Coding Guidance

Working With Strings

In order to support localization **YOU SHOULD NOT** have hardcoded UI display strings in your code. Instead, use resource files to consume strings.

For CPP

Use `StringTable` resource to store the strings and resource header file(`resource.h`) to store Id's linked to the UI display string. Add the strings with Id's referenced from the header file to the resource-definition script file. You can use Visual Studio Resource Editor to create and manage resource files.

- `resource.h`:

XXX must be a unique int in the list (mostly the int ID of the last string id plus one):

```
#define IDS_MODULE_DISPLAYNAME                      XXX
```

- `StringTable` in resource-definition script file `validmodulename.rc`:

```
STRINGTABLE
BEGIN
    IDS_MODULE_DISPLAYNAME              L"Module Name"
END
```

- Use the `GET_RESOURCE_STRING(UINT resource_id)` method to consume strings in your code.

```
#include <common.h>

std::wstring GET_RESOURCE_STRING(IDS_MODULE_DISPLAYNAME)
```

For C

Use XML resource file(.resx) to store the UI display strings and `Resource Manager` to consume those strings in the code. You can use Visual Studio to create and manage XML resources files.

- `Resources.resx`

```
<data name="ValidUIDisplayString" xml:space="preserve">
  <value>Description to be displayed on UI.</value>
  <comment>This text is displayed when XYZ button clicked.</comment>
</data>
```

- Use `Resource Manager` to consume strings in code.

```
System.Resources.ResourceManager manager = new System.Resources.ResourceManager(baseName, as
string validUIDisplayString = manager.GetString("ValidUIDisplayString", resourceCulture);
```

In case of Visual Studio is used to create the resource file. Simply use the `Resources` class in auto-generated `Resources.Designer.cs` file to access the strings which encapsulate the `Resource Manager` logic.

```
string validUIDisplayString = Resources.ValidUIDisplayString;
```

More On Coding Guidance

Please review these brief docs below relating to our coding standards etc.

- Coding Style
- Code Organization