

## Colorizing HC XCMD 1.1

This is the second release of this stack, and is much more comprehensive. The previous release – v1.0 – was an attempt to improve on the usability of the ColorizeHC XCMD that came with the Colorizing Hypercard 1.0 stack. It did this by standardising on the handlers where colour is displayed – at least, on the places where initial colour is drawn. ColorizeHC is not the only command available for adding colour to stack windows, but there are none cheaper – **it's free!** Note the XCMD itself is unchanged, it is the support scripts that are radically enhanced.

This new version of the XCMD stack goes far beyond basic colouring – **it provides support for any stack that must somehow modify the look of the screen as cards are opened.** It now provides some support for monochrome equivalents, and in particular the placing of windows (picture, pictoids etc.) over the card. Even if the card merely changes the visibility of buttons and fields as the card opens, use of this stack will result in a more orderly transfer. As such, I hope you find it a general tool that is not confined to use of the colorizeHC XCMD. [Quite a bit of the new stack was originally written on a Classic when I was off work, sick!]

**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. It has been tested on both a Classic and an LC running System 7.0 and 7.1.1 tuneup.

### 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.

### Adendum for v1.1.2

This XCMD stack has a few extra features over v1.1: message HCinit can be sent to allow the initialisation of colour, eg. putting which colours to use in global variables; and the stack will now adjust the depth to fit the available memory - although it moans if it has to. At the same time, the Aids stack has been introduced to reduce the size of the XCMD script.

## More Info

Version 1.0 was a single stack, that contained the scripts and the info pages on how to use it. In the current version, this has been split into 4 – the basic XCMD and support script, the Info pages, the supplied bitmaps, and 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. See note below about F.Rinaldi copyright.
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.
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 PictoidThe 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.

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 distributed with the standard Colorizing HC release. You should find these externals easy to obtain, but please check the conditions of use that come with them if you are engaged with commercial exploitation of Hypercard. 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. Don’t try to do so unless you have installed this XCMD in that stack.
- Stack “Colorizing HC Debug” requires Textoid in order to display its own window. You should copy that XCMD and also the “rinaldoid” WDEF and CDEF. Without this, output can only be sent to the message watcher.

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. 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. 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.

Having said this, 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. However, 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 XFCN’s or XCMD’s 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 1992,

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

E-mail: [jf@ap.co.umist.ac.uk](mailto: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:

- Allowing pictures etc. to be added to the window list, but not be automatically shown – they would be automatically killed. This covers places where you switch between several windows when in a card, and only wish a subset to be visible at the beginning. At the moment, these must be created, shown and closed explicitly in colorize and unColorize handlers.
- Simple graphics editor – it ought to be possible to provide some simple graphics editing, and then automatically generate the script to create that effect. This could be done as an extra stack, to avoid having to clutter the current one. 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!