

Invoke Native Methods with DllImport

Here are some samples:

```
import System.Runtime.InteropServices

[DllImport("user32.dll")]
def MessageBeep(n as uint) as int:
    pass

def beep():
    MessageBeep(0)

beep()
```

Actually, a more cross-platform way to make a beep sound is to call `Microsoft.VisualBasic.Interaction.Beep()` (add a reference to the `Microsoft.VisualBasic.dll` in the GAC or do `import Microsoft.VisualBasic from Microsoft.VisualBasic`). This is supported in both .NET and Mono.

```
import System.Runtime.InteropServices

[DllImport("msvcrt.dll")]
def puts (c as string) as int:
    pass

[DllImport("msvcrt.dll")]
def _flushall () as int:
    pass

puts("testing...")
_flushall()
```

Here is another example with the `EntryPoint` specified as a named parameter. Useful I guess if you want to use a different name for the dll call.

```
import System.Runtime.InteropServices

[DllImport("User32.dll", EntryPoint:"MessageBox")]
def msgbox(hwnd as int, msg as string, caption as string, msgtype as int):
    pass

def msgbox(msg as string):
    msgbox(0, msg, "Message", 0)

msgbox(0, "MessageDialog called", "DllImport Demo", 0)

msgbox("one more time")
```

For further information, see:

- [P/Invoke tutorial.](#)
- [Creating a P/Invoke Library](#)