Colorizing HC XCMD 1.1.5

Yet another version of this stack. The XCMD itself remains unchanged, but there are some new facilities and a few bug fixes. Most of the new facilities concern window support – for example, if used properly the stack will automatically move windows placed over a card when the card itself is moved. There is also some mechanisms to support Nigel Perry's Colour Text stack, although you will still need to get the XCMD stack from elsewhere.

ColorizeHC itself requires at least Hypercard 2.0v2 to work. This stack was made under Hypercard 2.1 (B-2.1 to be exact), but should work with 2.0v2 – it uses no extra facilities. The latest version has been tested on both a Classic and a Duo230 running System 7.1 and Hypercard 2.1.

Quick Start

I recommend that you install stuff regarding the instructions below. However, if you want to see if it is worth it before continuing…

You can just unpack the files into a single directory, and have a play in there. Once you have done that, you can browse through *Colorizing Hypercard 1.0'*, which shows what colorizeHC can do. The first few cards of *Colorizing HC Info*, describe the basic principles of changing your stacks to use colour – placing extra commands in open or close handlers to get extra messages sent out – and there is a summary of the colouring commands for you to use in HCcolorize routines, which actually place colour on the screen. The rest is up to you. The remainder of the Info stack discusses how to use the XCMD stack as a general aid in stacks where the appearance of cards is commonly altered on entry; and also describes the more advanced features of the stack.

Summary of changes for v1.1.5

The main changes are that the initialisation command names have changed: cHCsetup, cHCleave and cHCresume have become cHCopen, cHCsuspend and cHCresume. These should be simpler to remember. The old ones still work, as they are supported as synonyms. However, you are encouraged to use the new ones.

Apart from this most of the changes have been to do with windows. Using various commands, the XCMD stack can be told where windows are supposed to be in relation to the card window – so that it can move the cards using command cHCreWindow, which should be called from handler moveWindow. However, this is optional – as it showing the window automatically – which should cover cases where windows can be kept invisible. There is also support for Colour Text in the same way as pictoids – except that the handler in the XCMD stack replaces their equivalents in the Colour Text Stack, rather than calling them in turn.

Message HCunInit has been introduced for completeness. It is called by cHCsuspend and cHCclose if requested and after the colour has been closed down.

Some new read-only global variables to ease script writing. gHCwhite and gHCblack correspond to white and black respectively. gHCmaxColor is $2^{16}-1$, and is the maximum value for any of the RBG elements of a colour. gHCmaxDepth is the result of maxDepth(). It is suggested you use this to see if you can use colour with windows, etc, rather than relying on gOKtoColorize – this allows windows to be properly supported with the colour backdrops off.

There are some major changes in the window handling structure, that allow window id's to be used instead of names. Also, the XCMD stack no longer does an implicit break lock when opening a stack, but relies on its ability to move windows to where they should go. This results in much smoother transfer, arguably removing the need for the windows option – although this change has yet to be made. At the same time, the full name of a displayed file can no longer be used as the argument to cHCkillWindow, which is a minor incompatability with previous versions.

There is a bug fix in cHCunLock, where messages were never re-enabled.

The XCMD stack will now work in a stand-alone mode – it is sensitive to the Info and Aids stacks not being present.

More Info

Version 1.0 was a single stack, that contained the scripts and the info pages on how to use it. In more recent versions, this has been split into 4 – the basic XCMD and support script, the Info pages, the supplied bitmaps, and some authoring aids. Furthermore, the original Colorizing Hypercard stack has been rewritten to use the facilities of the XCMD stack, and this is now included as well. A couple of extra files are also distributed.

The files, and installation instructions, are as follows:

Colorizing HC XCMD
The XCMD file itself. If you have a folder you normally use for support stacks, place it there. Otherwise, place it in the same folder as the Hypercard application. Its presence is required by almost every stack that uses its facilities.

Colorizing HC Info
This stack describes how to use the XCMD stack – best to place it in the same folder, so you can use it as an online reference. Apart from "help" it is not required, but you will find its presence indispensable to start with.

Colorizing HC Extras
This stack contains coloured bitmaps that originally came with the Colorizing Hypercard stack. It should be placed in the same folder as the XCMD stack, or alternatively as Colorizing Hypercard 1.0'. The latter requires its presence to work.

Some of Rinald's
This stack contains some XCMDs by F. Rinaldi. See copyright info below.

Colorizing HC Aids
This stack contains some authoring aids that were previously in the XCMD stack, but have been moved out for space reasons - the Info stack says which. You don't have to explicitly use it, but it should be in the same folder as the XCMD stack if you intend to use the functions. It is

only required in stack design, and not when you merely use a stack.

Colorizing HC Debug

This is prototype debugging aid. If you wish to use it, place it in the same directory as the XCMD stack. See file “Debug README” for more information, and the note below about F.Rinaldi copyright.

Colorizing Hypercard 1.0'

This is a reworking of the original stack, that uses the XCMD stack. It is the reference for the ColorizeHC XCMD, and a good example of what is possible. Note that the stack is quite big, and not all of it has yet been changed to meet the recommendations in Info.

Colour Text Stack (cHC)

This is a version of Nigel Perry’s Colour Text Stack, rewritten to use the XCMD stack, and minus the Colour Text XCMDs – it expect there to be a stack “Colour Text Stack” containing these. Unfornately, **this cannot be distributed here**. See copyright information below.

Add REAL color to HC!

This is the original README for Colorizing Hypercard, and is included for reference – although what it says is phrased better in the stack itself.

Colorizing HC Template

This is a template stack, with the stack script already setup to allow for simple colouring.

Colorize and Pictoid Template

The XCMD stack contains some support for Pictoids, although the Pictoid XCMD is not distributed here. This template stack contains a stack script to setup both the XCMD stack and Pictoids.

Colour Text Template

Similar but applying to the Colour Text Package. Colour is disabled by default – reflecting the use of the XCMD stack to merely support the Colour Text Package with its window utilities.

F.Rinaldi Copyright A few of the stacks need XCMDs from “Rinaldi’s Externals” for full functionality. Since the conditions of use of these are different to this distribution – in terms of commercial exploitation, these are not inserted in the associated stack. Some are distributed in stack “Some of Reinaldi’s”, so that this distinction is clear. Essentially if you wish to use these resources for commercial purposes, you must get permission to do so, and **must not** distribute stack “Some of Rinaldi’s” with your software. In any case, the functions are not particularly central to this material – and you can make do without them. The following are affected:

- Stack “Colorizing HC Extras” requires FullResList in order to remake itself. This is distributed in “Some of Rinaldi
- Stack “Colorizing HC Debug” requires Textoid in order to display its own window. Since it is associated with development, and not final use, you should ask permission to use it for commercial work, full stop. You must obtain it yourself from an archive – copy that XCMD and the “rinaldoid” WDEF and CDEF into the Debug stack. Without this, output can only be sent to the message watcher.

Nigel Perry Copyright I had originally planed to distribute a stack called “Colour Text Stack (cHC)”, which was the Colour Text Stack without its XFCNs and rewritten to use the XCMD stack. However, Nigel Perry, who has the copyright, would not let me unless I distributed all the stuff here under the same enforced freeware license arrangements he uses. This can be summarised by:

> *(Without prejudice to the following, the conditions of use might be paraphrased as “Provided you don’t make money out of this software or charge me for your software [unless written as part of your job and sold as a commercial product], then you may use this software for free. If you wish to make money out of my software or charge me for yours then you don't get to use this for free”. That sounds fair to me.)*

Notice that this is a bit more all embracing than normal rules, because it rules out shareware writers. This does not apply to the resources here, which are free for all, and thus this stack is distributed separately. Related to this, the text of cHCcolorText should not be used for any other purpose.

Your attention is made to the disclaimer in the XCMD stack, which essentially denies all responsibility, but gives you the right to distribute this material as you see fit – apart from restrictions noted above. If you use the XCMD stack in your stack, and wish to distribute it, then obviously you can include the XCMD stack with your own. In fact, if you place it on archive sites, then you should include the XCMD stack – it is now quite happy to be disbributed on its own. However, I think it fair that you also point out that there is a full release, and indicate where it can be obtained from. Two reasons: there may be a newer version, and any recipients may be interested in using it themselves – and in trying to understand your scripts. The XCMD stack name deliberately does not include a version number, so that new versions can be installed easily. I’ve tried to maintain upwards compatibility with previous versions – not guaranteed, esp. not on monochrome systems – and I expect this will be a future goal too.

The following was included in the previous stack, and still applies:

I’m not going to guarantee to maintain this. It all depends on my interest – and whether I keep using it and see the need. If anybody wants to add features or fixes, feel free to – although please do let me know if you do. If you do so, it would help if you consider the following:

- You should only really think of adding features now in the last resort – the scripts are getting over complex as they are. [Too many conditional statements affects performance.] If you do add a new feature, try to work in an example, or provide one in another stack.

- Try to ensure your changes are upwards compatible – at least according to what the Info stack says should happen.
- Update the Info stack as well, if appropriate. Update the histories.
- Check on the functionality under a monochrome environment. All of the stacks distributed here will work after a fashion on monochrome, and should continue to do so.
- If possible, check the performance on a Classic or a Plus – this is a major problem, but some of the interim versions had even worse performance factors, and some thought has gone into containing the problem. [In reality, a few XFCNs or XCMDs would probably do wonders – hint anyone?]

If you send me any fixes, I'll try and co-ordinate them. Similar comments to BungDabba (whoever you are!), if you every write a new version of the XCMD itself, please let me know.

Anyway, I hope the examples wet your appetite enough to actually start using this stack – it would be nice to see some more colour stacks around! It can really make a difference to their look.

September 1993,

John Forrest
Dept of Computation,
UMIST,
Manchester M60 1QD,
UK.

E-mail: jf@ap.co.umist.ac.uk

PS. To use this system properly, you will need a colour editor of some kind. I've seen at least one free and usable colour paint program, so if you are strapped for cash try looking around.

Wish List

The following is a set of features that seem useful, and perhaps could be worked in to a new version:

- Simple graphics editor – it ought to be possible to provide some simple graphics editing, and then automatically generate the script to create that effect. Typical features would include placing bitmaps and colour fills – either relative to buttons/fields, or by allowing users to select an area and use that. This would be in "Colorizing HC Aids".
- Support for colour palettes and/or colour lookup tables – to be honest, I get confused between these two. It would be useful to enquire the values in the palette and change the current application palette – not least, to be able to set up the best colour in non-24 bit situations. This would require some externals. If anyone knows of some that already do this, and are preferably freeware, do let me know!

- At least for a while, there was a commercial product called Hypertint that did most of what ColorizeHC does and a lot more besides – colour text, effects etc. It might be possible to develop a bridge stack that ment that Hypertint called be used with a Colorize XCMD/colorizeHC interface. This would at least support effects properly.