Colorizing HC Debug

This stack contains a prototype debugger that attempts to show you the colorizeHC commands, so that you can see what commands are actually being issued. It is designed for if you are surprised at what happens, and are having difficulty working out which commands are being shown. It is normally initialised by giving the "dbColorizeHC" as an option to the XCMD stack's initialisation commands, but after you have entered a stack, you can type:

  start using stack "Colorizing HC Debug"

which will work until you stop or suspend the current stack. Essentially it intercepts the colorizeHC command, prints out the parameters, and performs the command as normal. By default, the output is placed in the message watcher window. However, if you have installed Textoids – see README – then you can optionally tell the debug stack to create its own window. This option is controlled by a button on the first card of the Debug stack itself. By default the window appears in the top left of the screen – if you want it somewhere else you can give the topLeft or rectangle on a field on that card, or better make the window visible, move it to the required position, and command click on the position field. Note that command "cHCw" will display the window if it is not visible.

It should be noted that some handlers have been copied from the XCMD stack itself – the debug stack cannot intercept instances of colorizeHC issued directly from the XCMD stack, and the copies here help get around this. Card 2 in this stack contains a list of items that are copied, and allows the stack script to be updated.

Bugs

The stack is but a prototype, and there are a few unresolved problems:
- The stack is slow. This is partly due to Textoids – they have functionality that really surpasses what is required. Having a scrolling window, in the manner of a terminal emulation, is not their strength – it's possible, but at obvious cost.
- There is an obscure bug which prevents the use of "pass" in the interception script – just after initialisation it misses the XCMD, and if ColorizeHC is used from the debug stack itself, the debug ColorizeHC handler is passed through twice! To get around both of these, I used "send". This not only slows down activity, but also stops errors from appearing.

Any suggestions would be welcome. The reason for an extra stack was to reduce the load when not debugging, and also to allow alternatives. Given the size of the XCMD script anyway, I fell it better to pursue this line.

John Forrest

September 1992.