

# TEMPLATES

A Mac OS X Service for the Finder

Version 1.1

Copyright © 2010, Jamie Curmi (<http://curmi.com>)



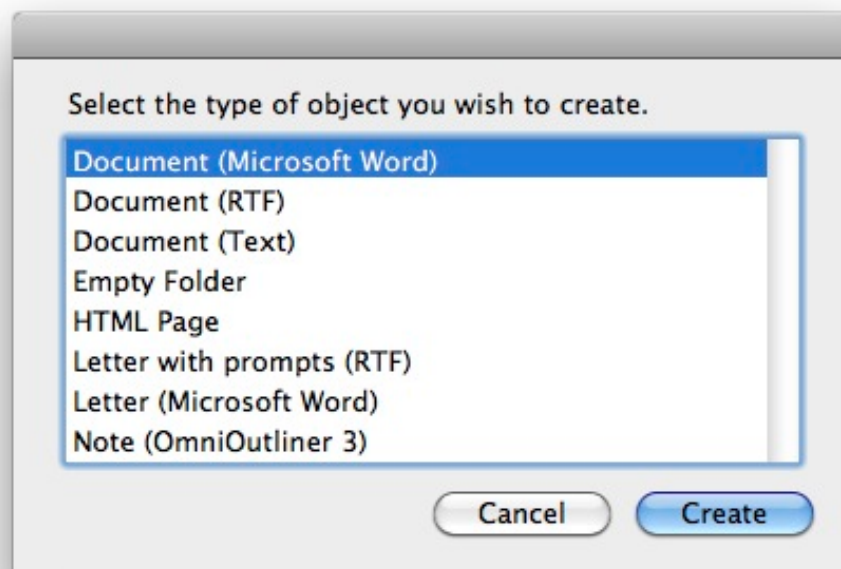
## ABOUT TEMPLATES

Templates is a Mac OS X Service for the Finder, allowing a user to make any system object a template, with the ability to quickly and easily create new instances of that object based on this template. Templates can also take advantage of AppleScript in documents, allowing for scripted, and even interactive, behaviours in creating an instance.

What's New in v. 1.1  
*Updated package allows installation for all users, or just the current user.*

System Requirements  
*Templates requires Snow Leopard (Mac OS X 10.6)*

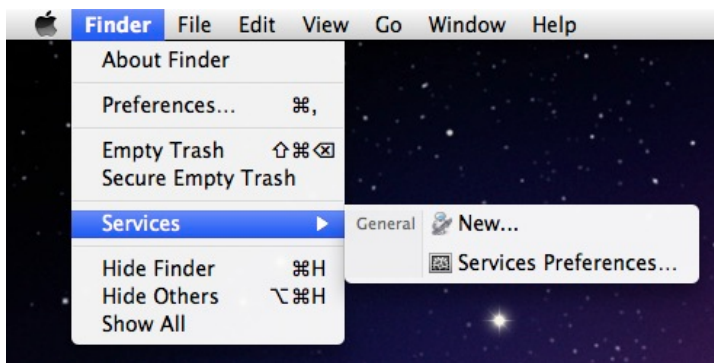
Do you ever find yourself copying one file to another, to ensure you have the same text or formats in a new document? Ever wanted to quickly create a new text file in a particular folder? Make a particular file a template for other files? Welcome to Templates.



# TEMPLATES

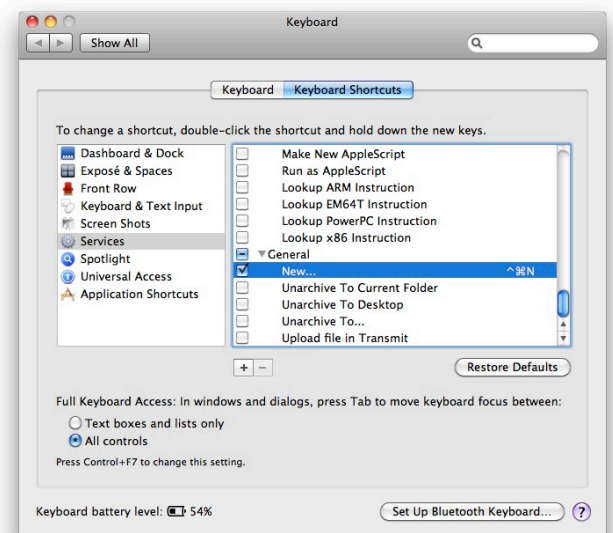
## INSTALLATION

Templates most likely came to you as an OS X package. Your browser may have automatically opened the package, otherwise locate it and double-click it, then follow the instructions to install the software.



Once installed, a new Service will now be available in your Services menu in the Finder. Go to the Finder, then to the Finder application menu, and choose Services. You should see something similar to that shown here.

It is best if you assign a keyboard shortcut to this Service, so that it is easily accessible. You can do this by choosing Services Preferences... from that menu, and in the preferences, adding a shortcut to the New... service. A suggested shortcut is CTRL-COMMAND-N.



## SETTING UP TEMPLATES

Template objects are stored in one of two locations:

1. /Library/Templates - for access by all users
2. Your home Library/Templates - for private use by you alone

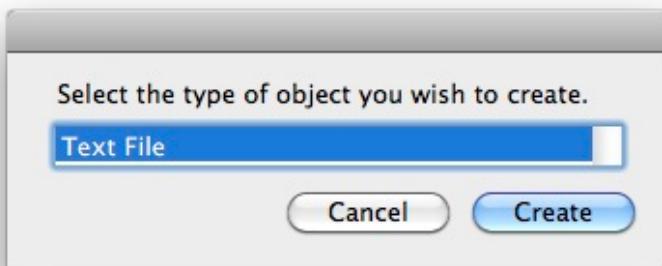
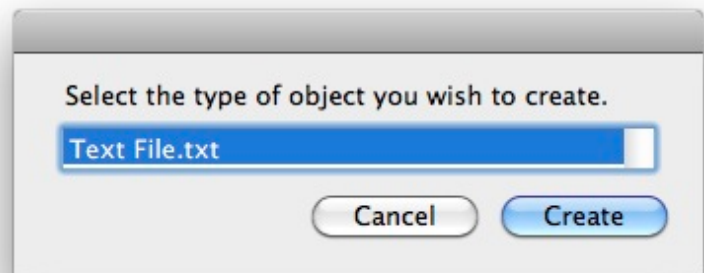
When you installed Templates, a default set of Templates is installed in the first location (unless you choose to install in your home area). You can create a new Templates folder in the other location for private use.

Now, you simply put any objects you wish to be templates in to one of these locations.

# TEMPLATES

## THE BASICS

Placing a simple text file, say `Text File.txt`, in to a `Templates` folder will allow that text file to be used as a Template. When you start the service from a Finder Window (from the services menu or your keyboard shortcut), a selection window will appear. You can either create the object in the current Finder window (or Desktop), or cancel.



However, Templates will also honour metadata you have on your Templates. For example, if you hide the extension on the template file, and even change the template file so that it opens by default with another editor other than Text Edit (say BBEdit), the dialog will hide the extension, and the created file,

when double clicked, will open with the alternate editor.

Of course, Templates is a lot more interesting when you have a more complete library of Templates.

## METADATA

Templates has its own set of metadata you can apply to any template object giving you the ability to:

- Change the display name of the object (the default is to use the filename)
- Change the default name of the object (the default is `Untitled`)
- Ask for a new name for the object

# TEMPLATES

- Open the object after creation (the default is to select in the Finder)
- Interpret any AppleScript in the object before creation

To do this you simply add appropriate flags to the Comments area of the object (select the template object in the Finder, go to the File menu, and then Get Info). The flags available are as follows:

- `:display=Display Name:`
- `:default=Default Name:`
- `:ask=true:`
- `:open=true:`
- `:interpret=true/file name:`

These can be strung together in to one large string, for example:

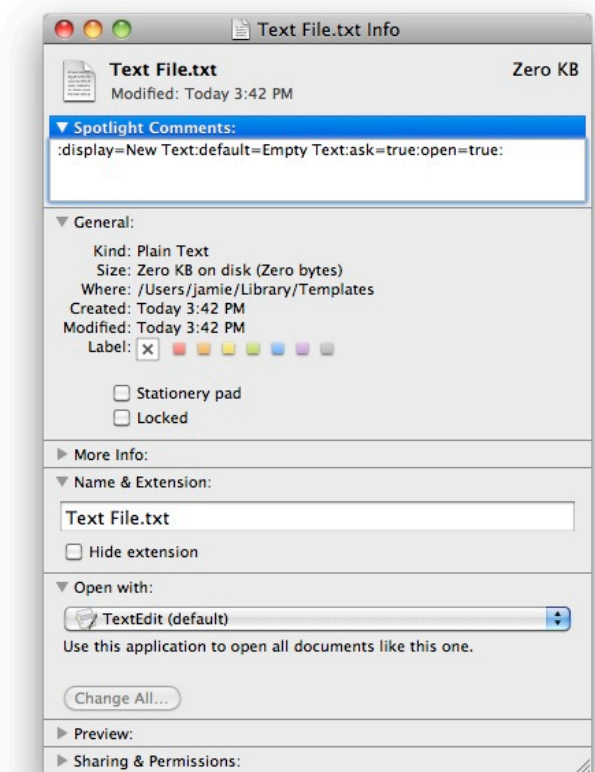
```
:display=New Text:default=Empty
Text:ask=true:open=true:
```

The example templates provided have other examples of these flag strings.

Further, you can use AppleScript to provide a default name, by placing AppleScript commands between `[[` and `]]` in the flag string. Each line of a multiline AppleScript should begin and end with `[[` and `]]` in the Comments field. As an example, the following string:

```
:default=New Note - [[current date as string]]:
```

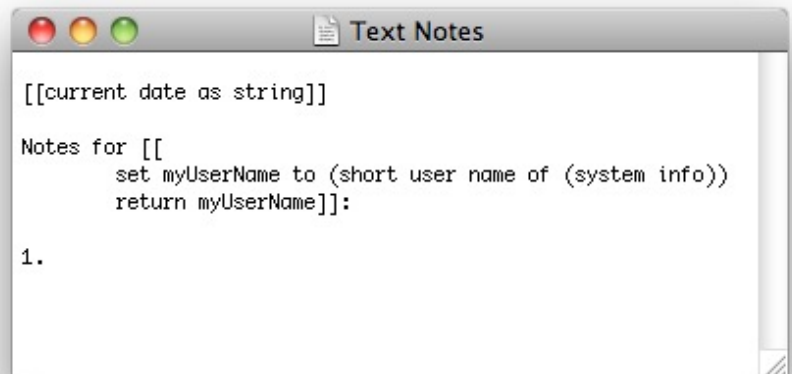
will ensure that any instance of this template will default to a name with the current date in it, for example "New Note - Sunday, 8 January 2010 1.47.30 PM".



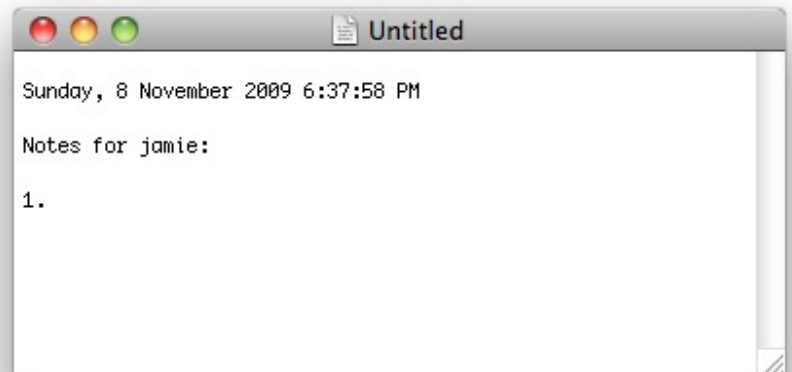
# TEMPLATES

## MORE APPLESCRIPT

For a single file template (not a bundle or directory), if you include the flag `:interpret=true:`, any AppleScript (between `[[` and `]]` markers) is interpreted before creation, and the text replaced with the result. For example, a text file template could look like the following:



And after creation, it may produce a file that looks like:



Further, the AppleScript can also be interactive. For example, using a script with text such as:

```
[[tell application "Finder"
    activate
    set result to display dialog "Enter name:" default answer ""
    set answer to text returned of result
end tell]]
```

# TEMPLATES

will produce a dialog on screen asking for the user to enter their name. The power of AppleScript means that your text documents (that includes Rich Text, HTML, XML etc.) can be quite dynamic, requesting user input, processing that input, and so on.

Some files you may wish to use as templates, however, may be bundles - they are in effect a folder with other files or folders within, that appear to the user as a single file in the Finder. In these cases, the file that you may want to interpret may be within this bundle. In these cases, use `:interpret=/file:` where `file` is the name of the file in the bundle. It could also be another folder with a file in that, for example, `:interpret=/Contents/file.xml:.` For an example, check out the "donate.rtf" bundle in the default Templates folder.

## ADVANCED APPLESCRIPT

There are certain variables that are set and can be used by any template interpreting AppleScript. These are:

- `__location__` : the path of the folder
- `__filename__` : the actual filename given to the instance
- `__name__` : the name without extension or numbers added for clashes
- `__extension__` : the extension of the filename, if any

See the sample template `HTML.html` for an example of the use of these variables.

Further, the result of every use of AppleScript is stored in a variable based on the order of interpretation, that is the result of first use is stored in variable `__1__`, the second in variable `__2__`, etc. This allows the user to access a previous result later in the text file.

If you want to perform AppleScript but not display that result - perhaps so you can use it later - you can do so by opening the AppleScript with `[ [!` instead of the usual `[ [`. The result will still be stored in the variable `__n__` (where `n` is the interpretation order number), but

# TEMPLATES

will not be displayed. See the sample "Letter with prompts (RTF)" for an example of use.

## TERMINAL INTERFACE

If you want to use Templates from the terminal, install the "new" script available as an additional download from the website in to an appropriate area on your system. Ensure the PATH variable in your shell points to the location of "new". You must still install the Service as discussed earlier for the shell script to work.

To create an instance of a template in the current directory, in the terminal you simply type:

```
new
```

You can also supply an argument - a number representing the number (from 1) of the selection you want to make. So if you know the template you want is number 4, you simply type:

```
new 4
```

## HOW MUCH DOES TEMPLATES COST

Templates is free. But if you feel like donating to help me to produce other cool products, please go to <http://curmi.com/donate>.

I hope you enjoy Templates as much as I enjoyed making it.

Jamie Curmi

<http://curmi.com>