The Component Description File: Description.plist

In order to unify the documentation of WebObjects' reusable components, a scheme has been devised to provide a structure for the common documentation elements of a component. This is a discussion of that scheme and how you can add your own description files to make your components self-documenting.

The basic idea is that all the documentation you might want to associate with a reusable component should be encapsulated inside the component itself. Also, it should be accessible at runtime so the documentation can be displayed from within applications such as code browsers, or component catalogs. This is done by placing all the documentation into the standard "property list" format within a standard file called "Description.plist." This description file can be parsed into a WOF app by simply initializing an NString from the Description.plist file, and subsequently sending that string the "propertyList" message. The result is an NSDictionary with the common documentation elements as key/value pairs.

As part of the scheme, a list of standard documentation elements has been provided; these are define below. However, as long as you adhere to the standard propertyList format, you may add your own definitions to suit your own purposes. Of course, only the standard elements (de facto or de jure) will be recognized by software other than your own.

The propertyList format allows for the ASCII representation of two container classes: NSArray and NSDictionary. To construct an NSArray, you must make a comma-delimited list of strings and enclose the list in parentheses. For example:

```
   ("Hello World", "now is the time", "for all good men")
```

will result in an NSArray of three NSStrings. To construct an NSDictionary, you must make a semi-colon-delimited list of key-value pairs all enclosed in curly braces ("{" and "}"). The key-value pairs are formed by placing an equals sign ("=") between two strings -- left side being the key, right side being the value.

Here's an example of an NSDictionary with three entries:

```
{
   "First Name" = "Charles";
   "Last Name" = "Lloyd";
   "EMail" = "clloyd@next.com"
}
```

When loaded into an application, the Description.plist file becomes an NSDictionary with numerous key-value pairs. You access the documentation elements of the reusable component via the keys which are predefined below. Some of the keys have values of NSString's, some have NSArray's, and some have NSDictionary's; the type stored is dependent upon the type of information being stored. For the standard elements the types are defined below.

It is important to note that the strings in this file can contain HTML. However, not all html will make sense. For example, relative URL will probably not work since these strings may be placed anywhere and you cannot control where that may be. Also, you must be sure to escape quotes (ie \") which are internal to the string since the propertyList parser will search for closing quotes.

Key	Value
componentName:	A string holding the component's name.
description:	A string for long description of the component. *This is required for the ReusableCompoents Demo framework.*
parameters:	An array of strings which describe the user-settable options of the component. The strings should alternate between a parameter name and parameter description. *This is required for the ReusableCompoents Demo framework.* [Note: Use of a dictionary here would have left the ordering of the parameters indeterminant, hence the choice of an array of alternating strings].
synopsis:	A string listing the component name and all of its possible parameters, even those that are mutually exclusive or optional.
shortDescription:	A string holding a on-line description of the component
authors:	An array of dictionaries, each one holding key-value pairs for:

name: String holding author's name
email: String holding author's email addr
image: String holding a path to author's image

company: A string holding the name and address of the company owning the code.

currentVersion: A string holding version information for the current component.

releaseDate: A string holding the date of release of the current component.

maintenanceHistory: An array of strings holding the reverse ordered maintenance history. The format of the strings is user defined, but one suggestion is to use the format:

<date>:<userId>:<description of change>

price: A string holding the price of the component.

license: A string holding the license agreement.

copyright: A string holding the copyright statement for the current component.

supportPolicy: A string holding the support policy for the component.

distributionPolicy: A string holding the distribution policy.

disclaimer: A string holding the disclaimer.

Below is an example of some of these documentation elements in Description.plist:

```
==============================================================

{
componentName = "MyComponent";

description = "This is a <i>really</i> useful component.
<br>
```

```
<br>
To use, simply plug it in, <b>add water</b>, and go to town.";

parameters = (
    "paramX",
    "The Xth parameter.  Used for determining the <b>Xth</b> thing"

    "paramY",
    "The Yth parameter.  Used for determining the <b>Yth</b> thing"

    "paramZ",
    "The Zth parameter.  Used for determining the <b>Zth</b> thing"
);

shortDescription = "Displays a <i>widget</i> and allows user to choose his favorite";

authors = (
    {
    author = "Charles Lloyd";
    email = "clloyd@next.com";
    image = "http://www.next.com/~clloyd/me.gif";
    },
    {
    author = "John Doe";
    email = "jdoe@dough.com";
    image = "http://www.next.com/~jdoe/me.gif";
    },
)

company =
"<h1>NeXT Software<h1><br>
900 Chesapeake Dr<br>
Redwood City, CA  94063<br>
415-780-9999";

currentVersion = "1.2.1";

releaseDate = "2/28/96";
```

```
maintenanceHistory = (
    "2/21/96:jdoe: Reworked all of Charles' cruft",
    "2/2/96:clloyd: Fixed ground hog's day bug",
    "1/25/96:clloyd: Added option for turning off radio buttons",
);

price = "No Charge";

supportPolicy = "Unsupported";

distributionPolicy = "You may freely distribute this code";

}
==============================================================
```