

Assigning terminals in BBX config.bbx

By Dirk Hart

Because pseudo tty numbers are assigned on a first-come first-served basis it can be difficult to assign the users the same 'terminal number' each time they log in . Here is a nifty way to tame those pesky tty numbers and get them to settle down.
cartoon

When each user logs in they customarily run a script named .profile in their home directory. The .profile does things like set the INTR key sequence, the TERMCAP environment variable and exec bbx. The bit of script shown here will also set an environment variable called BBTERM which bbx knows about and looks for.

In this example, set passes back the terminal number if it finds LOGNAME in the set.bbterm table otherwise XxXx is passed and BBTERM is set accordingly.

```
set `grep $LOGNAME /u/set.bbterm` XxXx
if [ "$1" != "XxXx" ]
then
    BBTERM=$1
else
    BBTERM=Tx
fi
export BBTERM
```

Fig.1: script to be added to .profile to set BBTERM

For example if someone logs in as pulsar BBTERM is set to T300.

```
T300 pulsar
T301 quasar
T302 venus
T303 mars
```

Fig.2: Example of set.bbterm

Be sure you have T300 and all the other terminals number mentioned in config.bbx.

```
alias T300 /dev/tty ansicoe
alias T301 /dev/tty ansicoe
alias T302 /dev/tty ansicoe
alias T303 /dev/tty ansicoe
alias T304 /dev/tty ansicoe
```

Fig.3: Part of config.bbx

If the terminal type is term instead of ansicoe, bbx takes the terminal type from the TERM environment variable.

There are similar ways of dealing with FacetTerm and multiple FacetWin logins as well.

The examples I gave is from a live BBX PRO5 system, I am quite certain it works on BBX Pro4, pretty sure it works on Pro3 and I'm quite surprised to hear someone is still using BBX Pro2 ... your customer surely knows how to squeeze a dime.

In BBx every tty line must be described in the config.bbx file. This lets every tty be assigned a unique terminal number such as T0, T1,T2...T300, or in your case TA..TZ, Ta..Tz.

There was a time when *all* BB applications came off of Basic Four systems which used terminal numbers such as T0..T9,TA..TZ and there were only serial tty's in those days. Life was simple and good and everyone wrote applications that used the terminal number (T300, eg) to control certain things like preventing more than 1 person from editing the same order, to keep the shipping dept from inspected the General Ledger, or to prevent the counter terminals from displaying cost data that customers might see and such like that. Many many applications were written that only ever expected the terminal to be 2 characters long (T2, e.g.) and the world was thrown into a spin when 3 character terminal numers (T10..T99) were introduced.

When BBx came along not only could you have 512 terminals, but they weren't restricted to 3 characters - you could have T1000 or even Ttony and the ilk - but your application still had to be able to handle such terminal numbers - if it used terminal numbers to control things at all, and application packages pretty much all do.

And then came PCs and LANs and pseudo ttys. Since the tty number assigned to an incoming connection over the network (ttyp0, ttyp1, ttyp2, etc.) is for all intents and purposes random we need to depend on other things to set the terminal number. One of those things is LOGNAME as I outlined previously. With FacetWin it might be the NETBIOS name of the PC.

Now back to config.bbx. Every incoming terminal based connection must be described. The format of the terminal definition entries in config.bbx are:

```
alias <terminal number> <tty number> <terminal type> <options>
```

For example,

```
alias T300 /dev/ttyi00 wy60
```

assigns T300 to the serial terminal connecting to the system on /dev/ttyi01 and that BBx will communicate with that terminal as though it were a Wyse 60 (wy60) terminal. Of course if that terminal is a VT100 terminal, odd things will happen to character positioning and other aspects of the display. If we use 'term' in place of wy60 BBx will take the terminal type from the TERM environment variable. BBx is TERMCAP based and comes with its own TERMCAP file, usually located in the same directory as config.bbx. This termcap has more function keys described for many terminals, but is otherwise a regular termcap.

If you have PCs connecting to BBx applications you can assign a value to an environment variable called BBTERM. BBx will check the environment upon startup and will assign the value of BBTERM to the current pseudo tty. For example, if you set BBTERM=T300 (and export it!) BBx will start its new session with a terminal number of T300. The skill is entirely in how you assign a value to BBTERM. In bob's .profile you can simply say BBTERM=T300;export BBTERM and that user will always get T300. In mary's .profile you can set BBTERM=T301 and mary always gets assigned T301. You can also use the same home directory for all users and make use of the little scriptlet previously posted to assign terminal numbers (preferred method for large installs). Note that if you are using BBTERM to assign the terminal number you don't need to fully describe the tty number, which is why you sometimes see aliases such as:

```
alias T300 /dev/tty term
```

To check your terminal number from a bbx console prompt ('>' aka 'READY arrow' or READY prompt) type in PRINT FID(0). If it says IO you have not properly assigned a terminal number and your display will be a mess.

Finally you should be aware that bbx devices usually start with a 'magic' character. T for terminal devices, L or P for printers, etc. All devices starting with T, such as T300, send control codes appropriate to terminals (remember termcap?) and similarly all L and P devices send control codes appropriate to printers.