

# datalinkclient(1)

## Name

datalinkclient - Example client to stream miniseed over datalink.

## Synopsis

**datalinkclient** [-vV] [--help] [-h=<host>] [-m=<match>] [--max=<maxRecords>] [-o=<outputFile>] [-p=<port>] [--timeout=<timeoutSec>]

## Description

Datalinkclient is an example client that can connect to a datalink service and stream miniseed to a file. [View source code here.](#)

The datalink protocol is specified here:

<https://iris-edu.github.io/libdali/datalink-protocol.html>

## Options

**-h, --host=<host>**

host to connect to, defaults to IRIS, rtserve.iris.washington.edu

**--help**

display a help message

**-m, --match=<match>**

match pattern, as a regular expression. For miniseed the conventions is NN\_SSS\_LL\_CCC/MSEED

**--max=<maxRecords>**

number of packets to receive before ending the connection, defaults to 10

**-o, --out=<outputFile>**

Output file (default: print to console)

**-p, --port=<port>**

port to connect to, defaults to IRIS, 18000

**--timeout=<timeoutSec>**

timeout seconds, defaults to 120

**-v, --verbose**

Verbose

**-V, --version**

Print version and exit

## Examples

Ask for a few packets of HH. data from station CO.BIRD, and output to a file.

```
> datalinkclient -m CO_BIRD_00_HH./MSEED --max 3 -o bird.mseed -v
initConnection to rtserve.iris.washington.edu:18000
Connection made
sendId
send 'ID seisFile:unknown:42:java | '
not bytes available 0
readPacket
readPacket(): blocking read for 3 bytes, available=53
Server ID: ID DataLink 2020.075 :: DLPROTO:1.0 PACKETSIZE:512
send 'MATCH | CO_BIRD_00_HH./MSEED'
send 'STREAM | '
readPacket(): blocking read for 3 bytes, available=0
Response: OK
readPacket(): blocking read for 3 bytes, available=0
D CO.BIRD.00.HHN 2023,249,17:32:40.6800 3.31 331
readPacket(): blocking read for 3 bytes, available=8470
D CO.BIRD.00.HHN 2023,249,17:32:43.9900 3.47 347
send 'ENDSTREAM | '
readPacket(): blocking read for 3 bytes, available=7865
D CO.BIRD.00.HHN 2023,249,17:32:47.4600 3.47 347
readPacket(): blocking read for 3 bytes, available=7260
D CO.BIRD.00.HHN 2023,249,17:32:50.9300 3.57 357
readPacket(): blocking read for 3 bytes, available=6655
D CO.BIRD.00.HHN 2023,249,17:32:54.5000 3.57 357
readPacket(): blocking read for 3 bytes, available=6050
D CO.BIRD.00.HHN 2023,249,17:32:58.0700 3.45 345
readPacket(): blocking read for 3 bytes, available=5445
D CO.BIRD.00.HHN 2023,249,17:33:01.5200 3.49 349
readPacket(): blocking read for 3 bytes, available=4840
D CO.BIRD.00.HHN 2023,249,17:33:05.0100 3.52 352
readPacket(): blocking read for 3 bytes, available=4235
D CO.BIRD.00.HHN 2023,249,17:33:08.5300 3.42 342
readPacket(): blocking read for 3 bytes, available=3630
D CO.BIRD.00.HHN 2023,249,17:33:11.9500 3.56 356
readPacket(): blocking read for 3 bytes, available=3025
D CO.BIRD.00.HHZ 2023,249,17:32:34.3700 3.44 344
readPacket(): blocking read for 3 bytes, available=2420
D CO.BIRD.00.HHZ 2023,249,17:32:37.8100 3.39 339
readPacket(): blocking read for 3 bytes, available=1815
```

```
D CO.BIRD.00.HHZ 2023,249,17:32:41.2000 3.38 338
readPacket(): blocking read for 3 bytes, available=1210
D CO.BIRD.00.HHZ 2023,249,17:32:44.5800 3.6 360
readPacket(): blocking read for 3 bytes, available=605
D CO.BIRD.00.HHZ 2023,249,17:32:48.1800 3.73 373
readPacket(): blocking read for 3 bytes, available=0
D CO.BIRD.00.HHZ 2023,249,17:32:51.9100 3.64 364
readPacket(): blocking read for 3 bytes, available=2420
D CO.BIRD.00.HHZ 2023,249,17:32:55.5500 3.92 392
readPacket(): blocking read for 3 bytes, available=1815
D CO.BIRD.00.HHZ 2023,249,17:32:59.4700 3.78 378
readPacket(): blocking read for 3 bytes, available=1210
D CO.BIRD.00.HHZ 2023,249,17:33:03.2500 3.76 376
readPacket(): blocking read for 3 bytes, available=605
D CO.BIRD.00.HHZ 2023,249,17:33:07.0100 3.69 369
readPacket(): blocking read for 3 bytes, available=0
```