# Creating a Preferences entry

This development tip is formatted in both Microsoft Word 5.1 and plain text. In Word, all normal text is in the New York font. and all code is in courier.

When your application is run, some data (such as viewBounds) can be stored in the base view, and some (such as names) can be stored in a soup entry in your application's soup, but how about preferences, like font size and preferred delay? For this type of data, one should use a preference entry. This document will discuss creating a function for creating and accessing such an entry.

The normal storage for preferences is in the System soup, located on the internal store. All applications that wish to create preference entries should use their appSymbol (as set in the "Settings..." dialog accessed from the "Project" menu) as their tag. The tag is simply a string, usually defined in the Project Data, using a DefConst(...):

```
	DefConst(‘kAppName, “Benz Cool App:BenG”);
```

Now, wherever you need to reference your application’s name, you can just use `kAppName`.

A typical preference entry might look like this:

```
	local blankPrefEntry :=
		{
		tag:		kAppName,
		someData: "Boy, this is a cool entry"
		};
```

The tag is used to access the information from the System soup, which is indexed on the tag slot. In addition, some applications may in the future use the tag slot to determine an entry's owner, so you should probably use you're appSymbol, and not something like "Benz Cool App Prefs".

To create an entry, you need a reference to the system soup. This is done by using the magic pointer ROM_systemSoupName,  which, in US Newtons is “System”. Since the system soup should be only on the internal store, we don’t need a union soup:

```
	local sysSoup := GetStores()[0]:GetSoup(ROM_systemSoupName);
```

Now, make your entry and add it to the soup:

```
sysSoup:Add(blankPrefEntry);
```

The Clone(...) is necessary because the frame you pass it is modified, and if it was created at compile time (in the Tool Kit) than it only exists in the pseudo-ROM of the package. Cloning it allows changes to be made directly to the frame.

To access your entry, first create a cursor in the system soup like this:

```
local searchSpec := {type: 'index, indexPath: 'tag};
local systemCursor := Query(sysSoup , searchSpec);
```

Now that you’ve got a cursor, you need to move it to your entry:

```
local prefEntry := systemCursor:GotoKey(kAppName);
```

Because of the way that the :GotoKey(...) method works, if there is no entry in the system soup with the appropriate name, it will move the cursor to the closest tag. This means that we have to make sure we’ve got the right entry by doing a StrEqual(...) on both the entry’s tag and the application name. If they’re not equal, then our entry doesn’t exist:

```
IF prefEntry AND StrEqual(prefEntry.tag, kAppName) THEN
     return prefEntry;
```

Now that we can create and access a preferences entry, it makes sense to encapsulate all of this into one function that can be called from anywhere in our app. This is done by DefConst(...)’ing a function in the project data, which we will call when we want the prefs:

```
DefConst(‘kGetPrefsFunc,
func() begin
     local internal := GetStores()[0];
     local sysSoup := internal:GetSoup(ROM_systemSoupName);

     local searchSpec := {type: 'index, indexPath: 'tag};
     local systemCursor := Query(sysSoup , searchSpec);

     local prefEntry := systemCursor:GotoKey(kAppName);

     IF prefEntry AND StrEqual(prefEntry.tag, kAppName) THEN
          return prefEntry;

     prefEntry :=
               {
               tag:      kAppName,
               someData: "Boy, this is a cool entry"
```

```
                    };
          sysSoup:Add(prefEntry);

          return prefEntry;
     end);
```

Now, whenever we need prefs in our app, we can get them by using the line:

```
     local prefs := CALL kGetPrefsFunc WITH ();
```

This will call our function and return a prefs entry, whether one exists or needs to be created. This function can be pasted directly into your project data, just make sure to change the blank pref entry to suit your purposes.

If you have any questions or comments about this document, or requests for future similar documents, please feel free to email me at America Online, gottlb.

Ben Gottlieb (gottlb)
PDA Forum Consulant
Keyword PDA is how to find us!