Pre event time required switch argument (integer)
-e	Post event optional switch
e1	Post event time required switch argument (integer)
-i	Input optional switch
i1	RngBufPathname required switch argument (string)
-c	Channel list optional switch
c1	Channel list file required switch argument (string)
-o	Output optional switch
o1	OutputPath required switch argument (string)
-x	Extension optional switch
x1	ExtNum required switch argument (integer)
-f	FullName optional switch
-d	Create directory optional switch
-t	Tolerance optional switch
t1	Tolerance optional switch argument (floating point)

-l Event list file name

Default is EVENTS\NAQS.elf

e.g. -l myevents.elf

The text file is generated by NAQSP, and can also be generated by a user manually. Each line in the file must be exactly 80characters long, and contains fields of type, date, starting time, and duration of the event in the following format:

```
                1       2       3       4       5       6

(column 12345678901234567890123456789012345678901234567890123456789012345678901234567890)

        E       Thu Sep 18 09:13:55 1996        *       00112
```

-s Pre event time in seconds

e.g. -s 5

It defines the starting time of the trace, which is 5 seconds before the starting time of the event. The default pre event time is 5 seconds.

-e Post event time in seconds

e.g. -e 10

It defines the ending time of the trace, which is 10 seconds after the starting time plus the duration. The default post event time is 10 seconds. Based on the event information given in the event file listed above, the extracted trace has the starting time approximately at 09:13:50, and has a time duration of at least 127 seconds.

-i Input ringbuffer(s) pathname.

e.g. -i D:\DSSOS2\RINGBUFF\OTTSPZ.RBF

Extract data from the ringbuffer D:\DSSOS2\RINGBUFF\OTTSPZ.RBF.

The ringbuffer pathname must be the ringbuffer's absolute pathname or pathname relative to the current working directory.

Wildcards are allowed in the ringbuffer's filename as a way of extracting data from more than one ringbuffer or being more general about the ringbuffer names.

The default is "RINGBUFF*".

This option cannot be used in conjunction with the -c option.

-c Input channel list file pathname.

e.g. -c D:\DSSOS2\EXTRACT.LST

Extract data from all the ringbuffers indicated in the file D:\DSSOS2\EXTRACT.LST. The channel list file pathname must be the file's absolute pathname or pathname relative to the current working directory.

The format of the channel list file is a file containing a list of filenames corresponding to the different ringbuffer channels, separated by either blanks or new lines.

See the -i option for how to specify a ringbuffer file name.

An example of a channel list file is as follows:

RINGBUFF\OTTSPZ.RBF

D:\DSSOS2\RINGBUFF\NMX*.RBF

This option cannot be used in conjunction with the -i option.

-o Output path of extracted data.

In manual and block mode the output is X-files, in debug mode the output is ASCII debug files.

e.g. -o D:\DAN\XDATA\

Direct all output to the directory D:\DAN\XDATA. The default is XDATA\.

-x Append a user supplied extension to all extracted files.

e.g. -x 017

This appends the extension ".017" to all extracted files. The default is an internally generated number that is incremented each time EXTRACTP is called. This only applies when using the default short file names option. See also -f

-f Use full file names.

Extracted X-files are named using the following convention.

Xchannel.yyyymmdd.hhmmss

The default X-file naming convention omits the date and time and names the files using the following convention.

Xchannel.nnn

This option requires the output path to be a HPFS formatted drive. See also -x.

-d Create directory for each event.

Extracted X-files are placed in a separate directory for each event, named using the following convention.

Dyyyymmdd.hhmmss

The default is to place all extracted X-files in the same directory.

This option requires the output path to be a HPFS formatted drive. See also -f

-t Tolerance in seconds

e.g. -t 0.05

The time tolerance between two compressed data blocks. For every compressed block extracted from the ringbuffer EXTRACTP calculates the time gap between its start time and the time of the last sample in the previous block plus 1/sample-rate. If the time gap matches or exceeds the given tolerance a new X-file is created. The default is 0.010 seconds.

-v Set the verbosity of the messages given by the program

e.g. -v 63488

The following verbosity masks are defined:

CRITICAL	0x8000	32768	system call failed
FATAL	0x4000	16384	kill the program
ERROR	0x2000	8192	stop this function
WARNING	0x1000	4096	continue but may be wrong
NOTICE	0x0800	2048	no problem but of interest
VERBOSE	0x0400	1024	Verbose messages
DEBUG	0x0200	512	Debug messages

The code to be specified with the -v switch is the sum of the codes of different levels you want to specify. For example if you want to set extractp to display all messages with ERROR or NOTICE or DEBUG levels the number you should use is: 8192+2048+512=10752. A messages is displayed if the result of ANDing the message verbosity with the verbosity mask is non-zero. The default mask is:

CRITICAL | FATAL | ERROR | WARNING | NOTICE = 63488.

If the verbosity mask is not specified by the user the program will use the default one.

Usage

The new X-file name depends if the -f option is used. If the -f option is used (long file names) the new X-file name is Xchannel.yyyymmdd.hhmmss where the date/time is now the date/time of the first sample in the new file. If -f is not specified the new X-file is named by incrementing the extension number.

See also

The section “Usage” on page 6.

Environment

EXTRACTP runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.8 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\extractp.fm
Date last revised: 2002-10-30

Name

EXTRACT SERVER

Version

1.2

Synopsis

extractserver [iniFile]

Description

Extract Server is a firewall-compliant data extraction service which runs alongside NAQS on the data-acquisition computer. Extract Server listens on a TCP socket for connections and data requests from the client program ExtractNT. Extract Server can be configured to accept either direct connections from the client, or to function in callback mode. In callback mode, whenever Extract Server receives a client connection it closes the socket immediately, and then initiates a new connection to the client. This callback feature is designed to satisfy the security requirement that all socket connections be initiated from inside the firewall.

When it receives a request, Extract Server first runs extractp or sohextrp to extract the required data from the ringbuffers, then calls x5decomp and decim if necessary, then compresses the files and returns them to the client via FTP.

If Extract Server is started via Watchdog, it will ping watchdog periodically to indicate it is functioning well.

Options

inifile the path to an Extract Server ini file. If no inifile parameter is specified, Extract Server looks for a file called Extract Server.ini in its working directory. The format of the ini file is described below.

Running Extract Server

Starting Extract Server

Extract Server may be started manually from the command line, or automatically by the watchdog. To start from the watchdog, simply add an entry for Extract Server to the Watchdog.ini file. To start Extract Server from the command line, type

ExtractServer or **ExtractServer *inifile***

where *inifile* is the path to an Extract Server configuration file.

Stopping Extract Server

It is important that Extract Server be shutdown properly in order for the application to release its system resources. To stop Extract Server properly, type

quit <enter>

in the Extract Server window**.** If Extract Server is shutdown by any other method, temporary files and directories may not be deleted, and spawned processes may not be terminated.

Definition of Inifile Parameters

The Extract Server configuration (ini) file contains the following two section sections :

[Extract Server]

This section defines the general working parameters for Extract Server.

Port = 1213

Definition: the TCP port where Extract Server listens for connections.

Verbosity = DEBUG

Definition: indicates the level of detail for the messages Extract Server writes to its log file. The options are:

DEBUG
INFO
WARNING
ERROR
FATAL

Specifying a verbosity level indicates that Extract Server will print out messages of that verbosity level and higher. For example,

Verbosity = WARNING

indicate that Extract Server will print WARNING, ERROR, and FATAL messages.

Timeout = 300

Definition: the number of seconds Extract Server should wait for one of its sub processes (extractp, decim, zip, etc) to return before concluding that the sub processes is not responding. When Timout seconds have passed without a sub process returning, Extract Server terminates the sub process, and notifies the client that an error has occurred.

Callback = TRUE

Definition: indicates whether or not Extract Server should function in callback mode. The possible values are TRUE or FALSE.

Directories = 2

Definition: the number of directories in which Extract Server is to look for ringbuffer files.

[Directories]

This section lists the path names of the directories in which Extract Server is to look for ringbuffer files. The value given for the Directories parameter in the section above must equal the number of entries in this section.

RingBuffDir = c:\nmx\user\ringbuff

Definition: the pathname of a directory containing ring buffer files.

Example of Inifile

```
[ExtractServer]
Port = 8189
Verbosity = INFO
Timeout = 300
Callback = FALSE
Directories = 3

[Directories]
```

RingBuffDir = c:\nmx\user\ringbuff
RingBuffDir = e:\ringbuff
RingBuffDir = d:\user\catherine\data

Files

extractp.exe
sohextrp.exe
x5decomp.exe
decim.exe

See Also

ExtractNT; Watchdog in the Software Reference Manual

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\extractsrv.fm
Date last revised: 2002-10-30

Name

HEADEDIT

Version

1.02

Synopsis

headedit [filespec] [-t][-d][-i] [inifile]

Description

Headedit is a utility program to edit the Y5 header of binary seismic data files (X5 files, Y5 files, and packet ringbuffers) in order to insert or change missing or incorrect information. Other files can not be modified by Headedit. Fields to be updated include response file name, station, network and site names, comment, latitude, longitude, elevation, instrument sensitivity, frequency, units and calibration units, etc. Only the values of specified fields are modified; other data are kept unchanged. The program is able to process multiple files in a single run, as specified by a wildcard file input specification. When multiple file are processed, it is specified for each file which fields have been modified and what new values have been assigned.

The program is able to make global changes and component-specific changes in a single run. For global changes, it modifies the specified header fields for all X5 files, Y5 files, and packet ringbuffers; for component-specific changes, it changes the specified header fields for the file whose STNID matches component STNID.

Headedit also provides functions to dump Y5 headers and to create a template inifile.

Usage and Options

Headedit reads field change messages from an inifile. It offers an option to create a standard template inifile temp.ini which lists all the fields that can be modified or inserted. Based on temp.ini, a new inifile can be generated easily by adding or deleting components, deleting fields, and modifying field data. It can be started manually from the command line by typing:

headedit -t

where -t is the switch for creating a template inifile.

For viewing the information of file header, *Headedit* offers another option to dump file headers. The main fields and their values of the file headers are written into file **message.txt** in a standard format. It can be started manually from the command line by typing:

headedit filespec -d

where -d is the switch for dumping file headers

filespec is the input file specification (including wildcards).

filespec may include files other than X5 file, Y5 file, or packet ringbuffers. Headedit is able to distinguish file types and processes X5 files, Y5 files, packet ringbuffers only. The main header fields are written into file message.txt. For other files, message.txt records a brief information about the files. If the system cannot find any file matching filespec, an error message is shown on the screen.

The following command changes header fields:

headedit filespec -i inifile

where -i is the switch for changing header fields

inifile is the name of the input file specifying field changes.

Headedit writes output to file message.txt. It specifies for each file which fields have been modified as well as error messages. If the system cannot find inifile, the screen will show an error message. Headedit reads inifile and checks if every assigned value is valid. If the assignment is invalid, the screen will display error messages and message.txt will show what the errors are.

Ini File

Inifile file contains the information of field changes. The file should follow the standard NMX inifile format. An inifile consists of one or more sections. Each section include a section name line followed one or more field lines of the form :

FieldName=value

Section names are enclosed in square brackets. For each field line, the lvalue is the standard name of the field. The rvalue is the value of each field which must be assigned to a value and cannot be empty. Each field value has a permitted value range. If the value is out of range, Headedit will terminate processing and no field will be changed. If a field value is a character string, every white space is accounted as a character. It is very crucial for the names of station, site, and network, especially STNID. Either inserting a white space in a string or adding a white space after a string will change the value. If a field is not needed to be changed, this field line should be deleted from the inifile.

Generally, an inifile for header editing consists of three parts. The first part is the first section which specifies the inifile name and the number of components in the inifile. It must be included as the first section of the inifile. The format of the section is as follows:

```
[ temp.ini ]
NumComponents=2
```

NumComponents must be assigned a value, but the inifile name here is not crucial and can be any name. In case of global change only, no component need to be specified and the NumComponents should be assigned to zero, i.e., NumComponents=0.

The second part of an inifile specifies the field changes on a global change basis. This part has only one section. For all X5 files, Y5 files, and packet ringbuffers given by filespec, the specified fields in this section will be modified. The format of the section is as follows:

```
[ ChangeAll ]
NetworkId=Network
Comment=General Comments
SensorType=SensorType
Sensitivity=1.0000
SensFreq=1.00
SensUnits=SensUnits
CalibUnits=CalibUnits
ResponseFile=ResponseFile
```

The section name has to be "ChangeAll". The above example lists every header field that can be specified for global change. Other header fields cannot be specified for global change. If a field listed above is not needed to be changed, the field line should be deleted from this section.

Some packet ringbuffers may not contain ResponseFile field. Headedit is able to check if the header includes ResponseFile field and does not assign ResponseFile field new value to those headers without ResponseFile field. ResponseFile field can be specified for global changes including those files without RespsonseFile field.

In case of component-specific changes only, this part should be empty. All field lines should be deleted. The section name line [ChangeAll] can be left in the inifile or deleted also.

The third part of the inifile specifies field changes on a component-by-component basis. This part may contain one or more sections. The number of sections is given by NumComponents in the first part. Each section name in this part consists of a keyword "Component" and an index number. The index number changes from 1 to the value of NumComponents. The keyword "Component" should not be changed. "Component" and index number are separated by white space. If the file header STNID matches the STNID of a component, Headedit will change the file header fields specified in this component. Same STNID should not be specified for two different components. Otherwise, Headedit will terminate processing, and no header field will be changed. The format of this part is as follows:

```
[ Component 1 ]
 STNID=STNID1
 NewSTNID=NewSTNID1
 NetworkId=Network1
 SiteName=SiteName1
 Comment=Comment1
 SensorType=SensorType1
 Latitude=45.0000
 Longitude=-70.0000
 Depth=0.00
 Elevation=0.00
 Azimuth=0.00
 Dip=0.00
 Sensitivity=1.0000
 SensFreq=1.00
 SensUnits=SensUnits1
 CalibUnits=CalibUnits1
 ResponseFile=ResponseFile1
[ Component 2 ]
 STNID=STNID2
 etc.
```

In the above example, [Component 1] section lists every field that can be specified for component-specific changes. Other header fields can not specified here. If a field do not need to be specified, the field should not be included in the component.

In case of a field is specified in both ChangeAll and Component sections, the field will be changed according to the component specification.

Environment

HEADEDIT runs under Windows 9x/NT/2000 and Solaris.

Examples

The following command creates a template inifile to a text file temp.ini:

```
headedit -t
```

temp.ini follows the standard inifile format.

The following command dumps headers of the all X5 files, Y5 files, and packet ringbuffers in current directory:

```
headedit *.* -d
```

File message.txt displays main fields of these headers.

The following command changes fields of the all packet ringbuffer files in current directory based on the specifications given by inifile headedit.ini:

```
headedit r*.* -i headedit.ini
```

File message.txt contains which files has been processed and which fields have been modified and what the old values and new values are.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\headedit.fm
Date last revised: 2002-10-30

Name

MAKESEED

Version

2.10

Synopsis

makeseed InputDirectory [-r ResponseDir] [-o OutputDestination][-y][-d]

Description

MAKESEED is a data conversion utility that creates SEED version 2.3 data volumes from Nanometrics X5 or Y5 files. The program reads in time series files and the system response file, and writes the SEED volume containing the SEED header and data in Steim compressed data format.

The single required command line argument of MAKESEED is a directory name. MAKESEED looks for X files and the system response file in the given directory path, and write SEED data volume in the same directory. The name of the response file is sought in each input data file, and is default to "seed.rsp" if not found. On the other hand, a user can specify

- the output directory path and/or name,
- the directory path to the system response file,
- input data format if data is in Y format, and
- output data content if the user wants to look at the header information only.

Options

Input Directory It is the first argument and the only mandatory argument of MAKESEED. It specifies the directory path to the input data files. The argument must end with '\' or '/' (Windows or UNIX) and is interpreted as the name of the directory containing the input X5 or Y5 data files. The response files are assumed to be in this directory unless specified by the -r option.

-r Response Directory This option and its mandatory argument specify the directory path to the system response files. The path name must end with a '\' or '/'. If this option is omitted the response files are assumed to be in the input directory as specified by the first argument.

-o Output Destination This option and its mandatory argument specify the directory path where the SEED file is created. If the option is not used, the SEED file is created in the input directory as specified by the first argument. The default name of the SEED file is based on the time stamp, which indicates the beginning of the data, in the following format:

Syyyymmdd.hhmmss

If the -o argument specifies a directory (i.e. ending with '\' or '/') the name of the SEED volume is based on the time stamp name, and the file is created in the specified directory.

If the -o argument specifies only a filename (no directory path) the SEED volume is given the specified name, and it is created in the input directory as specified by the first argument.

If the -o argument specifies both directory path and filename the SEED volume is given the specified name and is created in the specified directory.

-y The option specifies that the input data files are in the Nanometrics Y5 format. If the option is omitted, MAKESEED assumes that the input data files are in X5 format.

-d The option specifies that the output contains SEED header only, i.e., the data less volume. In this case, MAKESEED writes to the file the volume index control header, abbreviation dictionary control header, and station control header.

Usage

MAKESEED has command line help available for each of the above command line parameters, as well as others which are still in the process of implementation.

To get the general syntax for the command line and identify the names of the command line parameters and arguments type the following:

makeseed -h

This syntax help screen also gives additional information on how to get further help on a particular command line parameter.

The following are some examples in using MAKESEED program on a personal computer.

makeseed x:\data

converts all the X files in x:\data directory to a SEED volume named by the beginning time of the data record in the same directory. The system response information is assumed in directory x:\data.

makeseed x:\data\ -o x:\seeddata\event1.seed -y

converts all the Y files in x:\data directory to a SEED volume with name events1.seed in directory x:\seeddata. The system response information is assumed in directory x:\data.

makeseed x:\data\ -r x:\events\ -y

converts all the Y files in x:\data\ directory to a SEED volume named by the beginning time of the data record in the same directory. The system response information is searched in x:\events directory.

Note that the SEED data volume will be created even the system response information cannot be found.

Limitations

Current version of MAKESEED does not create mini-SEED volume, and does not write SEED Blockette 32, which contains seismic event information.

Current version of MAKESEED does not read in any auxiliary file, e.g., naqsp.log.

Current version of MAKESEED does not generate 0 stage response.

Files

seed.rsp file: a default response file for MAKESEED. On writing a SEED volume, a user should use the specific response file for his/her acquisition system provided by Nanometrics, or edit the file to make it describe properly the operating system that has acquired the time series in the SEED file.

Environment

MAKESEED runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.7 or later.

See Also

EXTRACTP, X5DECOMP

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\makeseed.fm
Date last revised: 2002-10-30

Name

RBFSUM

Version

1.06

Synopsis

rbfsum FileSpec [-s StartTime] [-d [N] | -h [N]] [-u]

Description

RBFSUM is a utility program to create a daily or hourly summary of the contents of one or more seismic data ringbuffer files. The summary for each file is presented in a tabular format, showing for each hour (or day):

- the number of seconds of data collected during the hour,
- the percentage of the data that was received as retransmits (either requested or sent by transmit-twice option),
- the number of seconds of data missed during the hour, and
- the number of breaks in the data which started during the hour.

The output is written to the standard output stream, and may be redirected to a file.

Options

FileSpec	Mandatory argument specifying which files to process. This may include wildcards. RBFSUM will create a summary for each ringbuffer whose filename matches FileSpec. Non-ringbuffer files are ignored.
-s StartTime	Optional argument specifying the time at which to begin the summary. StartTime should be specified in the format **yyyy-mm-dd-hh**. If StartTime is not specified, the summary will begin at the start of each file.
-d [N]	Optional argument specifying that the summary should report daily (rather than hourly) summaries. If N is given, RBFSUM will create a summary for N days (starting at StartTime). If N is omitted, the summary will end on the day containing the latest entry in the file. Hourly summaries are the default.
-h [N]	Optional argument specifying that the summary should report hourly (rather than daily) summaries. If N is given, RBFSUM will create a summary for N hours (starting at StartTime). If N is omitted, the summary will end on the hour containing the latest entry in the file. Hourly summaries are the default.
-u	Optional argument specifying that times should be printed as unformatted integers rather than in date/time format.

Note: -d and -h are mutually exclusive options.

Environment

RBFSUM runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.8 or later.

Examples

The following command creates an hourly summary for all ringbuffers matching RSTN01.BH*:

```
rbfsum  RSTN01.BH*
```

The following command creates an hourly summary for RSTN01.BHZ, for the 24-hour period starting at 6:00 am, March 15, 1997:

```
rbfsum  RSTN01.BHZ -s 1997-03-15_06:00 -h 24
```

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\rbfsum.fm
Date last revised: 2002-10-30

Name

RBTRIM

Version

1.05

Synopsis

RBTrim InputFile

Description

RBTRIM removes any unused space at the end of a ringbuffer file to reduce the file size before archiving. It operates on both data ringbuffers and state-of-health ringbuffers. RBTRIM produces a summary report showing the number of packets in the file, the packet size, and the file size after trimming.

RBTRIM has no effect on ringbuffers which have wrapped, or files which have already been trimmed.

The Orion data acquisition software cannot reuse files which have been trimmed.

Environment

RBTRIM runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.7 or later.

Configuration

RBTRIM requires one command line argument:

InputFile The pathname of the file (or files) to be trimmed. Wildcards are permitted.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\rbtrim.fm
Date last revised: 2002-10-30

Name

RESPONSE

Version

1.22

Synopsis

response [filename]

Description

RESPONSE generates a system response file, the content of which corresponds with the station response header information in the SEED (Standard Exchange of Earthquake Data) data volume. In order to use the program, a user must know the type of sensor, the sampling rate, and the frequency of the IIR filter of the acquisition system. The response file thus generated is a text file which contains a number of stages describing the parameters of the sensor, analog-digital converter, and digital filters in the system. The stages for the Orion or HRD such as the FIR and IIR filters are exact. The sensitivity of the seismometer stage is the value after damping. The poles and zeroes in this stage are nominal values, and therefore, the operator may want to edit the file to put in the exact pole positions for the actual seismometers. The output of the program, e.g., seed.rsp, is used by MAKESEED to generate the system response information in a SEED data volume.

Environment

Windows 9x/NT/2000 and Solaris.

Usage

Running RESPONSE with one optional command line argument

response [filename]

invokes an interactive process to build a response file based on the user's input. The command line argument specifies the name of the output file to be created. If no name is specified, output is written to a file named seed.rsp.

The user is prompted to provide the following input parameters:

Sample rate	The sampling rate (sample/second) used by the digitizer. The supported values are: 10, 20, 25, 40, 50, 80, 100, 120, 125, 200, 250, 500, 1000.
IIR corner frequency	The corner frequency (in mHz) of the IIR filter. The supported values are: 0, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000.
Sensor type	The sensor attached to the data acquisition system. The supported types are: S13, L4C, CMG40T, SS-1, CMG3T.

Notes

1. The fields rtmStartDate and rtmEndDate are used by the program MAKESEED in converting Nanometrics X or Y data files to a SEED data volume. The time interval defined by these fields must include the time when the data in the X or Y files are collected.
2. The response file generated here is system specific. A user must generate a new response file if any component, such as the sensor or sampling rate, of the acquisition system is changed.
3. The default name for the response file is seed.rsp, which is also used by the MAKESEED program when the response file name is not defined in the input data files (X or Y files). The file is

overwritten at the location every time the program is executed. A user is advised to rename the file as soon as it is generated.

4. The final gain of the system is 1 V/M/S by default. For a system equipped with a broadband sensor, a user must know if the recording is using the normalized gain or the full scale. If it is the latter, the sensitivity of both the seismometer and the anti-alias filter (stage 2) must be changed.

See also

MAKESEED (version 2.x)

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\response.fm
Date last revised: 2002-10-30

Name

REXTRACT

Version

1.01

Synopsis

Rextract FileSpec OutputDir StartTime Duration

Description

REXTRACT is used to extract a segment of data from a ringbuffer and save it to a new, smaller ringbuffer. This may be used to extract hourly or daily data segments in a form suitable for archiving. Since the output files are ringbuffers, data can be further extracted from them using the appropriate playback program. The output files are tagged with the channel name and the start time of the extracted data, using the following format:

Filename = "R" + station name (up to 5 characters)
+ "_" + channel name (up to 3 characters)
+ "_" + date and time (yyyymmdd.hhmmss)

REXTRACT operates on all ringbuffer types (time-series data, state-of-health, and serial-data ringbuffers).

Usage

REXTRACT requires four command line arguments:

FileSpec	The pathname of the file (or files) from which to extract data.
OutputDir	The directory to which to write the output files.
StartTime	Start time of the data segment to be extracted, in the form yyyy-mm-dd-hh-mm-ss. The hours, minutes and seconds may be omitted.
Duration	Duration in seconds of the data segment to be extracted.

Example

The following command will extract data for the entire day of February 7, 2000 from all ringbuffers in the current directory, and write the output files to directory FebData:

Rextract R*.* FebData 2000-02-07 86400

Environment

REXTRACT is supported on Windows 9x/NT/2000 and Solaris.

REXTRACT requires Nanometrics DLLs release 1.7 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\rextract.fm
Date last revised: 2002-10-30

Name

SerialExtract

Version

1.2

Description

SerialExtract is a utility program to extract serial data from one or more serial data ringbuffer files. These files contain binary data collected from a serial port, indexed by time.

The program provides several different operating modes. The default mode extracts data for user-specified time segments into simple binary files suitable for processing via domain-specific software. The summary mode provides a summary of the contiguous data sections and gaps (missing data) for each ringbuffer. The hourly summary mode provides an hour-by-hour summary of data packets (and bytes) received and packets lost. The debug mode (unsupported) provides details of each packet received by the ringbuffer.

Summary mode output is written to the standard output stream, and may be redirected to a file.

Synopsis

SerialExtract is started from a console window using the following command syntax:

serialextract [mode] fileSpec [-s starttime -d duration] [-o outputDirectory] [-g] [-u]

where the command-line arguments are as follows:

Mode = [-r | -s | -h | -d]

Mode sets the type of extract to be done:

-r	extract data into binary files (this is the default)
-s	create summary of contiguous data sections and gaps for each ringbuffer
-h	create hourly summary of data packets received and lost
-d	debug mode provides details of each data packet

FileSpec

Specifies the full or relative pathname for the input ringbuffers, and may include wildcards.

startTime

Specifies the start time for the data segment to be extracted, in the format yyyy-mm-dd-hh-mm-ss. The default (if starttime is omitted) is the time of the earliest packet in the ringbuffer.

duration

Specifies the duration of the data segment to be extracted, in seconds. The default (if this parameter is omitted) is to extract from starttime to the end of the ringbuffer.

outputDirectory

Specifies the full or relative pathname of the destination directory for output files written in regular extraction mode. The default is the current directory.

-g

This switch controls multiple file output in regular extraction mode. The default mode is to ignore gaps and write the entire extracted segment for each ringbuffer to a single file. If this switch is specified and there are gaps in the ringbuffer, then each contiguous block of data in the ringbuffer (between gaps) will be written to a separate output file. If there are no gaps in the ringbuffer, this

switch has no effect.

-u

If this switch is specified, all time values are written as unformatted integer seconds since 1970, rather than in date/time format.

Extraction Modes

Binary Extract Mode (default)

In this mode, SerialExtract extracts data from each ringbuffer into a binary file containing only the original data written to the ringbuffer, without any header or delimiters. The data format is identical to the format originally received from the data source. Output files are named using the channel name and start time, using the following naming convention:

File name= "S"
+ channel name (e.g. STN01_GPS)
+ "_"
+ date (yyyymmdd.hhmmss)

An example name is SSTN01_GPS_19990815.095030.

Summary Mode

This creates a summary of the contiguous data sections and gaps for each ringbuffer. The summary is written to standard output and may be redirected to a file. The summary consists of one row of output for each data section or gap, with seven columns:

start_time	the start time of the first packet of the data section (or, for a gap, the start time of the last packet before gap)
end_time	the start time of the last packet in the data section (or , for a gap the start time of the first packet after the gap)
packets	the number of packets in the data section (or packets lost in a gap)
bytes	the number of bytes in the data section
start_index	the ringbuffer index for the first packet in the data section
start_sequence	the sequence number of the first packet in the data section
end_sequence	the sequence number of the last packet in the data section

Only *start_time, end_time* and *packets* are shown for gaps.

Hourly Summary Mode

This prints an hourly summary for each ringbuffer to standard output. For each ringbuffer, the program produces a table describing the availability of packets for each hour. This table consists of one row of output for each hour in the specified extraction interval, with six columns:

hour	the start time of the interval
packets	the number of packets received in this interval
bytes	the number of payload bytes received in this interval
RxTx(%)	the percentage of packets received in this interval which were retransmitted packets
lost	the number of packets lost in this hour
#Gaps	the number of individual gaps which start in this hour

For lost packets, the suffix (+) indicates that the data-recording instrument was restarted and that SerialExtract was unable to determine exactly how many packets are missing; the number given is the minimum. The suffix (e) indicates that a gap spans one or more hour boundaries. In this case, the total number of missing packets is known, but their time distribution is not. SerialExtract apportions the missing packets to each hour using the start and end time of the gap.

Debug Mode

This writes a detailed listing of the contents of each ringbuffer to standard output. The listing consists of one line for each packet in the ringbuffer, with the following fields:

start time	the start time of the packet
instrument ID	the serial number of the data-recording instrument
sequence	the sequence number of the packet
ReTx flag	'R' indicates that this is a retransmitted packet
data	the first and last 15 bytes of the payload data are shown. Unprintable characters are shown as dots.

Examples

SerialExtract RSTA01.trm -s 1999_6_15_8:00 -d 3600 -o mydata -g

Extracts all the payload bytes from the ring buffer file RSTA01.trm to the output file(s). If this file contains one or more gap, there will be multiple output files. The output files will be in directory mydata.

SerialExtract -s RSTA*.trm

Extracts the summary from all ring buffer files matching the wildcard descriptor RSTA*.trm This summary describes the contiguous blocks and gaps.

SerialExtract -d RSTA01.trm

Extracts in debug mode from the ring buffer RSTA01.trm. This will output detailed information for each packet.

Environment

SerialExtract runs under Java 1.2.

Required Files

This program uses SerialExtract.jar and SerialExtract.bat and requires Nanometrics DLLs v1.7 or higher.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\serialextract.fm
Date last revised: 2002-10-30

Name

SOHEXTRP

Version

1.60.00

Synopsis

sohextrp Arg1 [-o o1] [-s s1] [-e e1 | -d d1] [-l l1] [-f] [-n] [-u] [-m] [-w] [-b] [-c] [-g] [-t] [-v] [-x] [-r] [-h] [-k] [-i] [-p] [-a] [-q]

Arg1	SOH Ringbuffer file required argument (string)
-o	Output directory optional switch
o1	Output directory required switch argument (string)
-s	Start Time optional switch
s1	Start Time required switch argument (string)
-e	End Time optional switch
e1	End Time required switch argument (string)
-d	Duration optional switch
d1	Duration required switch argument (integer)
-l	Log file optional switch
l1	Log file optional switch argument (string)
-f	Fast SOH optional switch
-n	Slow SOH optional switch
-m	MinMax optional switch
-w	GPS location optional switch
-b	GPS error optional switch
-c	GPS channel optional switch
-g	GPS (all) optional switch
-t	Triggers optional switch
-y	Events optional switch
-x	VCXO optional switch
-r	Invalid SOH bundles optional switch
-h	HRD information optional switch
-k	Orion temperatures optional switch
-i	Orion voltages optional switch
-p	Orion powering optional switch
-a	Orion (all) optional switch
-q	GPS time quality optional switch
-3	RM3 SOH optional switch
-4	RM4 SOH optional switch
-v	Vsat SOH optional switch
-u	Unformatted IP address optional switch
-z	Test Mode optional switch

Description and Usage

SOHEXTRP is used to extract various streams of information from the binary SOH data file.

The default action (if none of the flags [-l l1] [-f] [-n] [-u] [-m] [-w] [-b] [-c] [-g] [-t] [-y] [-x] [-r] [-h] [-k] [-i] [-p] [-a] [-q] [-3] [-4] [-v] are used) is to extract all streams .

By default, all pertinent stream data in the file is extracted except if the -s switch is used either alone or in conjunction with either -e or -d switches.

All files except the log file are written with one message per line in a comma-separated format suit-

able for direct importation into a spreadsheet program. The contents of these files is described later in section "Comma Delimited Data Files".

o Optional output directory switch.

Use this switch and its required parameter o1 to specify the output directory. If this switch is not used, the files will be placed in the current directory.

o1 Output directory parameter, required if o switch is specified.

Format: path1\path2for a relative path

\path1\path2for an absolute path

If this option is not used, the output is placed in the current working directory.

s Optional start time of data extraction switch.

Use this switch and its required parameter s1 to specify the start time of data extraction. If this switch is not used, extraction begins at the first record in the SOH file.

s1 Start time of data extraction parameter, required if s switch is specified.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1999-10-17-23-11-17

e Optional end time of data extraction switch.

Use this switch and its required parameter e1 to specify the start time of data extraction. If neither this switch nor the d switch is used, extraction continues to the last record in the SOH file. This switch cannot be used in conjunction with the d switch.

e1 End time of data extraction parameter, required if e switch is specified.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1999-10-17-23-11-17

d Optional duration of data extraction switch.

Use this switch and it's required parameter d1 to specify the duration of data extraction. If neither this switch nor the e switch is used, extraction continues to the last record in the SOH file. This switch cannot be used in conjunction with the e switch.

d1 Duration of data extraction in seconds, required if d switch is specified.

Format: Integer
e.g. 3600

l Optional log messages extraction switch.

If this switch is used, the messages log (file extension .log) file is created.

l1 Optional error severity specifier.

This parameter is a single character specifying the minimum error level to extract. This may be one of the following:

F:	extract fatal messages only
E:	extract error and fatal messages only
W:	extract warning, error and fatal messages
I:	extract information, warning, error and fatal messages

If this parameter is not specified, all log information is extracted.

u Unformatted IP address optional switch.

If this switch is used, all IP addresses are printed as unformatted integers rather than in dotted deci-

mal format. This affects streams BXB, RSE, RSS, RXB, TSE, and TXB.

Other command-line switches:

The balance of the switches determine which information streams (other than log messages, discussed above) are extracted. Each stream is written to a separate file in the output directory with different identifying file extension.

Table 1-1 File extensions for extracted streams

Flag	Information Stream	File Extension
f	fast state-of-health	fsh, fes
n	slow state-of-health	ssh, ses
m	MinMax data channel 1	mx1
m	MinMax data channel 2	mx2
w or g	GPS location information	loc
b or g	GPS error information	gps
c or g	GPS channel status	gst
t	triggers	trg
y	events	elf
x	VCXO calibration	vcx
r	error, unrecognized data	err
h	HRD information	hrd
k or a	Orion temperatures	tmp
i or a	Orion voltages	vlt
p or a	Orion power supply	pwr
q	GPS time quality	gtq
3	RM3	rsh, rx*
4	RM4	gsi, loc, prx, spe, spm
v	Libra	aus, bxb, epx, gsi, les, lgq, lis, loc, los, lsq, prx, rse, rss, rxb, sdb, spe, spm, tps, tse, txb
z	all packets marked as test packets	*

Comma Delimited Data Files

See the summary sheet for descriptions of the comma delimited data files.

Log File

The log file provides a record of the normal ongoing operation of the instrument, as well as specific incidents which may affect data quality or timing. Log messages for the HRD and Orion are stored in the SOH file and can be extracted using SOHEXTRP. Log messages from the RM4 and Libra

family of instruments are written directly to the Naqs log file, and are not available from SOHEXTRP.

Five different categories of messages are generated:

Info: information messages which document normal operation.
Warning: minor incidents which may slightly affect data quality or timing.
Error: incidents which may result in significant loss of data.
Fatal: serious malfunctions.
Debug: verbose trace messages.

By default, SOHEXTRP extracts all log messages from the SOH input file unless the optional l1 parameter is used to perform a selective extract of log information.

Log File Format

Each line of the log file represents a single log message, with the fields described in Table 1-2.

Table 1-2 Log file message fields

Field Name	Columns	Format	Description
Date	8-17	yyyy-mm-dd	date the message was generated
Time	19-26	hh:mm:ss	time the message was generated
Error Severity	28	character	error severity (see below)
Error Source	29	character	source processor (see below)
Functional Area	30	character	not currently used
Error Type	32-35	integer	a 4-digit error number
Message	37-76	character string	an explanatory message

Error severity is one of F (fatal), E (error), W (warning), I (information) or D (debug).

Error source is one of A (AUX), T (TCP) or D (DSP).

Required Files

When running SOHEXTRP the Orion message file ORION.MSG must be in the current directory or in the same directory as the SOHEXTRP program. This file contains the error code definitions and is required to properly interpret the log entries. Nanometrics DLLs version 1.8 or later (for Win32) are required.

Examples

sohextrp RBFAAB.SOH

Extracts all information from the SOH file RBFAAB.SOH to the output files. The output files will be in the current working directory.

sohextrp RBFAAB.SOH -l W

Extracts all warning, error and fatal message from RBFAAB.SOH to a log file RBFAAB.LOG in the current working directory.

sohextrp RBFAAB.SOH -l W

Extracts all warning, error and fatal log messages from RBFAAB.SOH to a log file RBFAAB.LOG in the current working directory.

sohextrp RBFAAB.SOH -o dir -f -w

Extracts to output directory "dir" the fast SOH and GPS location files. "dir" is relative to the current directory.

sohextrp RBFAAB.SOH -o \dir -s 1996-01-01-12:00:00 -d 60 -g

Extracts to output directory "\dir" the GPS location, GPS error , and GPS channel files for 60 seconds beginning at 12:00:00 January 1, 1996. "\dir" an absolute path.

Environment

SOHEXTRP runs under Windows 9x/NT/2000 and Solaris.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\sohextrp.fm
Date last revised: 2002-10-30

STATE-OF-HEALTH FILES DESCRIPTION

HRD SOH

VCXO Calibration Data (.VCX)			
1	Time	Seconds	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
2	Time	Date-Time	
3	VcxoValue	counts	VCXO value in counts
4	TimeDiffAtLock	microseconds	Time difference between GPS time and Digitizer time at point of GPS lock
5	TimeError	microseconds	Time error between the GPS and Digitizer
6	FreqError	microseconds/ second	Frequency error between the GPS and Digitizer
7	CrystalTemp	counts	Temperature of the VCXO circuit in counts
8	PLLStatus		defined as: 1 Digitizer PLL fine locked to GPS 2 Digitizer PLL coarse locked to GPS 3 Digitizer free-running using temp. compensation 4 Digitizer free-running using temp. comp. With GPS on
9	GPSStatus		defined as: 0 3D navigation 1 2D navigation 2 Single satellite 3 Searching for satellites 4 GPS engine powered off 5 Error state

GPS Time Quality (.GTQ)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	GPSOnTime	seconds	The time interval the GPS was on during this duty cycle
4	GPSOffTime	seconds	The time interval the GPS was off during this duty cycle
5	GPSTimeToLock	seconds	The time interval the GPS took to lock from GPS power on
6	TimeDiffAtLock	microseconds	The difference between GPS and Orion after GPS first locks
7	VCXOError	ppm	Calculation of the VCXO drift in ppm over the last duty cycle
8	TempCompOffset	microseconds	Average Vcxo drift in microseconds/sec.
9	GPSOffReason		defined as: 0 PLL time error was zero 1 GPS on time expired
10	GPSFinalMode		defined as: 0 3D Navigation 1 2D Navigation 2 Single satellite 3 Searching for satellites

GPS Satellite Status (.GST)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	SolnState		Defined as: 1 Cold start, 2 Acquisition 3 2D Navigation 4 3D navigation
4	FigMerit		Figure of Merit: based on Horizontal position error (1-best, 9-unlocked)
5	NSatforSoln		Number of Satellites for solution Number of Satellites in navigation solution
6	NSatTracked		Number of Satellites the GPS is tracking
7 to 11	Channel Activity Act1-Act5		Defined as: 0 idle 1 searching for satellite 3 tracking a satellite
12 to 16	SNR1-SNR5		Signal to noise ratio for each channel (0-63)

17 to 21	PRN1-PRN5	dB	PRN code for the satellite being tracked by each channel

GPS Location (.LOC)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Latitude	degrees	GPS calculated latitude
4	Longitude	degrees	GPS calculated longitude
5	Elevation	degrees	GPS calculated elevation

Fast External SOH (.FES)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	FSOH1	User defined	Calibrated fast SOH for channel 1
4	FSOH2	User defined	Calibrated fast SOH for channel 2
5	FSOH3	User defined	Calibrated fast SOH for channel 3

Slow External SOH (.SES)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	SSOH1	User defined	Calibrated slow SOH for channel 1
4	SSOH2	User defined	Calibrated slow SOH for channel 2
5	SSOH3	User defined	Calibrated slow SOH for channel 3

HRD Slow Internal SOH (.HRD)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	BattVoltage	Volts	Battery Voltage: Measures the input voltage to the HRD
4	VCXOTemp	Degree Celsius	VCXO Temp.: Measures the temperature of the VCXO circuit
5	RadioSNR	dB	Radio SNR: Measures the radio signal to noise ratio (not implemented)

Orion SOH

Min-Max1 Data (.MX1)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Min	Counts	Filtered minimum over the sample interval
4	Max	Counts	Filtered maximum over the sample interval

Min-Max2 Data (.MX2)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Min	Counts	Filtered minimum over the sample interval
4	Max	Counts	Filtered maximum over the sample interval

Event List (.ELF)			
1	StartTime	Seconds	The UTC start time of the event in seconds since 00:00:00, January 1, 1970.
2	StartTime	Date-time	The above time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Duration	seconds	The length of the event
4	Type		Defined as: 1 external, 2 internal triggered, 4 manual (calibration)
5	Trigger1		Defined as: 0 no trigger, 1 trigger

6	Trigger2		Defined as: 0 no trigger, 1 trigger
7	Trigger3		Defined as: 0 no trigger, 1 trigger
8	Trigger4		Defined as: 0 no trigger, 1 trigger
9	Trigger5		Defined as: 0 no trigger, 1 trigger
10	Trigger6		Defined as: 0 no trigger, 1 trigger

Trigger Data (.TRG)			
1	StartTime	Seconds	The UTC start time of the trigger in seconds since 00:00:00, January 1, 1970
2	StartTime	Date-time	The above time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	EndTime	Seconds	UTC end time of the trigger event
4	EndTime	Date-time	UTC end time of the trigger event in data-time format
5	Duration	Seconds	Duration of the trigger event(end time -start time)
6	Channel		Digitizer channel number(1-6)
7	Trig#		Trigger number (1-6)
8	type		0-threshold, 1- STA/LTA
9	LtaValue	counts	Long term average at the start of the trigger
10	EarlyPeakAmp	counts	Peak amplitude during the first a few seconds (programmable) of trigger
11	EarlyHalfPeriod	samples	Half period of the peak amplitude signal
12	EarlySamplesToPeak	samples	Time in samples from the start of the trigger to the peak amplitude
13	FnalPeakAmp	counts	Peak amplitude over the entire amplitude event
14	FinalHalfPeriod	samples	Halt period of the peak amplitude signal
15	FinalSamplesToPeak	samples	Time from the start of the trigger to the peak amplitude

Orion Internal Temperature Slow SOH (.TMP)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	UserCPUTemp	degree Celsius	User CPU temperature
4	VCXOTemp	degree Celsius	VCXO circuit temperature
5	DiskTemp	degree Celsius	Disk drive temperature

Orion Source Voltage SOH (.VLT)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	External	volt	External battery voltage
4	Internal	volt	Internal battery voltage
5	Mains	volt	Mains voltage

Orion Powering Status Slow SOH (.PWR)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	ChargeCurrent	amps	Charge current supplied to the internal battery while charging
4	DigitizerVoltage	volt	The supply voltage measured at the digitizer
5	ExtBatteryStatus		Defined as: 0 unknown 1 not present 2 in use 3 ready 4 discharged 5 charging 6 shorted cell 7 shorted cell in use

6	IntBatteryStatus		Internal battery status, Same as above
7	MainStatus		Mains status, Same as above
8	PowerSource		Defined as: 1 main supply on, 2 external battery supply on , 3 internal battery supply on
9	UserCPUStatus		Defined as: 0 both user_CPU and heater off, 1 user_CPU on, heater off, 2 heater on, user_CPU off
10	ChargeState		Defined as: 0 charged off 1 low current charge 3 high current charge

RM3 SOH

RM-3 SOH (.RSH)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	BatteryVoltage	Degree C	Battery voltage
4	ExternalVoltage	Degree C	External voltage
5	Temperature	Degree C	Temperature

RM-3 Receiver Status (receiver 1 to .RX1, receiver 2 to .RX2, etc.)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	SNR		Rx s/n ratio (average over the interval)
4	DataPackets		Valid data packets received during the interval
6	FillerPackets		Valid filler packets received during the interval
7	BadPackets		Bad packets (CRC error) received during the interval
8	DiscardedPackets		packets discarded (buffer overrun) during the interval

RM4 SOH

GPS Location (.LOC)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Latitude	degrees	GPS calculated latitude
4	Longitude	degrees	GPS calculated longitude
5	Elevation	degrees	GPS calculated elevation

GPS Satellite Information (.GSI)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Channel		Channel number (0-15)
4	Millisecs		Defined as: 0 Don't have good knowledge of integer millisecond range to this satellite. 1 msec from sub_frame data collection, 2 verified by a bit crossing time, 3 verified by successful position fix, 4 suspected msec error
5	PRN		PRN
6	AcquisitionType		Defined as: 0 unlocked, 1 search, 2 track
7	Elevation	degrees	Satellite elevation above horizon
7	Azimuth	degrees	Satellite azimuth CW from North
8	SignalLevel	dB	Signal level

Serial Port Map (.SPM)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Index		Index in table
4	Port		serial port number
5	MinitesSinceLast	minutes	number of minutes since last packet arrived
6	HrdID		instrument ID for HRD on this port

Packet Reader Error (.PRX)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Port		serial port number
4	BadPackets		Number of Bad packets read from this port since startup (mod 10 million)
5	GoodPackets		Number of Good packets read from this port since startup (mod 10 million)
6	LostPackets		Number of packets lost reading from this port since startup (mod 1 million)
7	TxPackets		Number of packets written to this port since startup (mod 10 thousand)

Serial Port Error (.SPE)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Port		serial port number
4	OverRuns		Number of serial port overrun errors, since startup
5	FrameErrors		Number of serial port frame errors, since startup

Libra SOH

GPS Location (.LOC)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Latitude	degrees	GPS calculated latitude
4	Longitude	degrees	GPS calculated longitude
5	Elevation	degrees	GPS calculated elevation

GPS Satellite Information (.GSI)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Channel		Channel number (0-15)
4	Millisecs		Defined as: 0 Don't have good knowledge of integer millisecond range to this satellite, 1 msec from sub_frame data collection, 2 verified by a bit crossing time, 3 verified by successful position fix, 4 suspected msec error
5	PRN		PRN
6	AcquisitionType		Defined as: 0 unlocked, 1 search, 2 track
7	Elevation	degrees	Satellite elevation above horizon
7	Azimuth	degrees	Satellite azimuth CW from North
8	SignalLevel	dB	Signal level

Serial Port Map (.SPM)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Index		Index in table
4	Port		serial port number

5	MinutesSinceLast		number of minutes since last packet arrived
6	HrdID		Instrument ID,for HRD on this port

Packet Reader Error (.PRX)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Port		serial port number
4	BadPackets		Number of Bad packets read from this port since startup (mod 10 million)
5	GoodPackets		Number of Good packets read from this port since startup (mod 10 million)
6	LostPackets		Number of packets lost reading from this port since startup (mod 1 million)
7	TxPackets		Number of packets written to this port since startup (mod 10 thousand)

Serial Port Error (.SPE)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Port		serial port number
4	OverRuns		serial port overrun errors since startup
5	FrameErrors		serial port frame errors since startup

Receiver Slot SOH (.RSS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter.
4	DQT_AGC	dB	Power level of the carrier received from the satellite, measured at the LNB output
5	CarrierOffset	Hz	Difference between the expected center frequency of the received carrier and the measured center frequency of the received carrier.
6	SymbolOffset	Hz	Frequency error in data clock
7	RxEb_No	dB	Receiver signal to noise ratio

Transmitter Slot Error SOH (.TSE)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter.
4	NumberOfBadPackets		Number of bad packets since the start of this TDMA configuration
5	NumberOfGoodPackets		Number of good packets since the start of this TDMA configuration

Receiver Slot Errors (.RSE)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter.
4	BadPackets		Number of bad packets since the start of this TDMA configuration
5	GoodPackets		Number of good packets since the start of this TDMA configuration

Libra Instrument SOH (.LIS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	10MhzFreqError	Parts/billion	Frequency error
4	SSPBTtemp	Degree C	SSPB temperature
5	ControllerTemp	Degree C	Controller temperature
6	ModemTemp	Degree C	Modulator temperature
7	BatteryVoltage	volts	Battery voltage

Libra Environment SOH (.LES)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	ExternalSoh1	User defined	External SOH for channel 1

4	ExternalSoh2	User defined	External SOH for channel 2
5	ExternalSoh3	User defined	External SOH for channel 3

Transmitter Port SOH (.TXB)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter..
4	TransmitteFreq(MHz)	MHz	Center frequency of the transmit carrier at the transceiver L-Band port.
5	TransmitterLevel	counts	Count of DAC input which controls power.

Receiver SOH (.RXB)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter..
4	ReceiverFreq(MHz)	MHz	Center frequency of received carrier.

Burst SOH (.BXB)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	TransmitterID		The instrument ID or IP address of the satellite interface of the transmitter.
4	SlotState		Where 0 find (sweeping for carrier), 1 verify (has carrier, looking for data), 2 track (received data)
5	BurstState		Where 0 not found, 1 found CW, 2 found UW, 3 found data
6	GoodBursts		Number of good bursts since the start of this TDMA configuration
6	BadBursts		Number of bad bursts since the start of this TDMA configuration

Epoch (.EPX)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	NextEpochStartTime		The start time of next epoch.

Libra GPS Time Quality (.LGQ)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	GPSStatus		Defined as: 0doing position fixes, 1 don't have GPS time yet, 2 need initialization 3 PDOP is too high 8 No usable satellites 9 only 1 usable satellite 10 only 2 usable satellites 11 only 3 usable satellites 12 the chosen satellite is unusable
4	UnusableSatellites		The number of unusable satellites
5	PDOP		Position dilution of precision
6	TDOP		Time dilution of precision

Libra System Time Quality (.LSQ)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)

3	SystemTimeQuality	nanoseconds	Defined as: -10 time_unknown, -5 time incorrect, -3 time reference bad, -1 time_not_good, >=0 worst prediction of time error
4	PLLmode	counts	Defined as: 1 fine_lock, 2 coarse_lock, 3 no_lock
5	TimeDisplacement	nanoseconds	Time error in nanoseconds
6	TimeVelocity	nanoseconds/ second	Time rate error
7	CurrentCompensation	counts	Current compensation

Libra Operation State (.LOS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	state		bitfield indicating operating state: bit 0 (LSB) network transmission state: 1: shutdown , 0: running

Serial Data Bytes (.SDB)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	PortNumber		Port number
4	BytesRead	byte	Bytes read since startup (mod 1 billion)
5	BytesWritten	byte	Bytes written since startup (mod 1 billion)

Telemetry Packet Sender Status (.TPS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	PortNumber		Port number
4	BadPackets		Number of Bad packets read from this port since startup (mod 10 million)
5	GoodPackets		Number of Good packets read from this port since startup (mod 10 million)
6	LostPackets		Number of packets lost reading from this port since startup (mod 10 thousand)
7	TxPackets		Number of packets written to this port since startup (mod 10 million)

Authentication Status (.AUS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	FramesBuilt		Number of CD1 subframes built since startup (mod 1 billion)
4	SignErrors		Number of subframes with invalid signature since startup (mod 10 thousand)
5	MissingSoh		Number of subframes with missing status bits since startup (mod 10 thousand)
6	MissingData		Number of subframes with missing data samples since startup (mod 10 thousand)

NMX bus Device List (.NBD)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Instrument ID		Instrument ID of Device 1
4	Instrument ID		Instrument ID of Device 2
5	Instrument ID		Instrument ID of Device 3
6	Instrument ID		Instrument ID of Device 4
7	Instrument ID		Instrument ID of Device 5
8	Instrument ID		Instrument ID of Device 6

TimeServer SOH

Instrument (.TSI)

1	Time	Seconds	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
2	Time	Date-Time	
3	Temperature	Degrees C	Temperature SOH circuit
4	Supply Voltage	Volts	Supply voltage SOH circuit
5	NMX bus Voltage	Volts	Bus voltage SOH circuit
6	Analog Input	Volts	External analog voltage
7	Status		0 disabled 1 enabled
8	Uptime	Minutes	Time since last reboot (mod 10 million)

Time Server Time PLL (.TST)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Fast Count	ns	Time Server system time
4	Status		Defined as: -0 initializing -1 no time -2 raw time -3 approximate time -4 measuring frequency -5-6 reserved -7 no lock -8 coarse lock -9 fine lock -10 superfine lock -11-15 reserved
5	Time Quality		-0 < 100 ns -1 < 200 ns -2 < 500 ns -3 < 1 micro s -4 < 2 micro s -5 < 5 micro s -6 < 10 micro s -7 < 20 micro s -8 < 50 micro s -9 < 100 micro s -10 < 1 ms -11 < 10 ms -12 < 100 ms -13 < 1 s -14 < 10 s -15 < 10 s
6	Time Error	sn	Measured time error
7	Frequency Error	Hz	Measured frequency error
8	TimeSince Lock Loss	s	Time spent in current state of GPS lock loss, if applicable (mod 10 million)

Time Server M12 GPS (.TSM)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Status		Defined as: -2 bad geometry -3 acquiring satellites -4 position hold -5 propagate mode -6 2D fix -7 3D fix
4	Auto survey mode		Defined as: -0 disabled -1 enabled
5	Insufficient visible sats		Defined as: -0 false -1 true

6	Antenna Status		Defined as: -0 ok -1 overcurrent -2 not connected -3 n/a
7	Engine Powered		Defined as: -0 not powered -1 powered
8	Visible Satellites		Number of visible satellites
9	Tracked Satellites		Number of tracked satellites
10	UTC Offset	Seconds	Difference between UTC and GPS time frames
11	Clock Bias	ns	Clock bias of GPS engine
12	Frequency Bias	Hz	Frequency bias of GPS engine
13	Receiver Temperature	Degrees C	Temperature measure on GPS engine
14	Antenna Voltage	Volts	Measured antenna voltage

NMX bus Master (.NBS)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Intstrument id		ID of the instrument these data relate to
4	Requests Received		Number of slot requests received (mod 10 million)
5	Permits Issued		Number of slot permits issued (mod 10 million)
6	NACK Issued		Number of slot denials issued (mod 10 million)

NMXbus SOH

NMX bus Request (.NBR)			
1	Time	Seconds	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
2	Time	Date-Time	
3	Instrument ID		Instrument id of TimeServer
4	Requests Sent		Number of slot requests sent (mod 10 million)
5	Permits Received		Number of slot permits received (mod 10 million)
6	NACK'S Recieved		Number of slot denials received (mod 10 million)

NMX bus Rx (.NBX)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Rx Good Packets		Number of good bus messages received (mod 1 billion)
4	Rx Bytes		Number of bytes received (mod 1 billion)
5	Rx Buffer Overrun		Number of Rx FIFO overruns (mod 10,000)
6	HDLC Errors		Number of HDLC errors; CRC, abort of other (mod 10,000)

NMX bus Tx (.NBT)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Tx Good Packets		Number of good bus messages sent (mod 1 billion)
4	Tx Bytes		Number of bytes transmitted (mod 1 billion)
5	Tx Buffer Underrun		Number of Tx FIFO underruns (mod 10,000)
6	Discarded Packets		Number of discarded packets; e.g dur to collisions of defers (mod 10,000)

Trident SOH

PLL Status (.TPL)			
1	Time	Seconds	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
2	Time	Date-Time	

3	Current State		Defined as: -0 INIT, not digitizing -1 TIME, correcting time error -2 ACQ0 -3 TRK1 -4 TRK2 -5 TRK3 -6 TRK4
4	DAC Counts		Value to DAC to control VCXO (ranges from 0 to 4096)
5	Time Error	Micro s	Time error in micorseconds relative to TimeServer (+ve indicates Trident ahead)
6	Temperature	Deg C	Temperature

Error Message

SOH Error (.ERR)			
1	Time	Seconds	The UTC Time since 00:00:00, January 1, 1970 that the SOH is sampled.
2	Time	Date-time	UTC time in a date-time format (YYYY-MM-DD HH:MM:SS)
3	Type		Unrecognized bundle type
4	Long0		Data in bytes 1-4
5	Long1		Data in bytes 5-8
6	Long2		Data in bytes 9-12
7	Long3		Data in bytes 13-16

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\sohinfo.fm
Date last revised: 2002-10-30

Name

SOHREADER

Version

1.00.00

Synopsis

sohextrp Arg1 [-o o1] [-s s1] [-d d1] [

Arg1-i	SOH Ringbuffer files required argument (string); space separated; wildcards ok
-o	Output directory optional switch
-s	Start Time optional switch
-d	Duration optional switch

Description and Usage

SohReader is used to extract various streams of information from the binary SOH data file.

The default action is to extract all streams from all files listed.

By default, all pertinent stream data in the file is extracted except if the -s switch is used either alone or in conjunction with the -d switches.

All files except the log file are written with one message per line in a comma-separated format suitable for direct importation into a spreadsheet program. The contents of these files is described later in section "Comma Delimited Data Files".

o Optional output directory switch.

Use this switch and its required parameter o1 to specify the output directory. If this switch is not used, the files will be placed in a directory of the same name as the SOH FILE in the current directory. (For example, ORIONS1.SOH will be placed in the ORIONS1 directory.) If the switch is used, the files will be placed in a directory of the same name as the file, as above, but the directory will be placed in the directory named in the o1 parameter.

o1 Output directory parameter, required if o switch is specified.

Format: path1\path2for a relative path

If this option is not used, the output is placed in the current working directory.

s Optional start time of data extraction switch.

Use this switch and its required parameter s1 to specify the start time of data extraction. If this switch is not used, extraction begins at the first record in the SOH file.

s1 Start time of data extraction parameter, required if s switch is specified.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1999-10-17-23-11-17

d Optional duration of data extraction switch.

Use this switch and its required parameter d1 to specify the duration of data extraction. If this switch not used, extraction continues to the last record in the SOH file.

d1 Duration of data extraction in seconds, required if d switch is specified.

Duration of data extraction in seconds, required if d switch is specified.

Format: Integer
e.g. 3600

Comma Delimited Data Files

See the summary sheet for descriptions of the comma delimited data files.

Required Files

When running SohReader the Orion message file SohReader.MSG must be in the current directory or in the same directory as the SohReader program. This file contains the error code definitions and is required to properly interpret the log entries. While this file contains the same information as ORION.MSG, it is not in the same format as ORION.MSG. Nanometrics DLLs version 1.8 or later (for Win32) are required.

Examples

SohReader RBFAAB.SOH

Extracts all information from the SOH file RBFAAB.SOH to the output files in the directory, RBFAAB. The directory will be in the current working directory.

SohReader RBFAAB.SOH -o dir

Extracts to output directory "dir/RBFAAB" all information in the SOH file. "dir" is relative to the current directory.

SohReader RBFAAB.SOH -o dir -s 1996-01-01-12-00-00 -d 60

Extracts all information to output directory "dir\RBFAAB" for 60 seconds beginning at 12:00:00 January 1, 1996. "dir" is relative to the current directory.

Environment

SohReader runs under Windows 9x/NT/2000 and Solaris.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\SohReader.fm
Date last revised: 2002-10-30

Name

SOHTOYP

Version

1.320

Synopsis

sohtoyp Arg1 [-o o1] [-s s1] [-e e1 | -d d1] [-y y1] [-f] [-n] [-u] [-m] [-w] [-b] [-c] [-g] [-x] [-h] [-k] [-i] [-p] [-a] [-q]

Arg1	SOH Ringbuffer file required argument (string)
-o	Output directory optional switch
o1	Output directory required switch argument (string)
-s	Start Time optional switch
s1	Start Time required switch argument (string)
-e	End Time optional switch
e1	End Time required switch argument (string)
-d	Duration optional switch
d1	Duration required switch argument (integer)
-y	File of Stream names optional switch
y1	File of Stream names required switch argument (string)
-f	Fast SOH optional switch
-n	Slow SOH optional switch
-u	Unscaled Slow SOH optional switch
-m	MinMax optional switch
-w	GPS location optional switch
-b	GPS error optional switch
-c	GPS channel optional switch
-g	GPS (all) optional switch
-x	VCXO optional switch
-h	HRD information optional switch
-k	Orion temperatures optional switch
-i	Orion voltages optional switch
-p	Orion powering optional switch
-a	Orion (all) optional switch
-q	GPS time quality optional switch

Type "-h" for syntax, "-h full" for full help, or "-h argid" for topical help.

Description and Usage

SOHTOYP is used to extract various streams of information from the binary SOH data file and convert these streams to Y5 file format output file sets. Each file set or channel is given a three character channel name as described below in table 1. A new file in a file set is started every time there is a time discontinuity (or tear). The file names are constructed by appending the channel name to the ring buffer name. The file extensions are numbered in sequence starting at 000.

For example, a ring buffer SOH file with name RORION.SOH will create a fast state of health 1 file set RORIONAF1.000, RORIONAF1.001 ...

The default action (if none of the flags [-y y1] [[-f] [-n] [-u] [-m] [-w] [-b] [-c] [-g] [-x] [-h] [-k] [-i] [-p] [-a] [-q] is used) is to extract all streams .

By default, all pertinent stream data in the file is extracted except if the -s switch is used either alone or in conjunction with either -e or -d switches, or the -e switch is used alone.

o Optional output directory switch.

Use this switch and its required parameter o1 to specify the output directory. If this switch is not used, the files will be placed in the current directory.

o1 Output directory parameter, required if o switch is specified.

Format: path1\path2for a relative path

\path1\path2for an absolute path

If this option is not used, the output is placed in the current working directory.

s Optional start time of data extraction switch.

Use this switch and its required parameter s1 to specify the start time of data extraction. If this switch is not used, extraction begins at the first record in the SOH file.

s1 Start time of data extraction parameter, required if s switch is specified.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1991-10-17-23-11-17

e Optional end time of data extraction switch.

Use this switch and its required parameter e1 to specify the start time of data extraction. If neither this switch nor the d switch is used, extraction continues to the last record in the SOH file. This switch cannot be used in conjunction with the d switch.

e1 End time of data extraction parameter, required if e switch is specified.

Format: YYYY-MM-DD-HH-MM-SS
e.g. 1991-10-17-23-11-17

d Optional duration of data extraction switch.

Use this switch and its required parameter d1 to specify the duration of data extraction. If neither this switch nor the e switch is used, extraction continues to the last record in the SOH file. This switch cannot be used in conjunction with the e switch.

d1 Duration of data extraction in seconds, required if d switch is specified.

Format: Integer
e.g. 3600

y Optional "file of stream names" file switch.

If this switch is used, the streams specified in the "file of streams" (y1 argument) are created.

y1 "File of Stream names" file, required if y switch is specified.

The balance of the switches determine which information stream groups are extracted (in addition to any selected by a "file of stream names file". A stream group is simply a collection of logically related streams.

State-of-Health File Set Descriptions

Table 1-1 State-of-Health file set descriptions

File Set Description	Channel Name	Units	Command Line Option
Fast State-of-Health 1	AF1	0 - 1023 counts	f
Fast State-of-Health 2	AF2	0 - 1023 counts	f
Fast State-of-Health 3	AF3	0 - 1023 counts	f
Fast State-of-Health 4 (New style)	AF4	millivolts	f
Fast State-of-Health 5 (New style)	AF5	millivolts	f
Fast State-of-Health 6 (New style)	AF6	millivolts	f
Unscaled Slow State-of-Health 1	AU1	0 - 1023 counts	u
Unscaled Slow State-of-Health 2	AU2	0 - 1023 counts	u
Unscaled Slow State-of-Health 3	AU3	0 - 1023 counts	u
Unscaled Slow State-of-Health 4	AU4	0 - 1023 counts	u
Unscaled Slow State-of-Health 5	AU5	0 - 1023 counts	u
Unscaled Slow State-of-Health 6	AU6	0 - 1023 counts	u
Unscaled Slow State-of-Health 7	AU7	0 - 1023 counts	u
Unscaled Slow State-of-Health 8	AU8	0 - 1023 counts	u
Battery Voltage (Scaled Slow SOH)	AS1	millivolts	n
Disk temperature (Scaled Slow SOH)	AS2	°C x 1000	n
Crystal temperature (Scaled Slow SOH)	AS3	°C x 1000	n
External channel 1 (Scaled Slow SOH)	AS4	units x 1000	n
External channel 2 (Scaled Slow SOH)	AS5	units x 1000	n
External channel 3 (Scaled Slow SOH)	AS6	units x 1000	n
External channel 1 (New style)	AS7	millivolts	n
External channel 2 (New style)	AS8	millivolts	n
External channel 3 (New style)	AS9	millivolts	n
MinMax 1	AM1	-8388608 to 8388607 counts	m
MinMax 2	AM2	-8388608 to 8388607 counts	m
VCXO value	AV1	counts x 100	x
Orion/GPS time difference (GPS time - VCXO time) at lock	AV2	microseconds x 10	x
Orion/GPS time difference (GPS time - VCXO time)	AV3	microseconds x 10	x
Frequency error	AV4	PPM x 10	x
Crystal temperature (same as AE5)	AV5	0 - 1023 counts	x
Locking status	AV6	See Note 1	x
Auxiliary processor status	AV7	See Note 2	x
GPS latitude	AL1	10E-4 degrees	w or g
GPS longitude	AL2	10E-4 degrees	w or g
GPS elevation	AL3	meters	w or g
GPS horizontal error estimate	AE1	meters	b or g
GPS vertical error estimate	AE2	meters	b or g
GPS time error estimate	AE3	meters	b or g
Orion/GPS time difference (GPS time - VCXO time)	AE4	-2.1E9 to 2.1E9	b or g
Crystal temperature (same as AV5)	AE5	0 - 1023 counts	b or g
GPS solution state	AC1	See Note 3	c or g
GPS number of satellites used	AC2	0 to 5	c or g
GPS utility channel	AC3	1 to 5	c or g
GPS figure of merit	AC4	See Note 4	c or g

Table 1-1 State-of-Health file set descriptions (Continued)

File Set Description	Channel Name	Units	Command Line Option
GPS satellite 1 PRN	A11	1 to 32	c or g
GPS satellite 1 signal to noise ratio	A12	0 to 63 dB	c or g
GPS satellite 1 activity	A13	See Note 5	c or g
GPS satellite 2 PRN	A21	1 to 32	c or g
GPS satellite 2 signal to noise ratio	A22	0 to 63 dB	c or g
GPS satellite 2 activity	A23	See Note 5	c or g
GPS satellite 3 PRN	A31	1 to 32	c or g
GPS satellite 3 signal to noise ratio	A32	0 to 63 dB	c or g
GPS satellite 3 activity	A33	See Note 5	c or g
GPS satellite 4 PRN	A41	1 to 32	c or g
GPS satellite 4 signal to noise ratio	A42	0 to 63 dB	c or g
GPS satellite 4 activity	A43	See Note 5	c or g
GPS satellite 5 PRN	A51	1 to 32	c or g
GPS satellite 5 signal to noise ratio	A52	0 to 63 dB	c or g
GPS satellite 5 activity	A53	See Note 5	c or g
Battery voltage	AH1	millivolts	h
VCXO temperature	AH2	Celsius x 10	h
Radio SNR	AH3	NOT IMPLEMENTED	h
AUX interface temperature	AT1	Celsius x 10	k or a
VCXO temperature	AT2	Celsius x 10	k or a
Disk temperature	AT3	Celsius x 10	k or a
External battery voltage	AI1	millivolts	i or a
Internal battery voltage	AI2	millivolts	i or a
Mains voltage	AI3	millivolts	i or a
Charge current	AP1	milliamps	p or a
HRD power supply voltage	AP2	millivolts	p or a
External battery status	AP3	See Note 6	p or a
Internal battery status	AP4	See Note 6	p or a
Mains status	AP5	See Note 7	p or a
Power source	AP6	See Note 8	p or a
User CPU state	AP7	See Note 9	p or a
Charger state	AP8	See Note 10	p or a
GPS on time	AQ1	seconds	q
GPS off time	AQ2	seconds	q
GPS time to lock	AQ3	seconds	q
Time difference at lock	AQ4	microseconds x 10	q
VCXO error	AQ5	PPM x 10	q
Temperature compensation offset	AQ6	counts x 1000	q
GPS off reason	AQ7	0-PLL error 1-GPS on time expired	q
GPS final navigation mode	AQ8	0-3D 1-2D 2-single satellite 3-search	q

Note 1

GPS status:

0	3D navigation
1	2D navigation
2	single satellite

3	searching for satellites
4	GPS engine powered off
5	error state

Note 2

PLL status:

0	Orion PLL fine locked to GPS
1	Orion PLL coarse locked to GPS
2	Orion free running using temperature compensation
3	Orion PLL test mode

Note 3

GPS solution state:

1	Cold Start (unlocked)
2	Acquisition (unlocked)
3	2-D Navigation locked
4	3-D Navigation locked

Note 4

GPS figure of merit: refers to expected horizontal position error e in meters as follows:

1	e < 7.5
2	7.5 < e <= 12.5
3	12.5 < e <= 25.0
4	25.0 < e <= 50.0
5	50.0 < e <= 250.0
6	250.0 < e <= 500.0
7	500.0< e <= 5000.0
9	solution not converging

Note 5

GPS satellite activity:

0	idle
1	searching
3	tracking

Note 6

External and internal battery status:

1	not present
2	in use
3	ready
5	charging
6	shorted cell
7	shorted cell in use

Note 7

Mains status:

1	not present
2	in use

Note 8

Power source:

1	mains
2	external
4	internal

Note 9

User CPU state:

0	off

1	AUX
2	heater

Note 10

Charger state:

0	off
1	low
3	fast

File of Streams File

As there are in excess of 50 SOH output streams, it is not feasible to select individual streams via command line switches. To find an easy way around this problem, sohtoyp supports a "file of stream names" file. The file is a plain text listing of whitespace delimited case insensitive channel names of the desired streams.

For example if the command

sohtoyp *.soh -O outdir -y streamnames

was issued and the file streamnames were as follows:

AS1 AS2 AL1 AV1

the data streams for the channels AS1, AS2, AL1, and AV1 will be created in directory outdir.

Examples

sohtoyp RBFAAB.SOH

Extracts all information from the SOH file RBFAAB.SOH to the output file sets. The output files sets will be in the same directory as RBFAAB.SOH.

sohtoyp RBFAAB.SOH -o dir -f -w

Extracts to output directory "dir" the fast SOH and GPS location group file sets. "dir" is relative to the current directory.

sohtoyp RBFAAB.SOH -o \dir -s 1996-01-01-12:00:00 -d 60 -g

Extracts to output directory "\dir" the GPS location, GPS error , and GPS channel file group streams for 60 seconds beginning at 12:00:00 January 1, 1996. "\dir" an absolute path.

sohtoyp RBFAAB.SOH -o \dir -s 1996-01-01-12:00:00 -d 60 -y streams

Extracts to output directory "\dir" whatever streams are listed in the file of streams file "streams" for 60 seconds beginning at 12:00:00 January 1, 1996. "\dir" an absolute path.

Environment

SOHEXTRP runs Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL version 1.8 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\sohtoyp.fm
Date last revised: 2002-10-30

Name

TAGGED FILE FORMAT

Description

The Nanometrics Tagged File Format is used by both Nanometrics Binary Pick Files and Nanometrics Y-files. It is designed to be quick and easy to read and write, especially using the C language. The tagged format allows the format to be extended without breaking backwards or forwards compatibility.

Tagged files are divided into records. Each record starts with a tag indicating the type of data in the record. Following the tag is the data. Each tag has the offset in bytes to the next tag so that if a program reading the file does not understand the type of data indicated by the tag, it can skip over it to the next tag. This preserves forward compatibility since older programs can read newer versions of the format as they can simply ignore any new records. This also preserves backward compatibility since newer programs can skip out-of-date records and look for the newer, replacement records.

There is no padding or alignment of the records in a tagged file. Each tag or block of data is written immediately following the last byte of the last record.

Each tagged file starts with a tag indicating the type of file. This tag has no data attached to it. It is immediately followed by the first data tag.

Data Types

The description of the file formats uses the following data types:

CHAR	signed 8 bit character
UCHAR	unsigned 8 bit character
SHORT	signed 16 bit integer
USHORT	unsigned 16 bit integer
LONG	signed 32 bit integer
ULONG	unsigned 32 bit integer
FLOAT	IEEE 32 bit floating point number
DOUBLE	IEEE 64 bit floating point number
REALTIME	DOUBLE containing the number of seconds since January 1, 1970
BOOL16	a 16 bit boolean value (integer) - either 0 (FALSE) or 1 (TRUE)
PTR	a 32 bit integer -- unused externally and should always be 0
UNIQUEID	a unique 32 bit integer that identifies an instance of a record

If there is an array of a data type this is indicated by square brackets containing the number of elements in the array, e.g. CHAR Name[13] indicates that Name is a character array containing 13 elements.

Some of the fields contain a string of characters. A string is defined as an array of characters. There are two types of strings used in the data files: zero terminated and blank padded.

Zero terminated strings (called ASCIIZ) are compatible with the C definition of a string. That is, an array of characters ending with an ASCII 0 (not the "0" character).

Blank padded strings (called BLANKPAD) are used when the entire array of characters must be printable. In this case there is no terminating zero. Every character in the array must be a printable so if an array entry is not used by the text it must be set to the space character.

The PTR field is only used internally by programs and never holds valid data. These fields should be set to zero when writing a tagged file and should be ignored when reading.

The UNIQUEID field is used to uniquely identify instances of records. In some cases a record needs to be associated with another record. This is done by assigning a unique number to the Self

field of the other record and then using this number in the first record. For example, an event record has to indicate which of its many solution records is the preferred one. It does this by giving the unique ID of the preferred solution in the event's PreferredSolution field. Each solution has a Self field with a unique number -- the solution whose Self field matches the PreferredSolution field is the preferred solution.

Tag Format

UCHAR	Format
UCHAR	Magic
USHORT	Type
LONG	NextTag
LONG	NextSame
LONG	Spare

Format	This is the byte order format for this data. Use the letter "I" for Intel format data (little endian) or the letter "M" for Motorola (big endian) format
Magic	This is a unique number that allows programs to check that this a valid tag. This number must be 31.
Type	This is the type of data attached to this tag. It must be one of the predefined tag types listed below.
NextTag	NextTag is the offset in bytes from the end of this tag to the start of the next tag. That means, the offset is the size of the data attached to this tag.
NextSame	NextSame is the offset in bytes from the end of this tag to the start of the next tag with the same type. If zero, there is no next tag with the same type.
Spare	Spare is added to pad the size of the tag to an even sixteen bytes. Also available for future use. Should always be zero.

Tag Types

The list below gives the tag types which have been defined so far. See Appendix B to see where and how the tags are used.

0	TAG_Y_FILE
1	TAG_STATION_INFO
2	TAG_STATION_LOCATION
3	TAG_STATION_PARAMETERS
4	TAG_STATION_DATABASE
5	TAG_SERIES_INFO
6	TAG_SERIES_DATABASE
7	TAG_DATA_INT32
8	TAG_PICK_FILE
9	TAG_UNASSOCIATED_PICKA
10	TAG_CRUSTAL_MODEL
11	TAG_CRUSTAL_LAYER
12	TAG_EVENTA
13	TAG_MAGNITUDE
14	TAG_PICKA
15	TAG_SOLUTION
16	TAG_HYPO_PARAMETERS
17	TAG_ASSOCIATION
18	TAG_STN_LOC_PARAMETERS
19	TAG_HYPO_STN_PARAMETERS
20	TAG_LOC_STN_PARAMETERS

21 TAG_LOC_PARAMETERS
22 TAG_X_FILE
23 TAG_DATA_STEIM
24 TAG_EVENT_COMMENTS
25 TAG_SOLUTION_COMMENTS
26 TAG_STATION_RESPONSE
27 TAG_PICKB
28 TAG_EVENTB
29 TAG_UNASSOCIATED_PICKB
30 TAG_RINGBUFFER_FILE
31 TAG_RINGBUFFER_INFO
32 TAG_RINGBUFFER_INDEX
33 TAG_RINGBUFFER_DATA
34 TAG_LOGBUFFER_FILE
35 TAG_LOGBUFFER_INFO
36 TAG_LOGBUFFER_DATA
37 TAG_SOHBUFFER_FILE
38 TAG_SOHBUFFER_INFO
39 TAG_SOHBUFFER_LABEL
40 TAG_SOHBUFFER_CALIB
41 TAG_SOHBUFFER_DATA
42 TAG_SKIP_DATA
43 TAG_END_MARKER

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\taggedfileformat.fm
Date last revised: 2002-10-30

Name

TRESPONSE

Version

1.01.00

Synopsis

tresponse [filename]

Description

TResponse generates a text file that contains all pertinent information on the response of an instrument connected to a Trident digitizer. This file may be used to generate channel response for seed files (using makeseed), or by other Nanometrics applications such as Atlas (for inverse instrument response, instrument simulation, etc.). The response file contains information on multiple stages, including the instrument, A2D converter, anti-alias filter, DC removal filter, and FIR filters used during decimation. Response for Trident are exact, while those for the default instruments are nominal values after damping. However, new instruments or more current information may be defined and added to the list of known instruments.

Environment

Any Microsoft Windows 9x or newer, Solaris

Usage

Running TRESPONSE with one optional command line argument

tresponse [filename]

invokes an interactive process to build a response file based on the user's input. The command line argument specifies the name of the output file to be created. If no name is specified, output is written to a file named seed.rsp.

The user is prompted to provide the following input parameters:

Instrument A list of defined instruments is shown. The list is read from the file instruments.txt as found in the current directory. If Other is chosen, the program will prompt for various parameters to define a new instrument. The new instrument will then be saved into instruments.txt and will be available next time tresponse is run.

Sample rate The sampling rate (samples/second) used by the digitizer. The supported values are: 10, 20, 40, 50, 100, 120, 200, 500, 1000.

DC Removal Frequency The corner frequency (in mHz) of the DC removal filter. The supported values are: 0, 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000.

Sensitivity The sensitivity of the Trident (counts/microvolt), as set in the user configuration.

Input Range (Vp-to-p) The input range setting (Volts peak-to-peak) for the Trident. Supported values are 40, 16, 8, 4, 2

Start Date The first date when the response is valid. This date should come before the date the response is used.

If Other is chosen as the instrument type, the user is then prompted to enter information about the new instrument. Once finished, the instrument will be added to the list in instruments.txt, available next time tresponse is run. The following questions are asked when defining a new instrument:

Instrument Name Name for the new instrument.

Ground Motion The type of ground motion measured by the instrument. Supported val-

	ues are m, m/s, and m/s/s (displacement, velocity, acceleration)
Gain	Enter the gain of the instrument, in Volts per unit of ground motion (V/m, V/(m/s), or V/(m/s/s) depending on the ground motion measured)
Gain Frequency	Enter the frequency (Hz) at which the instrument gain is measured. Acceptable frequencies range from 1.0 Hz to 5.0 Hz.
Impedance	Enter the instrument impedance (Ohms). Active instruments are usually 0. Passive instruments may generate significant impedance which may affect the corner frequency and gain of the anti-alias filter of the Trident.
Number of Zeroes	Enter the number of zeroes in the transfer function for the instrument. Follow up questions will ask for the real and imaginary parts of each zero.
Number of Poles	Enter the number of poles in the transfer function for the instrument. Follow up questions will ask for the real and imaginary parts of each pole.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\tResponse.fm
Date last revised: 2002-10-30

Name

UNZIP

To list/test/extract from a ZIP archive file

Note: This program is provided free of charge by Nanometrics for its customers. Our sole contribution is to distribute the program.

Synopsis

unzip [-cflptuvxz[ajnoqUV]] file[.zip] [filespec ...]

Arguments

file[.zip] Path of the ZIP archive. The suffix “.zip” is applied if the file specified does not exist. Note that self-extracting ZIP files are supported; just specify the ``.exe'' suffix yourself. [filespec] An optional list of archive members to be processed. Expressions may be used to match multiple members; be sure to quote expressions that contain characters interpreted by the operating system. See DESCRIPTION (below) for more details.

Options

-c	extract files to stdout/screen (``CRT'')
-f	freshen existing files (replace if newer); create none
-l	list archive files (short format)
-p	extract files to pipe; no informational messages
-t	test archive files
-u	update existing files; create new ones if needed
-v	list archive files (verbose format)
-x	extract files in archive (default)
-z	display only the archive comment

Modifiers

-a	convert to MS-DOS textfile format (CR LF), Mac format (CR), UNIX/VMS format (LF), OR from ASCII to EBCDIC, depending on your system (only use for TEXT files!)
-j	junk paths (don't recreate archive's directory structure)
-n	never overwrite existing files; don't prompt
-o	OK to overwrite files without prompting
-q	perform operations quietly (-qq => even quieter)
-s	[OS/2, MS-DOS] allow spaces in filenames (e.g., ``EA DATA. SF'')
-U	leave filenames uppercase if created under MS-DOS, VMS, etc.
-V	retain (VMS) file version numbers
-X	[VMS] restore owner/protection info (may require privileges)

Description

UnZip will list, test, or extract from a ZIP archive, commonly found on MSDOS systems. Archive member extraction is implied by the absence of the -c, -p, -t, -l, -v or -z options. All archive members are processed unless a filespec is provided to specify a subset of the archive members. The filespec is similar to an egrep expression, and may contain:

*	matches a sequence of 0 or more characters
?	matches exactly 1 character
\nnn	matches the character having octal code nnn
[...]	matches any single character found inside the brackets; ranges are specified by a beginning character, a hyphen, and an ending character. If an exclamation point or a carat (`!' or ``^') follows the left bracket, then the range of characters matched is complemented with respect to the ASCII character set (that is, anything except the characters inside the brackets is considered a match).

Environment Options

UnZip's default behavior may be modified via options placed in an environment variable. This can be done with any -n modifiers: in order to make UnZip quieter by default, orto make it always overwrite or never overwrite files as it extracts them. For example, to make UnZip act as quietly as possible, only reporting errors, one would use one of the following commands:

setenv UNZIP -qq	UNIX C shell
UNZIP=-qq; export UNZIP	UNIX Bourne shell
set UNZIP=-qq	OS/2 or MS-DOS
define UNZIP_OPTS "-qq"	VMS (quotes for LOWERCASE)

Environment options are, in effect, considered to be just like any other command-line options, except that they are effectively the first options on the command line. To over-ride an environment option, one may use the ``minus operator'' to remove it. For instance, to override one of the quiet-flags in the example above, use the command:

```
unzip --q[other options] zipfile
```

The first hyphen is the normal switch character, and the second is a minus sign, acting on the q option. Thus the effect here is to cancel a single quantum of quietness. To cancel both quiet flags, two (or more) minuses may be used:

```
unzip -x--q zipfile
```

or

```
unzip ---qx zipfile
```

(the two are equivalent). This may seem awkward or confusing, but it is reasonably intuitive: just ignore the first hyphen and go from there. It is also consistent with the behavior of UNIX nice(1).

Examples

To use UnZip to extract all members of the archive letters.zip, creating any directories as necessary:

```
unzip letters
```

To extract all members of letters.zip to the current directory:

```
unzip -j letters
```

To test letters.zip, printing only a summary message indicating whether the archive is OK or not:

```
unzip -tq letters
```

To extract to standard output all members of letters.zip whose names end in ``.tex'', converting to the local end-of-line convention and piping the output into more(1):

```
unzip -ca letters \*.tex | more
```

(The backslash before the asterisk is only required if the shell expands wildcards, as in UNIX; double quotes could have been used instead, as in the source example below.) To extract the binary file paper1.dvi to standard output and pipe it to a printing program:

```
unzip -p articles paper1.dvi | dvips
```

To extract all FORTRAN and C source files--*.f, *.c, *.h, Makefile (the double quotes are necessary only in UNIX and only if globbing is turned on):

```
unzip source.zip "*.[fch]" Makefile
```

To extract only newer versions of the files already in the current directory, without querying.

Note: Be careful of archives that contain no timezone information, and a ``newer'' file from an eastern timezone may, in fact, be older)

```
unzip -fo sources
```

To extract newer versions of the files already in the current directory and to create any files not already there (same caveat as previous example):

```
unzip -uo sources
```

In the last five examples, assume that UNZIP or UNZIP_OPTS is set to -q. To do a singly quiet listing:

```
unzip -l file
```

To do a doubly quiet listing:

```
unzip -ql file
```

To do a standard listing:

```
unzip --ql file
```

or

```
unzip -l-q file
```

or

```
unzip -l--q file
```

(extra minuses don't hurt).

Tips

The current maintainer, being a lazy sort, finds it very useful to define an alias ``tt'' for ``unzip -tq''. One may then simply type ``tt zipfile'' to test the archive, some-UnZip will report ``No errors detected in zipfile.zip,'' after which one may breathe a sigh of relief.

See Also

FUNZIP(1), ZIP(1), ZIPCLOAK(1), ZIPINFO(1), ZIPNOTE(1), ZIPSPLIT(1)

Authors

Samuel H. Smith, Carl Mascott, David P. Kirschbaum, Greg R. Roelofs, Mark Adler, Kai Uwe Rommel, Igor Mandrichenko, Johnny Lee, Jean-loup Gailly; Glenn Andrews, Joel Aycock, Allan Bjorklund, James Birdsall, Wim Bonner, John Cowan, Frank da Cruz, Bill Davidsen, Arjan de Vet, James Dugal, Jim Dumser, Mark Edwards, David Feinleib, Mike Freeman, Hunter Goatley, Robert Heath, Dave Heiland, Larry Jones, Kjetil J{ob Kemp, J. Kercheval, Alvin Koh, Bo Kullmar, Johnny Lee,Warner Losh, Fulvio Marino, Gene McManus, Joe Meadows, Mike O'Carroll, Humberto Ortiz-Zuazaga, Piet W. Plomp, Antonio Querubin Jr., Steve Salisbury, Georg Sassen, Jon Saxton, Hugh Schmidt, Martin Schulz, Charles Scripter, Chris Seaman, Richard Seay, Alex Sergejew, Cliff Stanford, Onno van der Linden, Jim Van Zandt, Antoine Verheijen, Paul Wells.

Versions

v1.2	15 Mar 89	Samuel H. Smith
v2.0	9 Sep 89	Samuel H. Smith
v2.x	fall 1989	many Usenet contributors
v3.0	1 May 90	Info-ZIP (DPK, consolidator)
v3.1	15 Aug 90	Info-ZIP (DPK, consolidator)
v4.0	1 Dec 90	Info-ZIP (GRR, maintainer)
v4.1	12 May 91	Info-ZIP
v4.2	20 Mar 92	Info-ZIP (zip-bugs subgroup; GRR, maint.)
v5.0	xx Aug 92	Info-ZIP (zip-bugs subgroup; GRR, maint.)

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\unzip.fm
Date last revised: 2002-10-30

Name

X5DECOMP

Version

1.005

Synopsis

x5decomp [-i input_file] [-o output_directory] [-v verbosity_mask]

Description

X5DECOMP decompresses each input_file in Nanometrics X5 data format to produce a Nanometrics Y5 data format file in output_directory. The Y-file produced has the same filename as the input filename except the first letter is changed to a Y (by convention X-file filenames start with a X and Y-files with a Y). The X-files contain data in compressed Steim blocks, as created by EXTRACTP, while the Y-files contain data in uncompressed 32 bit long integers.

Options

-i input_file	Input file specification for the X data files to be decompressed. This may contain both a directory and filename component; the filename component may contain wildcard characters. The default value is xdata/*.
-o output_directory	Output directory path in which to place the decompressed Y data files, expressed as either a relative or absolute directory name using a syntax appropriate for the operating system. The output directory will be created if it does not already exist. The default value is ydata.

Environment

x5decomp runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.7 or later.

See Also

EXTRACTP

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\x5decomp.fm
Date last revised: 2002-10-30

Name

XEXTRACT

Version

1.02

Synopsis

Xextract FileSpec OutputDir StartTime Duration

Description

XEXTRACT is used to extract a segment of data from an X5 file and save it to a new, smaller X5 file. This may be useful if you have saved data to large X5-files for archiving purposes, and now wish to extract event data to smaller files. The output files are standard X5 files, and use the standard Nanometrics long filename convention for X5-files:

Filename = "X" + station name (up to 5 characters)
+ "_" + channel name (up to 3 characters)
+ "_" + date and time (yyyymmdd.hhmmss)

Usage

XEXTRACT requires four command line arguments:

FileSpec	The pathname of the file (or files) from which to extract data.
OutputDir	The directory to which to write the output files.
StartTime	Start time of the data segment to be extracted, in the form yyyy-mm-dd-hh-mm-ss. The hours, minutes and seconds may be omitted.
Duration	Duration in seconds of the data segment to be extracted.

Example

The following command will extract 10 minutes (600 seconds) of data starting at 10:40 from each X5 file in the current directory, and store the resulting event files to directory Event1:

Xextract X* Event1 2000-02-07-10:40:00 600

Environment

XEXTRACT is supported on Windows 9x/NT/2000 and Solaris.

XEXTRACT requires Nanometrics DLLs release 1.7 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\xextract.fm
Date last revised: 2002-10-30

Name

Y5DUMP

Version

1.31

Synopsis

y5dump [-h] [-d] filename

Description

Y5DUMP is used to dump a version 5 X or Y-file into ASCII format. However, only the header of the X-file can be dumped as the data is compressed.

Options

[-h] - optional switch	dump the header (without this switch the file is just identified as a Y-file or not)
[-d] - optional switch	dump the data
filename	name of the Y-file or X-file to dump

Examples

To dump the header of the Y-file Y19921020.114055, type:

y5dump -h Y19921020.114055

Environment

Y5dump runs under Windows 9x/NT/2000 and Solaris.

This program requires Nanometrics DLL release 1.7 or later.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\y5dump.fm
Date last revised: 2002-10-30

Name

Y-FILE FORMAT

Description

This document defines the format used in the Nanometrics Y-file format version 5. There is a description of the physical format of the file and a description of the meaning of each field in the file.

You should read Tagged File Format before reading this manual page since a Y-file is an instance of a tagged file.

A Y-file always contains only one series of continuous data. If there a is break in the data, then you will need more than one Y-file to hold the data.

File Format

The first tag in a Y-file must be the TAG_Y_FILE tag. This must be followed by the following tags in any order:

TAG_STATION_INFO
TAG_STATION_LOCATION
TAG_STATION_PARAMETERS
TAG_STATION_DATABASE
TAG_SERIES_INFO
TAG_SERIES_DATABASE

The following tag is optional:

TAG_STATION_RESPONSE

Each tag must be followed by the data associated with the tag. See below for a description of the data for each tag.

The last tag in the file must be a TAG_DATA_INT32 tag. This tag must be followed by an array of LONG's. The number of entries in the array must agree with what was described in the TAG_SERIES_INFO data.

Some of the fields in the following structures do not have to be filled in correctly for DAN to able to load the file. The fields which are necessary are in bold and the optional fields are in italics.

Station ID

UCHAR Station[5]	(BLANKPAD)
UCHAR Location[2]	(BLANKPAD)
UCHAR Channel[3]	(BLANKPAD)
Station	Station is the five letter SEED format station identification.
Location	Location is the two letter SEED format location identification.
Channel	Channel is the three letter SEED format channel identification.

TAG_STATION_INFO

UCHAR	Update[8]
STNID	StationID
UCHAR	NetworkID[51] (ASCIIZ)
UCHAR	SiteName[61] (ASCIIZ)
UCHAR	Comment[31] (ASCIIZ)
UCHAR	SensorType[51] (ASCIIZ)
UCHAR	DataFormat[7] (ASCIIZ)
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.

StationID	StationID is the identification name of the station in SEED format. This uses a sub-record called STNID which is described above.
NetworkID	This is some descriptive text identifying the network.
SiteName	SiteName is some text identifying the site.
Comment	Comment is any comment for this station.
SensorType	SensorType is some text describing the type of sensor used at the station.
DataFormat	DataFormat is some text describing the data format recorded at the station.

TAG_STATION_LOCATION

UCHAR	Update[8]
FLOAT	Latitude
FLOAT	Longitude
FLOAT	Elevation
FLOAT	Depth
FLOAT	Azimuth
FLOAT	Dip
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.
Latitude	Latitude is the latitude in degrees of the location of the station. The latitude should be between -90 (South) and +90 (North).
Longitude	Longitude is the longitude in degrees of the location of the station. The longitude should be between -180 (West) and +180 (East).
Elevation	Elevation is the elevation in meters above sea level of the station.
Depth	Depth is the depth in meters of the sensor.
Azimuth	Azimuth is the azimuth of the sensor in degrees clockwise.
Dip	Dip is the dip of the sensor. 90 degrees is defined as vertical right way up.

TAG_STATION_PARAMETERS

UCHAR	Update[16]
REALTIME	StartValidTime
REALTIME	EndValidTime
FLOAT	Sensitivity
FLOAT	SensFreq
FLOAT	SampleRate
FLOAT	MaxClkDrift
UCHAR	SensUnits[24] (ASCIIZ)
UCHAR	CalibUnits[24] (ASCIIZ)
UCHAR	ChanFlags[27] (BLANKPAD)
UCHAR	UpdateFlag
UCHAR	Filler[4]
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.
StartValidTime	This is the time that the information in these records became valid.
EndValidTime	This is the time that the information in these records became invalid.
Sensitivity	Sensitivity is the sensitivity of the sensor in nanometers per bit.
SensFreq	This is the frequency at which the sensitivity was measured.
SampleRate	This is the number of samples per second. This value can be less than 1.0. (i.e. 0.1)
MaxClkDrift	This is the maximum drift rate of the clock in seconds per sample.
SensUnits	This is some text indicating the units in which the sensitivity was measured.
CalibUnits	This is some text indicating the units in which calibration input was measured.
ChanFlags	Text indicating the channel flags according to the SEED definition.
UpdateFlag	This flag must be "N" or "U" according to the SEED definition.
Filler	Pads out the record to satisfy the alignment restrictions for reading data on a

SPARC processor.

DATABASE
TAG_SERIES_DATABASE
TAG_STATION_DATABASE

UCHAR	Update[8]
REALTIME	LoadDate
UCHAR	Key[16]
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.
LoadDate	LoadDate is the date the information was loaded into the database.
Key	Key is a unique key that identifies this record in the database.

TAG_SERIES_INFO

UCHAR	Update[16]
REALTIME	StartTime
REALTIME	EndTime
ULONG	NumSamples
LONG	DCOffset
LONG	MaxAmplitude
LONG	MinAmplitude
UCHAR	Format[8] (ASCIIZ)
UCHAR	FormatVersion[8] (ASCIIZ)
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.
StartTime	This is start time of the data in this series.
EndTime	This is end time of the data in this series.
NumSamples	This is the number of samples of data in this series.
DCOffset	DCOffset is the DC offset of the data.
MaxAmplitude	MaxAmplitude is the maximum amplitude of the data.
MinAmplitude	MinAmplitude is the minimum amplitude of the data.
Format	This is the format of the data. This should always be "YFILE".
FormatVersion	FormatVersion is the version of the format of the data. This should always be "5.0"

TAG_STATION_RESPONSE

UCHAR	Update[8]
UCHAR	PathName[260]
Update	This field is only used internally by DAN for administrative purposes. It should always be set to zeroes.
PathName	PathName is the full name of the file which contains the response information for this station.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\Yfileformat.fm
Date last revised: 2002-10-30

Name

ZIP, ZIPCLOAK, ZIPNOTE, ZIPSPLIT

Package and compress (archive) files

Note: This program is provided free of charge by Nanometrics for its customers. Our sole contribution is to distribute the program.

Synopsis

zip [-cdeEfghjklmoqruwyz@] [-b temppath] [-n suffixes]
[-t mmddyy] [zipfile list] [-x list]
zipcloak [-d] [-b path] zipfile
zipnote [-w] [-b path] zipfile
zipsplit [-ti] [-n size] [-b path] zipfile

Description

Zip is a compression and file packaging utility for UNIX, VMS, MSDOS, OS/2, Windows NT, Minix, Atari and Macintosh. It is analogous to a combination of tar and compress and is compatible with PKZIP (Phil Katz ZIP) for MSDOS systems.

There is a companion to zip called unzip (of course) which you should be able to find the same place you got zip. Zip and unzip can work with files produced by PKZIP under MSDOS, and PKZIP and PKUNZIP can work with files produced by zip.

Zip version 1.9 is compatible with pkzip 1.93a. Note that pkunzip 1.10 cannot extract files produced by pkzip 1.93a or zip 1.9b. You must use pkunzip 1.93a or unzip 5.0 to extract them.

For a brief help on zip and unzip, run each without specifying any parameters on the command line.

Zip puts one or more compressed files into a single "zipfile" along with information about the files, including the name, path if requested, date and time last modified, protection, and check information to verify the fidelity of each entry. Zip can also be used as a filter, compressing standard input to standard output. Zip can pack an entire directory structure in a zip file with a single command. Compression ratios of 2:1 to 3:1 are common for text files.

Zip has one compression method (deflation) and can also store files without compression. It automatically chooses the better of the two for each file to be compressed.

Zip is useful for packaging a set of files to send to someone or for distribution; for archiving or backing up files; and for saving disk space by temporarily compressing unused files or directories.

How to Use Zip

The simplest use of zip is as follows: zip stuff *

This will create the file "stuff.zip" (assuming it does not exist) and put all the files in the current directory in stuff.zip in a compressed form. The .zip suffix is added automatically, unless that file name given contains a dot already. This allows specifying suffixes other than ".zip".

Because of the way the shell does filename substitution, files that start with a "." are not included. To include those as well, you can: zip stuff .* *

Even this will not include any sub directories that are in the current directory. To zip up an entire directory, the command: zip -r foo foo will create the file "foo.zip" containing all the files and directories in the directory "foo" that is in the current directory. (The first "foo" denotes the zip file, the second one denotes the directory.) The "r" option means recurse through the directory structure. In this case, all the files and directories in foo are zipped, including the ones that start with a ".", since the recursion does not use the shell's file-name substitution. You should not use -r with the

name ".*", since that matches ".." which will attempt to zip up the parent directory -- probably not what was intended.

You may want to make a zip file that contains the files in foo, but not record the directory name, foo. You can use the -j (junk path) option to leave off the path: zip -j foo foo/*

The -y option (only under UNIX) will store symbolic links as such in the zip file, instead of compressing and storing the file referred to in the link.

You might be zipping to save disk space, in which case you could: zip -rm foo foo where the "m" option means "move". This will delete foo and its contents after making foo.zip. No deletions will be done until the zip has completed with no errors. This option is obviously more dangerous and should be used with care.

If the zip file already exists, these commands will replace existing or add new entries to the zip file. For example, if you were really short on disk space, you might not have enough room simultaneously to hold the directory foo and the compressed foo.zip. In this case, you could do it in steps. If foo contained the sub directories tom, dick, and harry, then you could:

```
zip -rm foo foo/tom
zip -rm foo foo/dick
zip -rm foo foo/harry
```

where the first command would create foo.zip, and the next two would add to it. At the completion of each zip command, the directory just zipped would be deleted, making room in which the next zip command could work.

Zip will also accept a single dash ("-") as the zip file name, in which case it will write the zip file to stdout, allowing the output to be piped to another program. For example: zip -r - . | dd of=/dev/nrst0 obs=16k would write the zip output directly to a tape with the specified block size for the purpose of backing up the current directory.

Zip also accepts a single dash ("-") as the name of a file to be compressed, in which case it will read the zip file from stdin, allowing zip to take input from another program. For example: tar cf - . | zip backup - would compress the output of the tar command for the purpose of backing up the current directory. This generally produces better compression than the previous example using the -r option, because zip can take advantage of redundancy between files. The backup can be restored using the command unzip -p backup | tar xf -

When no zip file name is given and stdout is not a terminal, zip acts as a filter, compressing standard input to standard output. For example, tar cf - . | zip | dd of=/dev/nrst0 is equivalent to tar cf - . | zip - - | dd of=/dev/nrst0

Zip archives created in this manner can be extracted with the program funzip which is provided in the unzip package. For example, dd if=/dev/nrst0 | funzip | tar xvf -

Modifying Existing Zip Files

When given the name of an existing zip file with the above commands, zip will replace identically named entries in the zip file or add entries for new names. For example, if foo.zip exists and contains foo/file1 and foo/file2, and the directory foo contains the files foo/file1 and foo/file3, then: zip -r foo foo will replace foo/file1 in foo.zip and add foo/file3 to foo.zip. After this, foo.zip contains foo/file1, foo/file2, and foo/file3, with foo/file2 unchanged from before.

When changing an existing zip file, zip will write a temporary file with the new contents, and only replace the old one when the zip has completed with no errors. You can use the -b option to specify a different path (usually a different device) to put the temporary file in. For example: zip -b /tmp stuff * will put the temporary zip file and the temporary compression files in the directory "/tmp", copying over stuff.zip in the current directory when done.

If you are only adding entries to a zip file, not replacing, and the -g option is given, then zip grows

(appends to) the file instead of copying it. The danger of this is that if the operation fails, the original zip file is corrupted and lost.

There are two other ways to change or add entries in a zip file that are restrictions of simple addition or replacement. The first is -u (update) which will add new entries to the zip file as before but will replace existing entries only if the modified date of the file is more recent than the date recorded for that name in the zip file. For example: zip -u stuff * will add any new files in the current directory, and update any changed files in the zip file stuff.zip. Note that zip will not try to pack stuff.zip into itself when you do this. Zip will always exclude the zip file from the files on which to be operated.

The second restriction is -f (freshen) which, like update, will only replace entries with newer files; unlike update, will not add files that are not already in the zip file. For this option, you may want to simply freshen all of the files that are in the specified zip file. To do this you would simply: zip -f foo.

Note that the -f option with no arguments freshens all the entries in the zip file. The same is true of -u, and hence "zip -u foo" and "zip -f foo" both do the same thing.

This command should be run from the same directory from which the original zip command was run, since paths stored in zip files are always relative.

Another restriction that can be used with adding, updating, or freshening is -t (time), which will not operate on files modified earlier than the specified date. For example: zip -rt 120791 infamy foo will add all the files in foo and its sub directories that were last modified on December 7, 1991, or later to the zip file infamy.zip.

Also, files can be explicitly excluded using the -x option: zip -r foo foo -x *.o which will zip up the contents of foo into foo.zip but exclude all the files that end in ".o". Here the backslash causes zip to match file names that were found when foo was searched.

The last operation is -d (delete) which will remove entries from a zip file. An example might be: zip -d foo foo/tom/junk foo/harry/* *.o which will remove the entry foo/tom/junk, all of the files that start with "foo/harry/", and all of the files that end with ".o" (in any path). Note that once again, the shell expansion has been inhibited with backslashes, so that zip can see the asterisks. zip can then match on the contents of the zip file instead of the contents of the current directory.

Under MSDOS, -d is case sensitive when it matches names in the zip file. This allows deleting names that were zipped on other systems, but requires that the names be entered in upper case if they were zipped on an MSDOS system, so that the names can be found in the zip file and deleted.

More Options

As mentioned before, zip will use the best of two methods: deflate or store.

The option -0 will force zip to use store on all files. For example: zip -r0 foo foo will zip up the directory foo into foo.zip using only store.

The speed of deflation can also be controlled with options -1 (fastest method but less compression) to -9 (best compression but slower). The default value is -5. For example: zip -r8 foo foo. In nearly all cases, a file that is already compressed cannot be compressed further by zip, or if it can, the effect is minimal. The -n option prevents zip from trying to compress files that have the given suffixes. Such files are simply stored (0% compression) in the output zip file, so that zip doesn't waste its time trying to compress them. The suffixes are separated by either colons or semicolons. For example: zip -rn ".Z:.zip:.tiff:.gif:.snd" foo foo will put everything in foo into foo.zip, but will store any files that end in .Z, .zip, .tiff, .gif, or .snd without trying to compress them. (Image and sound files often have their own specialized compression methods.) The default suffix list is ".Z:.zip;.zoo:.arc:.lzh:.arj". The environment variable ZIPOPT can be used to change this default. For example under UNIX with csh: setenv ZIPOPT "-n .gif:.zip". The variable ZIPOPT can be used for any option and can include several options.

Under UNIX and under OS/2 (if files from an HPFS are stored), zip will store the full path (relative to the current path) and name of the file (or just the name if -j is specified) in the zip file along with the UNIX attributes, and it will mark the entry as made under UNIX.

If the zip file is intended for PKUNZIP under MSDOS, then the -k (Katz) option should be used to attempt to convert the names and paths to conform to MSDOS, store only the MSDOS attribute (just the user write attribute from UNIX), and mark the entry as made under MSDOS (even though it wasn't).

The -o (older) option will set the "last modified" time of the zip file to the latest "last modified" time of the entries in the zip file. This can be used without any other operations, if desired. For example: zip -o foo will change the last modified time of foo.zip to the latest time of the entries in foo.zip.

The -e and -c options operate on all files updated or added to the zip file. Encryption (-e) will prompt for a password on the terminal and will not echo the password as it is typed (if stderr is not a TTY, zip will exit with an error). New zip entries will be encrypted using that password. For added peace of mind, you can use -ee, which will prompt for the password twice, checking that the two are the same before using it. The encryption code is distributed separately, so the -e option may not be available in your version.

One-line comments can be added for each file with the -c option. The zip file operations (adding or updating) will be done first, and you will then be prompted for a one-line comment for each file. You can then enter the comment followed by return, or just return for no comment.

The -z option will prompt you for a multi-line comment for the entire zip file. This option can be used by itself, or in combination with other options. The comment is ended by a line containing just a period, or an end of file condition (^D on UNIX, ^Z on MSDOS, OS/2, and VAX/VMS). Since -z reads the lines from stdin, you can simply take the comment from a file: zip -z foo < foowhat

The -q (quiet) option eliminates the informational messages and comment prompts while zip is operating. This might be used in shell scripts, for example, or if the zip operation is being performed as a background task ("zip -q foo *.c &").

Zip can take a list of file names to operate on from stdin using the -@ option. In UNIX, this option can be used with the find command to extend greatly the functionality of zip. For example, to zip up all the C source files in the current directory and its sub directories, you can: find . -type f -name "*.[ch]" -print | zip source -@.

Note: The pattern must be quoted to keep the shell from expanding it.

Under VMS only, the -w option will append the version number of the files to the name and zip up multiple versions of files. Without -w, zip will only use the most recent version of the specified file(s).

The -l option translates the UNIX end-of-line character LF into the MSDOS convention CR LF. This option should not be used on binary files. This option can be used on UNIX if the zip file is intended for PKUNZIP under MSDOS. If the input files already contain CR LF, this option adds an extra CR. This ensure that "unzip -a" on UNIX will get back an exact copy of the original file, to undo the effect of "zip-l".

If zip is run with the -h option, or with no arguments and standard output is a terminal, the license and the command-argument and option help is shown. The -L option just shows the license.

About Pattern Matching

Note: This section applies to UNIX. Watch this space for details on MSDOS and VMS operation.

The UNIX shell (sh or csh) does filename substitution on command arguments. The special characters are ?, which matches any single character; * which matches any number of characters (including none); and [] which matches any character in the range inside the brackets (like [a-f] or [0-9]). When these characters are encountered (and not escaped with a backslash or quotes), the shell will look for files relative to the current path that match the pattern, and replace the argument with a list of the names that matched.

Zip can do the same matching on names that are in the zip file being modified or, in the case of the -x (exclude) option, on the list of files to be operated on, by using backslashes or quotes to tell the shell not to do the name expansion. In general, when zip encounters a name in the list of files to do, it first looks for the name in the file system. If it finds it, it then adds it to the list of files to do. If it does not find it, it will look for the name in the zip file being modified (if it exists), using the pattern matching characters above, if any. For each match, it will add that name to the list of files to do. After -x (exclude), the names are removed from the to-do list instead of added.

The pattern matching includes the path, and so patterns like *.o match names that end in ".o", no matter what the path prefix is. Note that the backslash must precede every special character (i.e. ?*[]), or the entire argument must be enclosed in double quotes ("").

In general, using backslash to make zip do the pattern matching is used with the -f (freshen) and -d (delete) options, and sometimes after the -x (exclude) option when used with any operation (add, -u, -f, or -d). zip will never use pattern matching to search the file system. If zip has recursed into a directory, all files (and all directories) in there are fair game.

Copyright

Acknowledgements

Thanks to R. P. Byrne for his Shrink.Pas program which inspired this project; to Phil Katz for making the zip file format, compression format, and .zip filename extension all public domain; to Steve Burg and Phil Katz for help on unclear points of the deflate format; to Keith Petersen and Rich Wales for providing a mailing list and ftp site for the INFO-ZIP group to use; and most importantly, to the INFO-ZIP group itself (listed in the file infozip.who) without whose tireless testing and bug-fixing efforts a portable zip would not have been possible. Finally we should thank (blame) the INFO-ZIP moderator, David Kirschbaum for getting us into this mess in the first place.

See Also

UNZIP(1), TAR(1), COMPRESS(1)

Bugs

Warning: zip files produced by this version of zip must not be *updated* by zip 1.0 or pkzip 1.10 or pkzip 1.93a, if they contain encrypted members, or if they have been produced in a pipe or on a non seekable device. The old versions of zip or pkzip would destroy the zip structure. The old versions can list the contents of the zip file but cannot extract it anyway (because of the new compression algorithm). If you do not use encryption and use regular disk files, you do not have to care about this problem.

Zip 1.9 is compatible with pkzip 1.93a, except when two features are used: encryption or zip file created in a pipe or on a non seekable device. pkzip versions above 2.0 will support such files, and

unzip 5.0 already supports them.

Without -y, when zip must compress a symbolic link to an non existing file, it only displays a warning "name not matched". A better warning should be given.

Under VMS, not all of the odd file formats are treated properly. Only zip files of format stream-LF and fixed length 512 are expected to work with zip. Others can be converted using Rahul Dhesi's BILF program. This version of zip does handle some of the conversion internally. When using Kermit to transfer zip files from Vax to MSDOS, type "set file type block" on the Vax. When transferring from MSDOS to Vax, type "set file type fixed" on the Vax. In both cases, type "set file type binary" on MSDOS.

Under VMS, zip hangs for file specification that uses DECnet syntax (foo::*.*).

Under OS/2, the amount of External Attributes displayed by DIR is (for compatibility) the amount returned by the 16-bit version of DosQueryPathInfo(). Otherwise OS/2 1.3 and 2.0 would report different EA sizes when DIRing a file. However, the structure layout returned by the 32-bit DosQueryPathInfo() is a bit different, it uses extra padding bytes and link pointers (it's a linked list) to have all fields on 4-byte boundaries for portability to future RISC OS/2 versions. Therefore the value reported by ZIP (which uses this 32-bit-mode size) differs from that reported by DIR. ZIP stores the 32-bit format for portability, even the 16-bit MS-C-compiled version running on OS/2 1.3, so even this one shows the 32-bit-mode size.

Note: LIKE ANYTHING ELSE THAT'S FREE, ZIP AND ITS ASSOCIATED UTILITIES ARE PROVIDED AS IS AND COME WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED. IN NO EVENT WILL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES RESULTING FROM THE USE OF THIS SOFTWARE.

That having been said, please send any problems or comments via e-mail to the Internet address zip-bugs@cs.ucla.edu. For bug reports, please include the version of zip, the make options you used to compile it, the machine and operating system you are using, and as much additional information as possible. Thank you for your support.

This document information

G:\Manuals & graphics\Manuals\ReferenceManual\PCSoftware\Playback\3.8\R1\zip.fm
Date last revised: 2002-10-30