Introduction for Version 1.2:

This is a palette based on the wonderful (2D)graphing application *nxyplot*, written by Tom Pulliam and Dennis Jespersen. *nxyplot* may be obtained with source code from the archives, such as sonata.cc.purdue.edu. In no way should any flaws in this palette reflect on their work. There are a couple of reasons for this- first, this palette is based on early versions of *nxyplot*, much of which has changed with later versions. Second, I have not worked on this as a good example of palette code, quite the contrary my code is somewhat convoluted (see below), but mainly for functionality.

How it works:

As with any palette, this one is a is a subclass of View, but think of it as a piece of graph paper which you can add to your application. You may set certain parameters (like is it Log-Log paper) using the Inspector panel. Unlike graph paper, it will do autoscaling.

How to use it:

Load the palette into IB either by double clicking on the palette, or using the IB menu item in Tools. Drag the view into your application and set the parameters with the Inspector. You may test the graph in IB's test mode by clicking the right mouse button on the view when in IB's Test Mode. This will bring up an Open panel which may be used to load a data file. *Be sure to load the palette*

before trying to load the examples.

To compile your application, it will be necessary to add the library libNXYpalette.a (an archived library of the compiled class code which should be put in a suitable directory, such as /usr/local/lib) and header files NXYView.h and Plot.h to your project. You may add all the source files to your project instead of the library file. You will also needed to add the Makefile.preamble to your project which identifies the classes to be loaded (the -u linker option.)

To plot your data, it will be necessary to write it to a (NXStream)stream. See the sample code which opens a file then writes the data to a stream. Your object which does this should then be connected to the graph view as the delegate of the view. Then include the method

```
-nxyView:sender provideDataStream:(NXStream **)stream;
```

in your object. NXYView checks to see if this selector is present before trying to access the data stream. If either the selector or the delegate is not present, an Alert panel will come up with the option of accessing data via an Open panel (see ~/nxyTestAppNoStream/TestNS.debug for an example of this.) This way the user may set up buttons which provide for the stream and test the application, either in IB test mode or by compiling, before the stream code is

present. See the test examples for more details.

Bugs and other comments:

Much of the functionality of nxyplot is not present in this palette; for example, it will not handle multiple files (data sets) although it will handle multiple columns of a single data set, but only with a single line type (that is, no dotted lines, etc.)

Sometimes when switching between data sets with the automatic scaling set will cause an Alert panel to appear saying that nxyplot is confused. Just continue, the proper min/max values are being set despite this Alert.

Small y-axis values sometimes get labeled as 0 (zero).

As I said before, I do not consider this a good example of code since I wrote it in a somewhat convoluted fashion. For example, there are times when the **view** messages **plot** for a value which in turn messages the **view** for the value. Why not just have view message itself? There are several (unclear) reasons for this. One is to stay consistant with the nxyplot code by rewriting as little as possible (or you could say I am just lazy.) The second, and most important reason to me, is that I still hold out that I will add an "Inspector" panel that "travels with the view." That is, it will become part of the users

application, so that the palette will work in the users application much the same way nxyplot works. [I vaguely know how to do this, and it has in fact already been done. Mike Mezzino has written such a palette that handles 2D and 3D plots, based on the plplot library, and has many of these features and more. Contact Mike for pricing and licensing. Also check out the graphing palette from Objective Technology.]

Report any bugs or strange behavior to me (charlie@technosci.com -- NextMail okay). The usual disclaimers apply--remember it's free and you get what you pay for.

Bug Fixes and Additions in 1.2:

Some minor bugs have been fixed--Text strings for axis labels and titles should no longer be clobbered when working in IB with multiple instances of NXYView; Bug in logo code that caused multiple page printing in IB test mode (and only in test mode) has been fixed; Some of the "ping-pong" code has been removed.

NXYView now accesses the data stream via a delegate. This is a better scheme for generic code (see the NeXT Advantage Plotter example.)

Pasteboard is now supported--the contents of NXYView may be copied to the

pasteboard to be pasted into other NeXT applications.

A switch on the Inspector to turn off the logo has been added.

Future Enhancements:

When I get the time, I hope to do the following additions to the palette (but not until after NextStep 3.0 is released):

> Add multiple line types and color (code is there for some of this is you want to do it)

> Add support for multiple files (code is in nxyplot 1.7)

> Add an Inspector Panel callable from the user's application

> Add legend support (again, the code is already there)

Acknowledgement:

My thanks to Tom Pulliam and Dennis Jespersen for their fine plotting package to build on.