

\$> jive5ab

<https://github.com/jive-vlbi/jive5ab.git>

Track 3.2.0-dev branch

- address compiler warnings
(reports by users with newer/newest compilers such as gcc16)
- add "startup-file" command line option
\$> jive5ab ... -i <path/to/file> ...

Track 3.2.0-dev branch

- address compiler warnings
(reports by users with newer/newest compilers such as gcc16)
- add "startup-file" command line option
\$> jive5ab ... -i <path/to/file> ...
\$> cat jive5ab.conf
mtu=9000;
set_disks=/path/to/my/raid/mnt*;
net_protocol=:128M;

Track 3.2.0-dev branch

- address compiler warnings
(reports by users with newer/newest compilers such as gcc16)

- add "startup-file" configuration option

```
$> jive5ab ... -i <path/to/file> ...
```

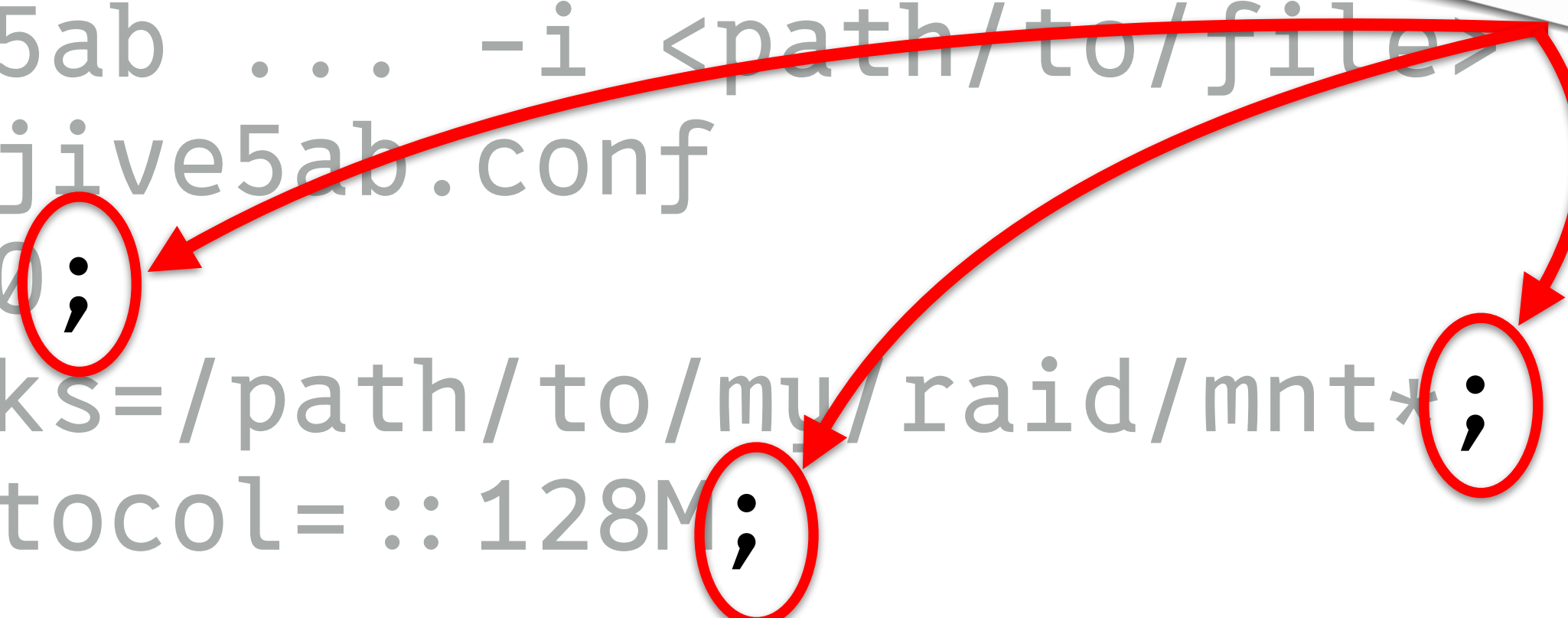
```
$> cat jive5ab.conf
```

```
mtu=9000;
```

```
set_disks=/path/to/my/raid/mnt*;
```

```
net_protocol=:128M;
```

These are essential!



Track 3.2.0-dev branch

```
$> git clone https://github.com/jive-vlbi/jive5ab.git
```

```
$> cd jive5ab
```

```
# this is the important bit: checking out the branch!
```

```
$> git checkout 3.2.0-dev
```

```
# regular config + build steps
```

```
$> (mkdir build && cd build && cmake [opts] ..)
```

```
$> (cd build && make -j <ncpu>)
```

Track 3.2.0-dev branch

```
$> git clone https://github.com/jive-vlbi/jive5ab.git  
$> cd jive5ab
```

```
# this is the important bit: checking out the branch!  
$> git checkout 3.2.0-dev
```

```
# regular config + build steps  
$> (mkdir build && cd build && cmake [opts] ..)  
$> (cd build && make -j <ncpu>)
```

```
# if / when updates are published:
```

```
$> git pull
```

```
# trigger a full build
```

```
$> rm -rf build
```

```
$> (mkdir build && cd build &cet; you get the idea)
```

\$> etc/etd

<https://github.com/jive-vlbi/etransfer.git>

Two opportunities to "track"

release/2.0

dev/3.0

Two opportunities to "track"

`release/2.0`

publishes non-breaking features, bug fixes

`dev/3.0`

Two opportunities to "track"

release/2.0

dev/3.0

publishes possibly breaking features, new stuff

Two opportunities to "track"

release/2.0

regular build step

```
$> make -j <ncpu>
```



if / when updates are published:

```
$> git pull
```

trigger a full build

```
$> rm -rf ./{Linux-amd64|Darwin-arm64}..
```

dev/3.0

see jive5ab slides for how to start "tracking a branch"

release/2.0

git version info into binary

```
$> et[dc] --version
```

```
et[dc] : feature/auth@b6152d1* : nativebit : dev : opt
```

git branch @ commit hash *dirty-or-not

issue-32: ^C might deadlock etd

issue-30: Add Secure Reliable Transport as supported protocol

```
$> etd [...] --data srt://192.168.178.6#4005 [...]
```

issue-16: etd keeps write-lock forever if client disappears?

issue-21: etd timeout needed

root cause: long data transfer makes fw forget command channel, etd keeps file lock

solution: implement TCP_KEEPALIVE to, well, keep the command channel alive

automatically enabled on systems that support TCP_KEEPALIVE

```
$> etd [...] --tcp-keepcount N --tcp-keepinterval Seconds --tcp-keepidle Seconds ...
```

issue-31: add UUID to log lines where possible (tracking individual transfers)

issue-25: Live progress per file?

A progress meter was added to **etc** (default: **tty**)

```
$> etc [...] --progress {off,tty,always,auto} [...]
```

```
[#####] 79.0% 3.39 GiB / 4.29 GiB [ 46.63 MBps] 69.37 s
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

release/2.0 (security)

issue-29: Add ACLs [Access Control List, red.]
restrict daemon read/write areas
\$> etd [...] **--acl </path/to/file.yaml>** [...]

security-F09: set umask(022) in daemonization

security-F13: replaces access-then-open of ACL
now atomic test+open

security-F04: reject negative transfer size
would trigger +Inf data source b/c of underflow

security-F02: better RNG [Random Number Generator, red.]

security-F10: more entropy and less predictability in UUIDs

```
$> cat file.yaml
read:
  default:
    deny: "**"
  allow:
    - "/public-outgoing/**"

write:
  default:
    deny: "**"
  allow:
    - "/data/projects/**"
    - "/scratch/*/uploads"
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0 (security!)

everything in release/2.0

issue-35: Add TLS1.3

```
$> etd [...] --command tls://192.168.178.6#4005 [...]
```

```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - LibreSSL  
$> make TLS=1
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0 (security!)

everything in release/2.0

issue-35: Add TLS1.3

```
$> etd [...] --command tls://192.168.178.6#4005 [...]
```

```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - LibreSSL  
$> make TLS=1
```

Prevents "sniffing" the plaintext **etc/etd** protocol

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0 (security!)

everything in release/2.0

issue-35: Add TLS1.3

```
$> etd [...] --data tls://192.168.178.6#4005 [...]
```

```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - LibreSSL  
$> make TLS=1
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0

everything in release/2.0

issue-35: Add TLS1.3 and AUTHentication

```
$> etd [...] --command tls://192.168.178.6#4005 [...]
```

```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - Libressl  
$> make TLS=1
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0

everything in release/2.0

issue-35: Add TLS1.3 and AUTHentication

```
$> etd [...] --command tls://192.168.178.6#4005 [...]
```

```
# ssh-style known_hosts (client)
```

```
$> cat ~/.etransfer_known_hosts
```

```
io17.mpifr.bonn#4004 29:4c:dd:cf:49:fb:... (server fingerprint)
```

```
$> etc --identity <~/.private.key> tls://verkout@../ [--tls-verify {strict,tofu,ask}]
```

```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - LibreSSL  
$> make TLS=1
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

dev/3.0

everything in release/2.0

issue-35: Add TLS1.3 and AUTHentication

```
$> etd [...] --command tls://192.168.178.6#4005 [...]
```

```
# ssh-style known_hosts (client)
```

```
$> cat ~/.etransfer_known_hosts
```

```
io17.mpiifr.bonn#4004 29:4c:dd:cf:49:fb:... (server fingerprint)
```

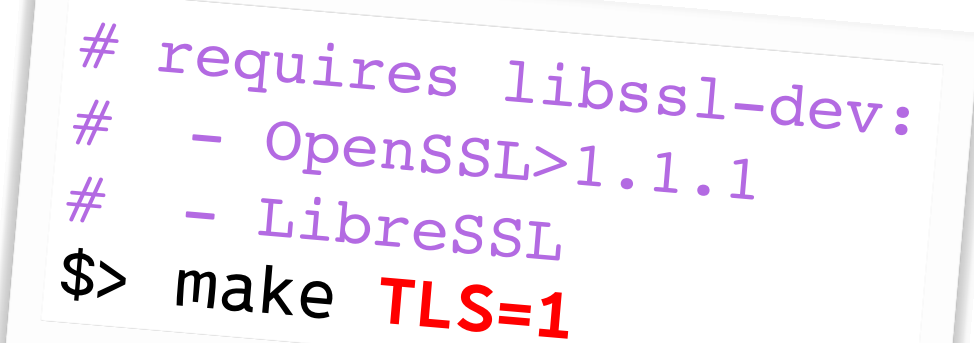
```
$> etc --identity <path/to/private.key> tls://verkout@io17.mpiifr.bonn#4004/ [...]
```

```
# ssh-style authorized_keys (daemon)
```

```
$> cat ~/.etd/verkout
```

```
ssh-ed25519 AAAAC3N<snip>wWsxTKw5 verkout@flexbuf4
```

```
$> etd [...] --cert <path/to/cert> --authorized-keys ~/.etd [--require-auth] [...]
```



```
# requires libssl-dev:  
# - OpenSSL>1.1.1  
# - LibresSL  
$> make TLS=1
```

AI-use disclosure: Claude Sonnet, Fable and GPT-Codex were used

Documentation

release/2.0

\$> et[dc] --help

web: <https://github.com/jive-vlbi/etransfer/tree/release/2.0>

dev/3.0

\$> et[dc] --help

web: <https://github.com/jive-vlbi/etransfer/tree/dev/3.0>