

3.3 Release Notes

Preferences

This file contains release notes for the 3.3, 3.2, 3.1, and 3.0 releases of Preferences. There were no changes for Release 3.3 or 3.2.

Notes Specific to Release 3.1

This bug of interest to developers was fixed in Release 3.1:

Reference: 31029

Problem: A module won't load if it's linked with libdbkit_s or libnikit_s

Description: Preferences couldn't load a module that referenced routines in the dbkit or nikit libraries.

Notes Specific to Release 3.0

These notes were included with the Release 3.0 version of Interface Builder.

New Features

The following new features have been added to Preferences since Release 2.0.

- Display Preferences—The color of the display's background can be set.
- Localization Preferences—Preferences contains a new layout for 3.0. The Localization layout provides control over all aspects of system localization. The keyboard mapping, system language, and measurement units can be specified.
- System Preferences—Several new font options have been added.
- Dynamic layout loading—Support has been added to allow loading of layouts into the Preferences application. This feature is described briefly below and in more detail in **[/NextLibrary/Documentation/NextDev/GeneralRef/15_Preferences.](#)**

Layouts are added by creating a directory for the new layout in

/NextLibrary/Preferences, **/LocalLibrary/Preferences**, or **~/Library/Preferences**, or by opening a **.preferences** file package from Workspace. The name of the file package should correspond to the name of the layout. The file package must contain at least two files, a tiff image, and an object module both with the same name as the directory. Additionally an InterfaceBuilder file may be contained in a **.lproj** directory. For an example look at the directory named **Template.bproj** in **/NextApps/Preferences**. The directory **/NextApps/Preferences.app/Template** can be renamed to "Template.preferences" and opened from Workspace. The file package named "Template.preferences" contains the following files.

```
/NextApps/Preferences/Template.preferences/Template  
/NextApps/Preferences/Template.preferences/Template.tiff  
/NextApps/Preferences/Template.preferences/English.lproj/  
Template.nib
```

The TIFF image file is the icon that will appear in the scrolling list of icons at the top of the Preferences window.

The object module should contain a subclass of the object Layout and should be named after the layout. The header for this object is located in **apps/Preferences.h**. This class will be instantiated when the layout is loaded. If your layout contains more than one mach object module then you should create a single object file using `ld`. When creating the Mach-O you

should use the "-r" option of **ld** to retain relocation symbols. For example:

```
ld -r -o Template.o AnotherClass.o funcs.o Template.o
```

This would create a single Mach-O called **Template.o** that contained three individual Mach-O's, **Template.o**, **AnotherClass.o**, and **funcs.o**. Note that the subclass of Layout must be specified as the last argument on the command line.