

Q: When should I use **NXLocalizedString()** and when should I use **NXLocalizedString()** in my code? The localization chapter in `/NextLibrary/Documentation/NextDev/Concepts` just discusses **NXLocalizedString()**.

A: **NXLocalizedString()** is an old name, and you should use **NXLocalizedString()** instead. The name **NXLocalizedString()** is only there for compatibility.

Q: The program **genstrings** complains if **NXRunLocalizedAlertPanel()** contains a constant (like from a `#define`) instead of a string. This means that we may end up having "OK" defined multiple times in the string table, since most alert panels include an "OK" button. Should I avoid using **NXRunLocalizedAlertPanel()**?

A: **NXRunLocalizedAlertPanel()** will be obsolete in the future, and we strongly recommend that you use **NXRunAlertPanel()** instead. It's true that **genstrings** doesn't eliminate duplicate entries from **NXRunLocalizedAlertPanel()**.

Q: The localization chapter in `/NextLibrary/Documentation/NextDev/Concepts` suggests using `#define` to retrieve a string just once. It seems like if we want to avoid extra lookups, we should define a variable, initialize it once to the result of an **NXLocalizedString()**, and then use it where necessary.

A: You are right. If you want to avoid extra lookups, you should define a variable to have the value. For example, you could do the following:

```
const char *fooString = NULL;
#define FOO_STRING (fooString = fooString ? fooString :
                  NXLocalizedString("Foo", NULL, "Nonsense."))
```

That way, the string will only be looked up if it is needed and will only be looked up at most once. The macro is a convenient way to make sure that the string has been initialized each time you want to use it, especially if you want to use it as an argument.

Q: How big a performance hit do I take for a lookup from a string table? At what size is it worth breaking up a string table into multiple files?

A: Lookups are cheap. It's only worth going to multiple files if you have hundreds of entries, or if you have a large number of entries which are almost never accessed or which are only accessed together such as error conditions or strings for a certain panel like an Inspector Panel.

QA866

Valid for 3.0 only