

OV 7.3 Seamless Guardian/OSS support

In OutsideView 7.3 or newer there is an enhanced sub mode “UNIX/OSS” that extends the basic 6530 emulation, via operation differences and extensions, to work optimally with UNIX environments that use TERMINFO and or TERMCAP files* to control the terminal behavior. OutsideView 7.3 or newer will now dynamically switch between modes depending on the application environment that is in use.

**NOTE: K-series OSS environments typically use only the TERMCAP file to control terminal behavior, while S-Series or Integrity NonStop systems may use either TERMCAP or TERMINFO files.*

The mode indicator in the lower right corner of your OutsideView screen will show either “UNIX/OSS” or “CONV” or “BLOCK”. CONV and BLOCK are standard 6530 terminal emulation modes. UNIX/OSS indicates the extended functional mode is operative.

NOTE: The UNIX/OSS mode changes the operation of your keyboard. Various local action keys like page up and down are now function keys that are sent to the host to be interpreted. So if you are at a UNIX/OSS command prompt and you want to look back in the buffer you will now need to use the scroll bar and your mouse. Also the mouse can no longer be used to reposition the cursor when in this mode. !

Requirements

To configure your NonStop environment to take advantage of this new capability, you need the following items:

1. OSS preinstalled and running on your NonStop host
2. Administrator logon for super.super
3. Example “tn653_termcap.txt” File from the OV installation CD.
4. Example “tn653_terminfo.txt” File from the OV installation CD.
5. Windows workstation running OutsideView 7.3 or above

If TERMINFO support is desired, then you will need three additional files from the Free Library for OSS (FLOSS):

6. “gzip.tar.Z”
7. “ncurses.tar.Z”
8. “vim.tar.Z”

The FLOSS library is available from the ITUG web site.
http://www.itug.org/member/ituglib/lib_instructions.cfm

Background

In the UNIX environment, support for terminal interaction has gone through three basic stages over the years.

1. Applications that are hard coded for a specific terminal type
2. Applications that are interfaced via a TERMCAP definition file
3. Applications that are interfaced via a compiled TERMINFO file

UNIX first saw a lot of use in educational environments where every effort was made to use heterogeneous computer terminals acquired by the institution. This operational requirement led to the development of a terminal capabilities file so that programs could dynamically interact with different types of terminals connected to the same UNIX system.

The OSS personality on the NonStop is initially at the level of Berkley UNIX; which can be extended by installing various packages (like ncurses) out of the ITUG FLOSS (free library for OSS) to add more modern capabilities and utilities. The FLOSS library can be accessed via the ITUG web site.

http://www.itug.org/member/ituglib/lib_instructions.cfm

On most modern UNIX distributions, when you invoke the VI text editor you are actually invoking an improved version of it named VIM; which is also available from the FLOSS library.

To support VIM you need to install “ncurses” and VIM as separate packages. Until the “ncurses” package is installed, TERMCAP is the controlling interface to the terminals under OSS. After installation of “ncurses” TERMINFO is the controlling interface for applications that are TERMINFO aware.

The normal ITUG FLOSS libraries are precompiled for the “S” series system and need to be recompiled for the “K” series (which is beyond the scope of this document). There is currently an alternate site that has a number of the FLOSS packages compiled for the new “H” Integrity/Itanium systems.

One of the drawbacks to the TERMCAP file was that it had to be dynamically scanned to locate the correct terminal definition and then parsed to locate the requested capabilities, this added significant overhead to start time for UNIX applications.

With newer versions of UNIX, this transitioned to a new functional definition named TERMINFO that changed this format to be more easily maintained; using expanded capabilities definitions and precompiled files that reduced the application startup overhead.

To prepare your NonStop OSS or UNIX system to use this new functionality takes some pre-configuration steps on the host to enable these enhanced interactions with OutsideView.

This can be done on an individual account basis or globally for everyone who connects to a NonStop and invokes the OSS personality. We will document the global procedure for TERMCAP first.

Global OSS TERMCAP setup

The procedure sets the OSS UNIX personality to default to a terminal type of “tn6530-8”, rather the initial value of “xterm”. Doing this will allow your users to seamlessly move back and fourth between the Guardian and OSS personalities on your NonStop system with out changing the terminal emulation type that they are using.

On most UNIX systems, the host telnet process dynamically sets the terminal type through negotiation when the telnet connection is initiated to the host. Under OSS, this is a static value that is controlled via the global profile file in the “/etc” directory. If you have individual users who still need to use a different terminal type, this can be set by editing a personal profile file in their home directory or by adding and using a simple script command to the profile file (as documented in the Section “Changing Terminal Type dynamically”.

To make the OV 7.3 terminal definition globally available on your system, perform the following steps from a command line window on a windows based workstation.

NOTE: In the example below, user commands/interactions are in bold text and assume we are working out of c:\tmp directory and have previously copied the tn653_termcap.txt file from the utilities sub-directory of the installation CD or CD image.

Transfer 7.3+ plus TERMCAP information to host

In the sequence of commands below, we open an FTP session to a host, log in as super.super, change over to the OSS file space and upload the new TERMCAP for OV 7.3 as a file fragment.

```
C:\tmp> ftp [host name, e.g. s72k]
Connected to s72k.cp.crystalpoint.com.
220 s72k.crystalpoint.com FTP SERVER T9552G07
(Version 3.w TANDEM 29JUL2005) ready.
User (s72k.cp.crystalpoint.com:(none)): super.super
331 Password required for SUPER.SUPER.
Password:
230 User SUPER.SUPER logged in. OSS API enabled
```

```

ftp> quote oss
257 OSS API enabled.
ftp> cd /etc
250 CWD command successful.
ftp> put tn653_termcap.txt
200 PORT command successful.
150 Opening data connection for tn653_termcap.txt
(172.16.0.169,1279d).
226 Transfer complete.
ftp: 1020 bytes sent in 0.00Seconds 1020000.00Kbytes/sec.

```

Set the default terminal type for all users

We now open a terminal session to the NonStop host and perform the following steps:

```

TACL 1> logon super.super
Password:
Last Logon: 20 DEC 2005, 09:38
Last Unsuccessful Attempt: * NONE *
TACL (T9205D46 - 27JUN2005), Operating System G06, Release G06.27.00
(C)1985 Tandem (C)2005 Hewlett-Packard Development Company, L.P.
CPU 0, process has no backup
December 20, 2005 9:42:27
(Invoking $SYSTEM.SYSTEM.TACLLOCL)
(Invoking $SYSTEM.STARTUP.TACLCSTM)

Loaded from $SYSTEM.STARTUP.TACLMACS:

V H F I P TYPE F I

Current volume is $SYSTEM.STARTUP
$SYSTEM STARTUP 1> osh
/home/super: cd /etc
/etc: ls -l
total 354
drwxrwxrwx 1 SUPER.SUPER SUPER 4096 Mar 3 2005 dns923
-rw-rw-rw- 1 SUPER.SUPER SUPER 1596 Jul 28 2003 inetd.conf
drwxrwxrwx 1 SUPER.SUPER SUPER 4096 Mar 29 2004 install_obselete
drwxrwxrwx 1 SUPER.SUPER SUPER 4096 Dec 6 18:59 install_obsolete
-rwxrwxrwx 1 SUPER.SUPER SUPER 488 Aug 31 1993 magic
-rwxrwxrwx 1 SUPER.SUPER SUPER 185 Dec 17 1993 printcap.sample
-rwxrwxrwx 1 SUPER.SUPER SUPER 268 Dec 6 18:59 profile
-rwxrwxrwx 1 SUPER.SUPER SUPER 268 Aug 9 1995 profile.sample
-rwxrwxrwx 1 SUPER.SUPER SUPER 166146 Jun 13 1994 termcap
-rw-rw-rw- 1 SUPER.SUPER SUPER 1000 Dec 21 13:01 tn653_termcap.txt
/etc: cat profile
# Remove /bin/unsupported from PATH if you DO NOT want to
# use the unsupported utilities on your system
# Note: /usr/ucb contains the OSS ftp client
#export PATH=/bin:/usr/ucb
export PATH=/bin:/bin/unsupported:/usr/bin:/usr/ucb
export TERM=xterm

```

```
export PS1='$PWD: '  
/etc: cp profile profile.sav  
/etc: cp termcap termcap.sav  
/etc: cat tn653_termcap.txt >>termcap  
/etc: mkdir /usr/local  
/etc: mkdir /usr/local/bin  
/etc:
```

In the above sequence of commands we open a terminal session, login as SUPER.SUPER and selected the OSS personality. We then change to the “/etc” directory, perform a directory listing and typed out the master profile file for our reference. Note that the default terminal type for OSS is “xterm”. We then made a copy of the original profile file for future use/reference by copying it to profile.sav.

We then make a copy of the master TERMCAP file to TERMCAP.SAV and appended our new definition for OV 7.3 6530 terminal emulation to the end of the file.

As a final step we created the primary program execution directory for the FLOSS libraries on the assumption that we will also load the ncurses later and other packages in the future.

Change the default terminal type

Now we need to change the default terminal type from “xterm” to “tn6530-8”, to do that we go back to our windows command window and perform the following additional steps:

```
ftp> get profile t1.txt  
200 PORT command successful.  
150 Opening data connection for profile  
(172.16.0.169,1282d) (392 bytes).  
226 Transfer complete  
ftp: 401 bytes received in 0.01Seconds 40.10Kbytes/sec.  
ftp> !notepad t1.txt
```

While in notepad we make the following changes to the profile file:

```
# Remove /bin/unsupported from PATH if you DO NOT want to  
# use the unsupported utilities on your system  
# Note: /usr/ucb contains the OSS ftp client  
#export PATH=/bin:/usr/ucb  
export PATH=/usr/local/bin:/bin:/bin/unsupported:/usr/bin:/usr/ucb  
export TERM=tn6530-8  
export PS1='$PWD: '
```

Once changes have been made, exit Notepad, saving the changes.

```
ftp> put t1.txt profile
200 PORT command successful.
150 Opening data connection for profile
(172.16.0.169,1283d).
226 Transfer complete.
ftp: 401 bytes sent in 0.00Seconds 401000.00Kbytes/sec.
ftp> bye
221 Goodbye.
```

In the above series of commands we first bring the global profile file that applies to all users down to the local pc, we then use the notepad program to edit the file before transferring it back to the host.

NOTE: In the path value we specify above, the commands in the /usr/local/bin location would take precedence over any like-named command in the standard OSS distribution/installation. For instance, if the “grep” were downloaded from the FLOSS library into the /usr/local/bin, that version of grepo would superceded the standard version..

A more conservative value for the path definition would be:

export PATH=/bin:/bin/unsupported:/usr/bin:/usr/ucb:/usr/local/bin:

At this point, to make the changes active for your terminal session, exit and re-invoke the OSS personality.

```
/home/super: exit
$SYSTEM STARTUP 3> osh
/home/super:
```

NOTE: To confirm changes are in place, you may use the command set to see active settings:

```
/home/super: set
ERRNO=4010
FCEDIT=/bin/ed
HOME=/home/super
IFS='
'
LINENO=1
LOGNAME=SUPER.SUPER
MAILCHECK=600
OPTARG
OPTIND=0
```

```
PATH=/usr/local/bin:/bin:/bin/unsupported:/usr/bin:/usr/ucb
PPID=1
PS1='$PWD: '
PS2='> '
PS3='#? '
PS4='+ '
PWD=/home/super
RANDOM=1163074432
SECONDS=16
SHELL=/bin/sh
TERM=tn6530-8
TMOU=0
_=$PS1='$PWD: '
```

Personal OSS TERMCAP setup

This procedure is very similar to the global setup procedure except that we are working with the user personal profile file and changes and file uploads are made to the users home directory.

```
C:\tmp>ftp ktenk
Connected to ktenk.cp.crystalpoint.com.
220 ktenk.cp.crystalpoint.com FTP SERVER T9552D40 (Version
3.e TANDEM 26FEB98) ready.
User (ktenk.cp.crystalpoint.com:(none)): tst.joe
331 Password required for TST.JOE.
Password:
230 User TST.JOE logged in. GUARDIAN API enabled
ftp> quote oss
257 OSS API enabled.
ftp> pwd
257 "/home/joe" is current directory.
ftp> put tn653_termcap.txt tn653termcap
200 PORT command successful.
150 Opening data connection for tn653termcap
(172.16.0.169,2522d).
226 Transfer complete.
ftp: 1020 bytes sent in 0.00Seconds 1020000.00Kbytes/sec.
ftp> get .profile t1.txt
200 PORT command successful.
150 Opening data connection for .profile
(172.16.0.169,2524d) (580 bytes).
226 Transfer complete
ftp: 598 bytes received in 0.01Seconds 59.80Kbytes/sec.
```

```
ftp> !notepad t1.txt
```

While in notepad, we make sure that following two lines are present in the file to set the terminal type and to specify where the controlling TERMCAP file is located:

```
export TERM=tn6530-8  
export TERMCAP=/home/joe/tn653termcap
```

Once changes have been made, exit Notepad, saving the changes.

```
ftp> put t1.txt .profile  
200 PORT command successful.  
150 Opening data connection for .profile  
(172.16.0.169,2525d).  
226 Transfer complete.  
ftp: 598 bytes sent in 0.00Seconds 598000.00Kbytes/sec.  
ftp> bye  
221 Goodbye.
```

In the above sequence of commands we opened a FTP session to the host and then changed over to the OSS file space and uploaded the new TERMCAP definition for OV 7.3 as the file “tn653termcap”.

We then edited the user’s personal profile file “.profile” by bringing it down to the local PC and using the notepad program. While in notepad, we set the terminal type and specified the location of the controlling TERMCAP file

Depending on how the user’s OSS login account was created, they may not have a personal profile file. One option is to simply copy the one from the “/etc” directory and modify that. Another option is to create a small one with just the above two configuration lines in it.

TERMINFO Support

To advance to the next level of terminal support, which is TERMINFO, it is necessary to download three packages from the FLOSS library on the ITUG web site. If your system already features TERMINFO support, skip to the topic **COMPILE TERMINFO File** below.

http://www.itug.org/member/ituglib/lib_instructions.cfm

They are:

1. “ncurses.tar.z” which provides TERMINFO support.
2. “vim.tar.Z” which is the improved VI editor that supports function keys.

3. "gzip.tar.Z" which is the gzip utility that does an uncompress of files that end in a type of ".z" (lower case "Z"). Note this package is not required, though since you are performing this process it is a good idea to do this right now.

For the following example we make the assumption that you pre-created and downloaded these files to a directory of "C:\TMP". The next step is to open a DOS style command window on your PC and use the intrinsic FTP to perform the following steps:

Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

```
C:\Documents and Settings\support>cd \tmp
```

```
C:\tmp>dir
```

Volume in drive C has no label.

Volume Serial Number is 14A5-86F5

Directory of C:\tmp

```
12/19/2005  03:16 PM    <DIR>          .
12/19/2005  03:16 PM    <DIR>          ..
12/19/2005  03:16 PM                554,745  gzip.tar.z
12/19/2005  02:48 PM                7,810,397 ncurses.tar.z
12/19/2005  02:50 PM               12,322,753 vim.tar.z
12/22/2005  11:24 AM                  1,020  tn653_termcap.txt
12/22/2005  11:23 AM                  1,258  tn653_terminfo.txt
              5 File(s)                20,690,173 bytes
              2 Dir(s)   222,971,322,368 bytes free
```

```
C:\tmp>rename *.z *.Z
```

```
C:\tmp>ftp s72k
```

Connected to s72k.cp.crystalpoint.com.

220 s72k.crystalpoint.com FTP SERVER T9552G07

(Version 3.w TANDEM 29JUL2005) ready.

User (s72k.cp.crystalpoint.com:(none)): **super.super**

331 Password required for SUPER.SUPER.

Password:

230 User SUPER.SUPER logged in. OSS API enabled

```
ftp> quote oss
```

257 OSS API enabled.

```
ftp> pwd
```

257 "/home/super" is current directory.

```
ftp> mkdir /home/super/floss
```

257 MKD command successful.

```
ftp> cd /home/super/floss
```

```
250 CWD command successful.
ftp> put tn653_terminfo.txt
200 PORT command successful.
150 Opening data connection for tn653_terminfo.txt (172.16.0.165,3383d).
226 Transfer complete.
ftp: 1258 bytes sent in 0.00Seconds 1258000.00Kbytes/sec.
ftp> binary
200 Type set to I.
ftp> put gzip.tar.Z
200 PORT command successful.
150 Opening data connection for gzip.tar.Z (172.16.0.121,5006d).
226 Transfer complete.
ftp: 554745 bytes sent in 2.27Seconds 244.81Kbytes/sec.
ftp> put ncurses.tar.Z
200 PORT command successful.
150 Opening data connection for ncurses.tar.Z (172.16.0.121,5007d).
226 Transfer complete.
ftp: 7810397 bytes sent in 28.64Seconds 272.67Kbytes/sec.
ftp> put vim.tar.Z
200 PORT command successful.
150 Opening data connection for vim.tar.Z (172.16.0.121,5008d).
226 Transfer complete.
ftp: 12322753 bytes sent in 43.74Seconds 281.73Kbytes/sec.
ftp> bye
221 Goodbye.
```

```
C:\tmp>
```

In the above example the following steps occurred:

1. We changed to the directory that we download the files from the ITUG user library from.
2. We renamed all files that ended in lower case “Z” to an uppercase “Z”. Note the three files that we downloaded the FLOSS library using Internet Explorer were more and likely saved with the lower case dot “Z” extension. If you download additional packages from the FLOSS library be sure to remember if they end in lower or upper case “Z” and rename them before transferring them to your host.
3. We then invoked the intrinsic command line FTP for windows and logged into the host as SUPER.SUPER
4. We issued the “quote oss” command to make sure that we are in the OSS file space even though the message had informed us that the OSS API was already enabled.
5. We created our upload directory at “/home/super/floss” and changed to that directory.

6. We upload our terminfo definition file for OutsideView, which we will compile later in this document.
7. We invoked binary file transfer mode so that future transfers would do a binary copy from our pc to the host.
8. We then uploaded the three archive files. Note it does not matter what case you use to specify the filename in the put command. The windows command line ftp uses the case of the filename as it is stored in the directory. That is why we renamed them before transferring the files.

We then use OutsideView to connect to the NonStop system (as super.super) and invoke the open system shell and perform the following commands:

```
$SYSTEM STARTUP 1> osh
/home/super:cd /home/super/floss
/home/super/floss: ls -l
total 40407
-rw-rw-rw- 1 SUPER.SUPER  SUPER  554745 Dec 19 15:15 gzip.tar.Z
-rw-rw-rw- 1 SUPER.SUPER  SUPER  7810397 Dec 19 15:09 ncurses.tar.Z
-rw-rw-rw- 1 SUPER.SUPER  SUPER   1236 Dec 21 13:46 tn653_terminfo.txt
-rw-rw-rw- 1 SUPER.SUPER  SUPER 12322753 Dec 19 15:10 vim.tar.Z
/home/super/floss: uncompress gzip.tar
/home/super/floss: uncompress ncurses.tar
/home/super/floss: uncompress vim.tar
/home/super/floss: cd /
```

NOTE: It is correct behavior that some error messages display while the commands below take place. Also, be patient; these actions take a while to perform.

```
/: tar xvf /home/super/floss/gzip.tar
/: tar xvf /home/super/floss/ncurses.tar
/: tar xvf /home/super/floss/vim.tar
```

In the preceding example we performed these steps:

1. Switch from the TAACL to the OSS space
2. Changed to the upload directory that was made with the remote FTP client.
3. Uncompressed the TAR file that contains the GZIP file compress utility.
4. Uncompressed the TAR file that contains the NCURSES library.
5. Uncompressed the TAR file that contains the VIM editor utility.
6. Invoked TAR to unpack the GZIP package
7. Invoked TAR to unpack the NCURSES package
8. Invoked TAR to unpack the VIM package

COMPILE TERMINFO File

The final step is use the terminal information compiler (TIC) to compile the TERMINFO definition file for OV.

```
/: cd /home/super/floss  
/home/super/floss: tic tn653_terminfo.txt  
/home/super/floss:
```

After this is done, the NonStop OSS has been configured to support the OV 7.3 UNIX/OSS sub mode for both TERMCAP and TERMINFO aware UNIX programs.

If it is desired to replace the usage of VI with it's more powerful and modern VIM, you can add the following line to the appropriate profile file.

```
alias vi=vim
```

Changing Terminal Type Dynamically

With the addition of a simple script to the appropriate profile file you can add a command that can be executed by the user to change the terminal type on the fly.

The simple script for this adds a command (capital) "T".

Typing "T" alone tells you the term type, while typing "T newterm" ("T ansi", for example) will show you what your TERM was and now is.

Cautions:

- 1) Functions do not carry over to sub-shells, so if you shell out of an application this function will not be available.
- 2) Functions run before any commands of the same name in the PATH so if there were a /usr/local/bin/T command, the function T would run first (unless you invoked it as "/usr/local/bin/T").

After making this change our /etc/profile file would have the following contents:

```
# Remove /bin/unsupported from PATH if you DO NOT want to # use the unsupported  
utilities on your system  
# Note: /usr/ucb contains the OSS ftp client  
#export PATH=/bin:/usr/ucb  
export PATH=/usr/local/bin:/bin:/bin/unsupported:/usr/bin:/usr/ucb  
export TERM=tn6530-8
```

```
export PS1='$PWD: '
```

```
#
```

Shell script to allow user to dynamically change their terminal type

```
T() {  
if [ -z "${1}" ]  
then  
    echo "Your TERM type is set to $TERM"  
else  
    echo "Your TERM was $TERM \c"  
    TERM=${1}  
    export TERM  
    echo "and is now $TERM"  
fi ;  
}
```