# MCR10202
# Fix issue where GCOS TSS `status` Command Prints 19YY
# Revision 1.0

Eric Swenson

October 29, 2026

## 1 Problem Description

Ticket #362 describes an issue with the GCOS TSS simulator, `gcos_tss` (`gtss`), where the `status` command prints the year as 1925 instead of 2025:

```
user status on may 16,1925 at 14:13:09 log-on at 14:13:03
```

### 1.1 The status command

`status` is not Multics code, but is part of the GCOS FAST library (`>unb>gtss_fast_library_`), in the program `.TSSTT`. The GCOS simulator loads from four program descriptors in `gtss_prgdes_alm_.incl.alm` (STAT, CONN, AFT and TEMP, i.e. the `status`, `conn`, `aft` and `temp` commands) run it. No source for these has been shipped by Honeywell in any Multics release.

The GCOS simulator, itself, does not supply the year. `.TSSTT` gets the date with the `DRL TIME` derail, which `gtss_drl_time_` correctly implements as GCOS specifies: `MM/DD/YY` in ASCII, stored here in words 1752–1753 ("MM/D", "D/YY"). The program then makes the century itself, with a constant:

```
000374 lda  777777,dl
000375 ldq  061071,du    ASCII "19" in the upper half
000376 ansa 1753         keep "YY"
000377 orsq 1753         1753 = "19YY"
```

To find this, the program was disassembled into GMAP source. The generated source for `.TSSTT` reassembles, with the GMAP in `gcos_system_software_`, to exactly the module in `gtss_fast_library_`.

### 1.2 Related problem in the simulator

Looking for other two-digit-year handling in the simulator found a problem in Multics code. The GCOS absentee login and logout banners, built by `gtss_abs_login_banner_` and

gtss_abs_logout_banner_, print the year as year - 1900:

```
HIS SERIES 6000 ON 10/10/126 AT 13.45 DEFERRED # ...
**ON AT 13.45 - OFF AT 13.58 ON 10/10/126
```

The other places that store a two-digit year (file creation dates in gtss_dq_, gtss_drl_t_cfio_, gtss_attributes_mgr_, gtss_filact_funct18_) use mod (year, 100), as GCOS formats require, and are correct.

# 2 Proposed Fix

## 2.1 .TSSTT

In the fixed GMAP source for tsstt.gmap, the instruction at 0374 becomes a transfer to a new routine appended to the end of the program; the three original words after it stay where they are but are no longer executed, so no address in the program moves:

```
* Ticket #362: the year used to be printed as 19YY whatever the
* year. Go to CENTRY, which picks the century from YY. The
* three words after the TRA are the original code, no longer
* executed (kept so that no address moves).
       TRA    CENTRY
       ...
L00400 TSX1   L01042
       ...
*
* Century for the "user status on" date (ticket #362). 1753 holds
* "D/YY" from DRL TIME. YY below 70 is 20YY, otherwise 19YY. Uses
* only A and Q, as the code it replaces did; returns to L00400.
CENTRY LDA    L01753         "D/YY"
       ANA    262143,DL      0,"YY"
       LDQ    25648,DU       "20",0
       CMPA   28208,DL       "70"
       TNC    2,IC           YY < "70": 20YY
       LDQ    25145,DU       "19",0
       STA    L01753
       ORSQ   L01753         "CCYY"
       TRA    L00400
```

Years 00–69 print as 2000–2069 and 70–99 as 1970–1999, which is correct for any date the Multics clock can give until 2070.

The program has no unused words to hold the new routine: its zero areas are buffers (DRL PASUST copies the UST into 01766–02265; 02451–02522 is the DRL KIN input buffer). So the module grows from 1192 to 1201 words, which changes its program descriptors.

## 2.2 Installing a changed GCOS module

gcos_tss_build_library (gtbl) is the tool for this. Given a library holding the new module, it appends the module to gtss_fast_library_ and updates every descriptor in

gtss_prgdes_alm_.incl.alm whose GECALL name is .TSSTT:

```
STAT: PRGDES 1,1,15875,1364,1192,172,172,
CONN: PRGDES 1,1,15875,1364,1192,172,172,
AFT:  PRGDES 1,1,15875,1364,1192,172,172,
TEMP: PRGDES 1,1,15875,1364,1192,172,172,
```

become 1,C,OFFSET,1373,1201,172,172, where C and OFFSET are the component and word where gtbl put the module. gtss_prgdes_ext_ (bound_gcos_tss_.2) includes that file, so it is reassembled and bound_gcos_tss_ rebound.

The module is made on Multics. GMAP, run under the batch simulator, assembles tsstt_fix.gmap into an absolute object deck, and the new command gcos_make_module (gcmm) makes from the deck a module in the form gcos_extract_module writes (preface, DCWs, data), which gtbl accepts. No existing Multics tool does this (SYSEDIT, GCOS's own tool for it, doesn't handle an absolute deck under the simulator). A new command, gcos_make_module, is proposed in MCR10201, (bound_gcos_tools_), which **must be installed first**. Its MCR shows that it rebuilds absolute modules of gcos_system_software_ word for word from their GMAP sources.

gtbl appends, so the old .TSSTT stays in gtss_fast_library_ as unused space. That also means the new library still works with the old bound_gcos_tss_ (its descriptors still point at the old module), so the library can be installed before the bound segment.

### 2.3 Banners

The two banners print the last two digits of the year, with a leading zero where needed, as GCOS printed them:

```
      call ioa_$rs ("HIS SERIES 6000 ON ^d/^d/^[0^]^d AT ^d.^d DEFERRED # ^a",
                ret_string, ret_len,
                month, dom, mod (year, 100) < 10, mod (year, 100),
```

and similarly for the logout banner.

### 2.4 Source

There is no source library for the GCOS software. This MCR proposes creating >ldd>unb>source>gtss_fast_library_.s.archive holding the fixed source as _tsstt.gmap (the module name in lower case, with . changed to _), so that later fixes to GCOS programs have a place to go.

## 3 Documentation

No documentation changes.

## 4 Test

Done on Multics (2026-10-28) in a private directory:

1. GMAP (gcos_system_software_) assembled tsstt.gmap (unchanged program) and tsstt_fix.gmap. The first deck loads the same words as the library module.

2. tsstt_test.ec made the module from tsstt_fix.objdk with a private gcos_make_module (gcmm tsstt_fix.objdk tsstt_fix.module -name .TSSTT), copied gtss_fast_library_ and gtss_prgdes_alm_.incl.alm, ran gtbl on them, reassembled gtss_prgdes_ext_, and bound a private bound_gcos_tss_. Only the four descriptors changed. A private gcos_tss also needs links to the installed GTSS.MCFC segments in its directory (link >unb>GTSS.MCFC.(FILES NAMES CALLERS_(1 2 3 4))), since gtss_mcfc_init_ looks for them there.

3. In a new process, the private gcos_tss:

```
*status
channel 0400 ts0
user status on oct 28,2026 at 8:10:24 log-on at 8:10:17
proc time used 0.07 sec., 0 file i/o char key i/o
list of open files: none
```

   aft and temp behave as before. conn stops with "Unimplemented drl T.CONN", as it does with the installed version: the simulator has no handler for that derail.

DRL TIME uses the real clock, so other years can't be tried on Multics. The new routine was run on the host in an instruction emulator for YY = 00, 25, 69, 70 and 99, giving 2000, 2025, 2069, 1970 and 1999.

Still to test, after installation: the steps above with the installed gcos_tss, and a GTSS absentee (deferred) job, whose login and logout banners should read MM/DD/26.

# 5 Installation

1. Install gcos_make_module (MCR10201).

2. In a work directory holding tsstt_fix.gmap, assemble it with GMAP using this job deck, tsstt_fix.jd, where DIR is the working directory's pathname (cards start in column 1 and end by column 72):

```
$     snumb  g1
$     ident  1234,ident
$     gmap   deck,ncomdk
$     limits 20,128k
$     data   g*
$     select DIR>tsstt_fix.gmap -ascii
$     prmfl  c*,r/w,l,DIR>tsstt_fix.objdk
$     endjob
```

   then make the module and add it to a copy of the library:

```
create tsstt_fix.objdk
gcos tsstt_fix.jd -hold -truncate
adjust_bit_count tsstt_fix.objdk
```

```
gcmm tsstt_fix.objdk tsstt_fix.module -name .TSSTT
copy >unb>gtss_fast_library_ gtss_fast_library_
copy >ldd>include>gtss_prgdes_alm_.incl.alm ==
gtbl tsstt_fix.module -modules .tsstt -output_library gtss_fast_library_
    -program_descriptors gtss_prgdes_alm_.incl.alm
```

gcmm should report 1201 words, entry 254.

3. Install `gtss_fast_library_` in `>unb`.
4. Install the rest with mbuild in the normal fashion. This includes installing `gtss_prgdes_alm_.incl.alm` into `>ldd>include`, `gtss_prgdes_ext_.alm` reassembled, the two banner sources recompiled, `bound_gcos_tss_` rebound.
5. Create `>ldd>unb>source>gtss_fast_library_.s.archive` from `tsstt_fix.gmap`, renamed `_tsstt.gmap`.

# 6 Files Changed

| | |
|---|---|
| `>unb` | `gtss_fast_library_` (fixed `.TSSTT` appended by `gtbl`) |
| `>ldd>include` | `gtss_prgdes_alm_.incl.alm` (STAT, CONN, AFT, TEMP descriptors, by `gtbl`) |
| `bound_gcos_tss_.1.s.archive` | `gtss_abs_login_banner_.pl1`, `gtss_abs_logout_banner_.pl1` |
| `bound_gcos_tss_.1.archive` | both objects (recompiled) |
| `bound_gcos_tss_.2.archive` | `gtss_prgdes_ext_` (reassembled, source unchanged) |
| `>unb` | `bound_gcos_tss_` (rebound) |
| `>ldd>unb>source` | `gtss_fast_library_.s.archive` (new): `_tsstt.gmap` |

# 7 Version History

| | | |
|---|---|---|
| 2026-10-29 | 1.0 | Initial version of the MCR. |