This is README for the Starter App, written by
Robert Vasvari Dec 1994.

The Starter Application

What is Starter ?

Starter is an app that does nothing and looks great doing it... :)

When a developer set out to build a new app, there are many
functionalities and mechanisms that need to be implemented.
The purpose of the Starter app is to provide the developer
with a framework that has all these mechanisms already working
in some basic way, so that instead of implementing them from
scratch, they can simply be extended in a kind of "fill in the

blanks" way.

Starter is great for developers who are not yet familiar with all the basic Mechanisms in NEXTSTEP and would still like to write an app that conforms to the general guide lines of NS. The code is not complicated, and browsing through it is a great way to learn NEXTSTEP progrmming.

List of these general features:

1: Create, destroy, open, revert and save document files

All documents are in their seperate zones.
When a document is closed, its zone is destroyed.
All three save functions (Save, Save As, Save To) are provided. Saving documents is done by opening a stream, putting clear text into it, then saving it to a file.

2: Multiple screen Inspector panel that changes with the currently selected document.

This Inspector is modeled

after the Workspace Inspector. The top part contains
a textfield for the filename that will change its font
to make sure the entire name fits (within reasonable
limits, of course). The default implementation contains
two screens. More screens can very easily be added. The
screens can be invoked by by typing <COMMAND><Screen#>
just like the Workspace.

3: Multiple screen Preferences window.

Each screen contains
items that are saved as defaults for the program.
The Set and Revert buttons become enabled/disabled according
to changes made (even on the textfields).
Pressing the Set button saves the changes into the Defaults
Database. There are utility routines provided to get defaults
of different types with error checking (util.h/m).

4: Menus that enable/disable themselves according to current selection.

This done through the [NXApp updateWindows] method
which gets called after any change in the program. The method
in turn calls each window in the app (including the menus) to
update themselves.

5: **A Controller object** that is the delegate of NXApp, it performs all interfacing between the menus, the worspace and the documents. Also keeps a list of documents. A document can either be opened or just put to the fron if it is already open. Upon termination a standard process is performed to save altered documents.

6: **A RemoteManager object** is a simple Distributed Objects interface. It makes Starter a server, such that other apps can remotely message it to open, hide docs, etc... These simple operations demonstrate the ease of use for the DO interface.

List of classes:

AdjFontTextField:	changes fontsize according to contents.
ClockView:	Displays date graphically in the inspector.
CommandView:	routes <COMMAND><Screen#> keystrokes to the inspector.
Controller:	Application's delegate.

DocumentWindow:	sends updateWindows message on Miniaturize.
Document:	Generic document that Save/Load itself.
InspectorManager:	subclass of MultipleScreenManager that dynamically changes its contents according to the current selection.
KeyPanel:	subclass of panel with a button that shows a return sign only if the panel is the Key Window af the app.
MenuController:	subclass of controller that updates all menucells.
MultipleScreenManager:	Changes the screens by the popup button on the top of the panel.
PreferenceManager:	subclass of MultipleScreenManager, saves changes per screeen.
RemoteServer:	provides methods that can be called from other apps through the Distributed Objects interface.
util.m/h:	utility routines.

How do I use Starter ?

```
There are a few simple steps to perform to change the name to the new name
of your application:
1: Change the name of the project directory to AppName;
2: Change the name of English.lproj/Starter.nib to English.lproj/AppName.nib;
3: in defs.h change the value for DOC_EXTENSION to the new document ext.;
4: Start Project Builder by double-clicking on PB.project;
   a: In the Attributes Window, change project name and doc extension;
   b: In the Files/Interfaces Window remove Starter.nib and add AppName.nib;
   c: Save PB.project
5: build app;
6: At this point you are ready to implement your own features..
```

Foundation Kit users:

```
I also have a version of Starter App that is moreless "foundationized"
That version is available upon request.
```

Correspondence:

```
Comments, bugreports, suggestions should all go to:
```

vrobi@futon.sfsu.edu

Enjoy!!!

=[vrobi]=