

# jive5ab 2.8.1

and support acts `m5copy` and `vbs_{fs,ls,rm}`

# New developments in jive5ab 2.8.1

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

debugging fun:

- `disk2file` (on FlexBuff) seemed to be trigger
- did not trigger @JIVE
- observed different malfunction after several hours of hammering (output to screen stopped but no SIGSEGV)

O/S and/or compiler and/or hardware sensitive?

sure it was *memory corruption* related to multithreading issue

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

Now what?

- assess MT\* safety of all systemcalls
  - a number of MT-unsafe versions were used (FIXED)
  - some MT-safe wrappers needed to be written
  - this is all invisible to the user/outside

Did not fix the problem at all ...

(\*) MT-safe = Multi-Thread safe = reentrant:

*cf. POSIX1.c: a "function whose effect, when called by two or more threads, is guaranteed to be as if the threads each executed the function one after another in an undefined order, even if the actual execution is interleaved"*

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

Now what?

- threads closing file descriptors in consistent manner (FIXED)
- serialize access to memory deallocator (FIXED)
- again totally invisible for user/outside

Did not fix the problem at all ...

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

Now what?

- actually think about what happens when triggered @station

Led to finding systematic problem:

inadvertently using memory after it was freed in `disk2file`

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

Now what?

- FIXED!

Adding insult to injury:

introduced by self in 2.8.0 in attempt to eliminate memory leaks

# Many fixes in 2.8.1

SIGSEGV reported by one (1) FlexBuff station: showstopper!

Found same internal approach in

- `disk2net (FlexBuff)`
- `file2disk`

All fixed; code in (even) better MT-safe shape now



# More fixes in 2.8.1

## Outward visible

- `record?` reply on FlexBuff made FS unhappy
  - introduced by self in (failed ...) attempt to consistentify the reply format across platforms (FIXED)
- suggestion by BeppeM `record?` (FlexBuff) returns
  - `record? 0 : [on|off] : ...`
  - **was** `active|inactive`

# More fixes in 2.8.1

## Outward visible

- `jive5ab` 2.8.0 lost ability to erase/initialize new disk packs
  - introduced by self, focussing on non-bank mode ...
- leftovers from change-of-internal-unit of frequency MHz→Hz
  - few instances overlooked in high-res time stamps of 2.8.0

# More fixes in 2.8.1

## Outward visible

- `net2out` cleanup code was not always called ....
  - could get stuck in mode w/o being able to repair
- (unlikely) failure in `record=on` on Mark5B could trigger ...
  - could get stuck in mode w/o being able to repair

# 2.8.1 and the past

Everyone upgraded?

- provide only 64-bit WHEEZY binary package?
  - with- (for Mark5\*), and
  - without SDK9.4 (FlexBuff)

# m5copy from 1.48→1.53

## Fixes

- handle changes in `record?` reply
- `file2net?` can return mixed state of running+not connected
  - causes `m5copy` to terminate w/ error
- wildcarded files are now transferred in alphabetical order
- sped up scanning wildcarded `vbs` recordings
- attempt to detect copying scan to bank label or VSN
  - missing 2<sup>nd</sup> slash in `mk5://.../A` or `mk5://.../MPI-001`
  - should read `mk5://.../A/` or `mk5://.../MPI-001/`

# vbs\_fs

## Fixes

- file descriptor leak indexing Mark6 recordings
- detect random or sequential access pattern
  - only read whole chunk if sequential access detected
- add `-I <pattern>` command line option
  - index only recordings matching `<pattern>`

# All tools: jive5ab, m5copy, vbs\_\*

Fix in all utilities:

- now able to deal with scan names with “.” or “+” in the name

# !



```
# !/bin/sh
```

```
ls *.txt | sed ...
```

```
# !/usr/bin/python
import re
rxVSN = re.compile(
```

```
# !.../m 5copy < ...>  
gg035a_ys_no0001  
n16c1_ib_no0120
```

...

# !.../m 5 copy < ...>

gg035a\_ys\_no0001

n16c1\_ib\_no0120

...

'template' command line



```
# !... /m 5 copy < ... >
```

```
gg035a_ys_no0001
```

```
n16c1_ib_no0120
```

```
...
```

```
# !.../m 5copy -udt-m 4500 {0} file://../
```

```
# !.../m 5copy -udt-m 4500 {0} file://../
```

```
# !.../m 5copy -udt-m 4500 {0} file://../
```

Python string formatting:  
“replace with field #*n*” – see URL below

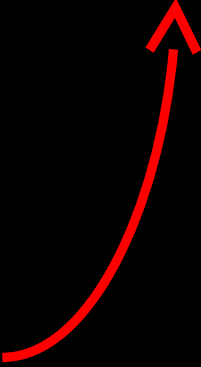


<https://docs.python.org/2/library/string.html#format-specification-mini-language>



```
# !.../m 5copy -udt-m 4500 {0} file://../
```

Not limited to one field per input line  
No spaces in fields though □



<https://docs.python.org/2/library/string.html#format-specification-mini-language>

```
# !.../m 5copy -udt-r {0} m k5 ://../{1} {2}
```

```
# send scans 1-10 to Bonn @ 128M bps  
128M 1-10 file://103.bonn.m pg.de/data/
```

```
# send n16c1 to JIVE @ 2G bps  
2G n16c1_hh_* vbs://flexbuf6.jive.nl/
```

```
# !.../m 5copy -udt-r {0} m k5 ://../{1} {2}
```

```
# send scans 1-10 to Bonn @ 128M bps  
128M 1-10 file://103.bonn.m pg.de/data/
```

```
# send n16c1 to JIVE @ 2G bps  
2G n16c1_hh_* vbs://flexbuf6.jive.nl/
```

```
# !.../m 5copy -udt-r {0} m k5 ://../{1} {2}
```

```
# send scans 1-10 to Bonn @ 128M bps  
128M 1-10 file://1003.bonn.m pg.de/data/
```

```
# send n16c1 to JIVE @ 2G bps  
2G n16c1_hh_* vbs://flexbuf6.jive.nl/
```