

Useful info about the Q-Link Stratus:

Dial up the local Telenet node, enter 3 carriage returns and then (at the '@' prompt) enter '703391' (that's an "L", not a "1").

Our login is 'chip'. Password is 'lunch'.

To login type

```
login chip -module ml
```

It will prompt for the password.

Directory structure:

```
%cvideo#d010>lucas
```

Our home directory. The only interesting thing here is the 'abbreviations' file. Update this at your peril.

```
%cvideo#d010>lucas>microcosm
```

The master directory for OUR source code on the Stratus. The alias 'work' will cd you to this directory from anywhere in the system.

EVERYTHING that is ours should be compiled from this directory, even though the source is in one of its subdirectories (otherwise the include files won't work -- bunch of include files that are used everywhere are to be found here).

```
%cvideo#d010>lucas>microcosm>Actions
```

Sources for the generic action code. A bunch of files of the form 'actions\_something.pll'. These are all included in a single master file named 'actions.pll'.

To recompile, use the command 'pll Actions>actions' from the master directory.

```
%cvideo#d010>lucas>microcosm>Classes
```

Sources for the class behavior code. There is one file for each class, named 'class\_something.pll'. To recompile, use the command 'pll Classes>class\_something' \*or\* 'plc something' from the master directory.

```
%cvideo#d010>lucas>microcosm>Structs
```

Sources for struct definition include files used everywhere. There is one such file for each class names 'struct\_something.incl.pll' plus a bunch of additional ones that are used in various places. You don't need to compile these, they are source included.

```
%cvideo#d010>lucas>microcosm>Misc
```

Sources for miscellaneous routines used in our code. There are three files, 'helpers.pll', 'bits.pll' and 'messages.pll'.

'helpers.pll' is the bulk of the miscellaneous routines. Basically, anything of ours that is not elsewhere is here (of course).

'bits.pll' is all the bit-twiddling routines that PL/1 doesn't let you do nicely.

'messages.pll' is all the message I/O routines for the behavior code.

Compile this stuff with 'pll Misc>whatever' from the master directory.

```
%cvideo#d010>lucas>microcosm>Linkable
```

The linkable object modules ('.obj' files) for ALL of our stuff. Whenever you recompile some things (from the master directory always), you should enter the command 'mv \*.obj Linkable', otherwise changes will not get picked up by a rebind.

```
%cvideo#d010>lucas>toolbox
```

Contains a bunch of utilities we've written. The alias 'cdtool' takes you here from anywhere.

In particular, this directory contains OUR version of Twiddle, so always 'cdtool' before running twiddle, otherwise it may not know about some of the newer classes.

```
%cvideo#d010>quantum>stratus>source>microcosm
```

Source and object files for the regionproc and the like. The alias 'cdbind' takes you to this directory. There is an abbreviation for this directory name, 'bindir', that can be used in command arguments at any time.

You can recompile the regionproc from this directory with the command 'pll regionproc'. You don't need to do anything with the '.obj' file after recompiling regionproc.

This directory also contains the stuff we need to update twiddle when we add a new class. There is a file 'init\_twiddle.pll' here. Here you will also find 'init\_twiddle.doc' and 'twiddle.doc' which tell you how to upgrade twiddle and how to use twiddle, respectively. If you update 'init\_twiddle.pll' according to these instructions, you then need to enter the following commands (in order, from this directory):

```
pll init_twiddle
bind twiddle init_twiddle
cdtool
cp bindir>twiddle.pm . -delete
```

```
%cvideo#d010>quantum>run_lucas
```

The "run directory". The alias 'cdrun' will take you here. There is also an abbreviation for this directory name, 'rundir', that can be used in command arguments.

This is where you do a reset from. The command 'reset' (which will only work from this directory) will reset Habitat.

The log file also lives here. It is named 'log.log'.

Also in this directory are the commands 'stop\_monitor' and 'start\_monitor' which, respectively, bring down and put back up the Q-Link test system. Be careful that ALL processes (except your login process, of course) are dead before doing a 'start\_monitor'. They don't stop instantaneously when you do a 'stop\_monitor'.

You can leave "mail" for Janet by putting messages in files here with names like 'note\_to\_janet\_47' (do an 'ls' first to pick the next number in the sequence). Janet also leaves mail for us here, in files with names like 'note\_to\_chip\_23'. The alias 'mail' will check for these files from anywhere. You just have to remember what the most recent one was, as it doesn't just show you the new messages...

Other useful command aliases we have created:

'fixup' will rebind the regionproc, install it AND reset. Can be

issued anywhere.

'redo' will rebind and reinstall but WON'T reset. This is useful for making a new version while you are still testing an old one.

'rebind' will rebind. Only works from 'bindir'. Not useful unless you know what you're doing.

'tl' will print out the last ten lines of the log file (from anywhere).

'up' will tell you what processes are running. This is how you tell if the regionproc has crashed. The three processes to look for are 'MCdata', 'microcosm' and 'RegionP0' (sometimes also 'Region.0').

'nopause' always issue this command after having done a 'start\_monitor'. It will fix up the terminal I/O that 'start\_monitor' always messes up (by default, the Stratus always pipes ALL terminal output through something like the unix 'more' command, which is usually very irritating).

'grep' approximates the unix grep command.

'alias' greps through the abbreviations file, in case you forgot a command or a directory name.

'edit filename' invokes the awful Stratus line editor. There is a copy of the Stratus commands manual (which explains the line editor) on my desk.