Subject: Support for plug-in architecture
Posted by slashupp on Sat, 21 Mar 2020 02:21:03 GMT
View Forum Message <> Reply to Message

(linux)

I have an app that will work well with plug-ins, i.e. using custom-controls that are compiled into dynamic shared libs that can be loaded by the executable without needing to recompile the app itself.

My current design generates all as one big blob-executable and needs to be re-build for each small
change or addition of a custom-control, but would be much better using plug-ins.

Since Windows DLL's can be created, why can the linux-equivalent of dynamic shared object (.so) not
be created? It looks like a simple change to compiler and linker flags that will enable this?

I've hacked the [Setup/Build methods] flags to produce a .so lib, which works with well with 'extern "C"'-functions, but fails to pass an accessable custom-control (which is mangled C++) back.

Barring out-of-the-box support for plug-in/.so development, is there any other way to implement a plug-in design using Upp?