

Yap

Yap, *Yet Another Previewer*, is a simple text-editor which provides the ability to interactively execute its documents as PostScript programs. Yap demonstrates:

- Execution of random PostScript code in a separate window server context
- Caching drawing for fast scrolling purposes
- A fairly generic document class managing multi-window text documents loaded from nib files
- Find/Replace panel

YapOutput.m and YapWrap.psw contain the code which creates a separate context that is focused on a window belonging to the application's main context and send PostScript down to this second context for execution. This second context is kept around as long as no errors occur; however, in case of errors a new one is created.

If your application needs to import EPS files, then its probably best to stick to the NSEPSImageRep class (or NSImage, which uses NSEPSImageRep to manage its EPS representations) rather than using the code in YapOutput.m. NSEPSImageRep uses a separate context to execute EPS files in a manner similar to Yap and will also properly deal with bounding box and font information present in EPS files.

Document.m contains the Document class, which manages the documents. The source code for Document demonstrates, among other things, how to read a nib file

over and over again to create multiple document windows. YapDocument is a subclass of Document to add the few Yap-specific features.

Preferences.m contains a preferences handler for getting values in and out of defaults. It uses an instance of IntegerFormatter to deal with integer input in text fields. (The Foundation Kit contains an industrial strength number formatter, but this simple one is provided for educational purposes...)

Finally, note that the "install" rule in the Yap Makefiles not only attempts to install the application in the release as it comes from NeXT, but it also attempts to copy the source over to the Examples directory. Thus it's not too useful for you.