dagflood User Guide

EDM04-03

Protection Against Harmful Interference

When present on equipment this document pertains to, 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 document, 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 their 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.

Disclaimer

Whilst every effort has been made to ensure accuracy, neither Endace Technology Limited nor any employee of the company, shall be liable on any ground whatsoever to any party in respect of decisions or actions they may make as a result of using this information.

Endace Technology Limited has taken great effort to verify the accuracy of this document, but nothing herein should be construed as a warranty and Endace shall not be liable for technical or editorial errors or omissions contained herein.

In accordance with the Endace Technology Limited policy of continuing development, the information contained herein is subject to change without notice.

Website

http://www.endace.com

Copyright 2011 Endace Technology Ltd. All Rights reserved.

Contents

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 initialization
- `dagflood` File Transmit Examples

dagflood Usuage

The following arguments can be used with `dagflood`:

Option	Description
`-?,--usage,-h,--help`	Display help (this page).
`-c,--count <count>`	Transmit file <count> times before exiting.
`-d,--device <device>`	DAG device to use. Default: `dag0`.
`-f,--fname <filename>`	File containing ERF records to send.
`-l,--burst-max <bytes>`	Set burst_max (Maximum data burst length). Default: 1 MiB. ***Note:*** *The value specified for Maximum data burst length needs to be divisible by 8.*
`-r,--mode <api-mode>`	API mode. 1 = commit bytes, 2 = copy bytes.
`-t,--terminate <seconds>`	Set program to terminate after <n> seconds. Default: 0 (continue indefinitely).
`-u,--delay <microseconds>`	Inter burst delay (microseconds).
`-v,--verbose`	Increase verbosity.
`-V,--version`	Display version information.
`-x,--no-flush`	Do not flush transmit buffer when a signal is caught.

The use of a file containing the packets is mandatory. If the content 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.

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.

Card Initalization

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` or `dagconfig`.

The output of the tool must report at least one transmit stream, `txtstreams=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=64:64`.

Increasing buffer can also result in an increased transmit delay.

The initialization of a DAG card would appear as:

```
dagfour –d dag0 default mem=0:128
```

dagflood File Transmit Examples

The following examples show how to transmit using `dagflood`.

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 60`
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:
Refer to the EDM04-06 daggen User Manual for further details.

Version History

Version	Date	Reason
1		Previous version.
2	August 2005	
3	September 2007	New template and some sections revised.
4	June 2008	Correct card initialization information and dagflood file transmit examples. Updated copyright information.
5	November 2008	Updated dagflood option –l.
6	November 2011	Update branding.

endace.com