dagflood User Manual

2.5.5r1

EDM04.05-03r1

Endace Software CD-ROM 2.5.3 r1
www.endace.com
Copyright ©2005 Endace Measurement Systems. All rights reserved.
Copying of this disc is prohibited.

Endace Measurement Systems | New Zealand | www.endace.com

Leading Network Intelligence

Published by:

Endace Measurement Systems® Ltd
Building 7
17 Lambie Drive
PO Box 76802
Manukau City 1702
New Zealand
Phone: +64 9 262 7260
Fax: +64 9 262 7261
support@endace.com
www.endace.com

International Locations

New Zealand	Americas	Europe, Middle East & Africa
Endace Technology® Ltd	Endace USA® Ltd	Endace Europe® Ltd
Level 9	Suite 220	Sheraton House
85 Alexandra Street	11495 Sunset Hill Road	Castle Park
PO Box 19246	Reston	Cambridge CB3 0AX
Hamilton 2001	Virginia 20190	United Kingdom
New Zealand	United States of America	Phone: ++44 1223 370 176
Phone: +64 7 839 0540	Phone: ++1 703 382 0155	Fax: ++44 1223 370 040
Fax: +64 7 839 0543	Fax: ++1 703 382 0155	support@endace.com
support@endace.com	support@endace.com	www.endace.com
www.endace.com	www.endace.com	

Typographical Conventions Used in this Document

- Command-line examples suitable for entering at command prompts are displayed in `mono-space courier font`. The font is also used to describe config file data used as examples within a sentence. An example can be in more than one sentence.

  Results generated by example command-lines are also displayed in `mono-space courier font.`

- The software version references such as 2.3.x, 2.4.x, 2.5.x are specific to Endace Measurement Systems and relate to Company software products only.

Protection Against Harmful Interference

When present on product this manual pertains to and indicated by product labelling, the statement "This device complies with part 15 of the FCC rules" specifies the equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the Federal Communications Commission [FCC] Rules.

These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment.

This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications.

Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Extra Components and Materials

The product that this manual pertains to may include extra components and materials that are not essential to its basic operation, but are necessary to ensure compliance to the product standards required by the United States Federal Communications Commission, and the European EMC Directive. Modification or removal of these components and/or materials, is liable to cause non compliance to these standards, and in doing so invalidate the user’s right to operate this equipment in a Class A industrial environment.

Table of Contents

1.0 ARGUMENTS USED WITH DAGFLOOD 1
1.1 `dagflood` Usage... 2
1.2 Card Initialisation... 3
1.3 `dagflood` File Transmit Examples 3

1.0 ARGUMENTS USED WITH DAGFLOOD

Introduction

One of the programs able to use the transmit capabilities of the DAG cards is *dagflood*.

The purpose of *dagflood* is to transmit packets stored in a file at a full rate. Packets in the file should be defined in Extensible Record Format [ERF].

There are two ways to have a file with ERF packets. One is to use *dagsnap*, which captures received in a DAG card into a file. In this case *dagconvert* is used to make packets 64 bit aligned.

The other way is to use *daggen to* construct a file with ERF packets. *daggen* is a packet constructor which reads packet definitions from a config file and writes them to a file. The complete manual of *daggen* can be found at the *doc* directory of DAG software release.

Transmit packets at full rate requires a fast read from where the packets are stored. Accessing a file at disk is a slow process. Access speed to the file is increased by having it in the memory. Dagflood will put the file with the packets into memory. If a 1 GB file is to be transmitted at least 1 GB of physical memory is required on the system. Dagflood provides options to repeat the file contents many times.

Dagflood is a good starting point for understanding how the DAG API transmit functions work.

In this chapter

This chapter covers the following sections of information.

- dagflood Usage
- Card Initialisation
- dagflood File Transmit Examples

1.1 dagflood Usage

Description The following arguments can be used with `dagflood`:

```
usage: dagflood –f <filename> [options]

 Arguments:
    <file> file containing ERF records to send

Options:
-c   <count>          transmit file <count> time before exiting this page
-h,  --help, --usage
-V,  --version        display version information
-v,  --verbose        increase verbosity
-d   <device>         use dag device <device>
                          default: /d/dag0
-t   <n>              set program to terminate after <n> seconds
                          default: 0 (continue indefinitely)
-l                    set burst_max (maximum data burst length0
                             default: 1 MiB
-u                    inter burst delay (microseconds)
-x                    don’t flush transmit buffer when a signal is caught
-r   <n>              API mode:
                          1: commit bytes
                          2: copy bytes
```

The use of a file containing the packets is mandatory. If the contents of a file is to be transmitted several times, a choice can be made to either:

- Repeat the contents a number of times (`-c option`)
- Repeat the contents during a certain amount of seconds (`-t option`)

With the `-l` option the size of the chunks copied are controlled every burst. With the `-u` option a delay can be specified between two chunk copies. Changing these two parameters provide control over transmit speed. By default they are optimized for full rate transmits.

If required, the `-V` option is used to update reports about the transmit every second.

There are two ways to run dagflood in function of the API implementation by choosing the `-r` option. This is provided only for educational purposes to demonstrate two different ways to do the same thing.

1.2 Card Initialisation

Description

DAG cards must be initialized correctly to use the transmit capabilities. This involves checking the card's firmware has transmit capabilities using one of the card's configuration tools, `dagthree`, `dagfour`, `dagsix` or `dagseven`.

The output of the tool must report at least one transmit stream, `txt streams=1`.

The default parameters of these tools give the minimum recommended amount of buffer space for transmit streams.

If the reserved memory is 128 MB per card, a call like `dagfour default` would split that space unevenly, `mem=120:8`, 120 MB for receive buffering and 8 MB for transmit buffering.

Using the above configuration will still require more buffering space for transmit if a transmission will be at full rate, or the machine is heavily loaded. In such cases the user can split space in two halves, `mem=64:64`.

If the receive capabilities are not being used then all space can be given to transmit streams `mem=0:128`.

Increasing buffer size can also result in an increased transmit delay.

The initialization of a DAG4 card would appear as:

```
dagfour –d dag0 default mem=64:64
```

1.3 `dagflood` File Transmit Examples

Description

The examples of `dagflood` file transmits include sending file once and one hundred times. There are examples of transmitting a file for sixty seconds, transmitting one slowly, and transmitting a file continuously while watching reports.

Other examples include transmitting files captured with a `dagsnap` and files generated with `daggen`.

Continued on next page

1.3 dagflood File Transmit Examples, continued

Examples The following

Sending a File	Description
Once.	`dagflood –f file.erf –d dag0`
One hundred times.	`dagflood –f file.erf –d dag0 –c 100`
For 60 seconds.	`dagflood –f file.erf –d dag0 –t 100`
Slowly.	`dagflood –f file.erf –d dag0 –l 1024 –n 500`
While watching reports.	`dagflood –f file.erf –d dag0 –v`

Captured with a `dagsnap`.	`dagsnap –d dag0 –s 5 –o file1.erf`
	`dagconvert –i file1.erf –o file2.erf –A 8`
	`dagflood –f file2.erf –d dag1`

Generated with `daggen`.	`daggen –f traffic –x traffic_group_1 –o file.er`
	`dagflood –f file.erf –d dag0`

NOTE: See `daggen` manual for further details.	