1. Introduction

This manual documents some configuration and ins
GNUstep GUI Library and also differences between
libraries that implement the OpenStep AppKit specif
MacOS-X AppKit implementation.

1.1 Overview

The GNUstep GUI Library is a library of objects usef
applications. For example, it includes classes for dr
graphics objects on the screen: windows, menus, bu
fields, and events. There are also many peripheral c
operating-system-independent interfaces to images,
pasteboards, printing. There are also workspace su
data links, open/save panels, context-dependent he

1.2 Implementation Details

Following are some implementation details of the GUI library. Thes
mostly be of interest to developers of the GUI library itself.

1.2.1 Drag and Drop

The drag types info for each view is kept in a global map table (pro
by locks) and can be accessed by the backend library using the fur

```
NSArray *GSGetDragTypes(Nsview *aView);
```

Drag type information for each window (a union of the drag type inf
the views in the window) is maintained in the graphics context. The
backend can get this information (as a counted set) using -

```
- (NSCountedSet* ) _dragTypesForWindow: (int)winNum;
```

Whenever a DnD aware view is added to, or removed from a windo
type information for that view is added to/removed from the type inf
for the window, altering the counted set. If the alteration results in a
in the types for the window, the method making the change returns

1.2.2 NSWorkspac e

Here is (I think) the current state of the code (largely untested)
The make_services tool examines all applications (anything w
.debug, or .profile suffix) in the system, local, and user Apps D
In addition to the cache of services information, it builds a cac
information about known applications (including information a
types they handle).
NSWorkspace reads the cache and uses it to determine whicl
to use to open a document and which icon to use to represent
document.
The NSWorkspace API has been extended to provide methoc
finding/setting the preferred icon/application for a particular file
NSWorkspace will use the 'best' icon/application available.
To determine the executable to launch, if there was an
Info-gnustep.plist/Info.plist in the app wrapper and it had an N
field - use that name. Otherwise, try to use the name of the ap

MacOS X format in the Info-gnustep.plist/Info.plist for the
http://developer.apple.com/techpubs/macosxserver/Syst
/Developer/YellowBox/ReleaseNotes/InfoPlist.html.
In the NSTypes fields, I used NSIcon (the icon to use for
NSUnixExtensions (a list of file extensions correspondin
NSRole (what the app can do with documents of this typ
cache, I generate a dictionary, keyed by file extension, w
the dictionaries containing the NSTypes dictionaries of e
that handle the extension.
I tested the code briefly with the FileViewer app, and it s
the icons as expected.
With this model the software doesn't need to monitor loa
just register to recieve notifications when the defaults da
and check an appropriate default value. At present, there
files used by the software:

`~/GNUstep/Services/.GNUstepAppList'

wrapper. An app wrapper then need only contain a
understands the -GSOpenFile argument and uses i
though provision of a GNUstep-info.plist and variou
make things prettier.
For instance - you could set up xv.app to contain a
would start the real xv binary passing it a file to ope
argument was given. The Info-gnustep.plist file cou

```
    {
 NSExecutable =  "xv";
 NSIcon = "xv.t iff";
 NSTypes = (
  {
    NSIcon = " tiff.tiff";
    NSUnixExte nsions = ( tiff,  tif );
```

adaptation into libgnustep-gui. FSF must
various editorial changes in code. Very o
formatting, maintenance of various conve
always given authorship credit and show
approval.
Contributors must assign their copyright t
upon acceptance. Assigning copyright to
may be freely distributed.
Assistance in providing documentation, t
support is strongly encouraged.

Extensions, comments, and suggested m
libgnustep-gui features are also very wel

[<<] [>>] [Top]

This document was generated by Adam Fedor on