
Subject: Re: MacOS X woes

Posted by [lundman](#) on Fri, 29 Dec 2006 04:41:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

While looking into your "No 1" task, I discovered PackageOrganiser/Add/Remove-Flags.. siigghh.. all my complaining about no link-options. Anyway, it would be nice if the "WHEN" could also sense the BuildEnvironment-name used. "GCC32" and "GCC32CARBON" or what not.

Anyway, very simple carbon app compiled in U++ as "Core console project", straight out of developer.apple.com, so no credits to me:

UPPCarbon.cpp:

```
#include <Carbon/Carbon.h>
```

```
#include "UPPCarbon.h"
```

```
#define kWindowTop 100
```

```
#define kWindowLeft 50
```

```
#define kWindowRight 250
```

```
#define kWindowBottom 250
```

```
const EventTypeSpec  eventList[] =
{
    { kEventClassWindow, kEventWindowClose },
    { kEventClassWindow, kEventWindowActivated },
    { kEventClassWindow, kEventWindowDeactivated },
};
```

```
static pascal OSStatus MyWindowEventHandler(EventHandlerCallRef nextHandler,
                                           EventRef theEvent,
                                           void *userData)
```

```
{
    OSStatus result = eventNotHandledErr;
    WindowRef theWindow = (WindowRef) userData;
    UInt32 whatHappened;

    whatHappened = GetEventKind(theEvent);

    switch(whatHappened) {

        case kEventWindowClose:
            DisposeWindow(theWindow);
            QuitApplicationEventLoop();
            result = noErr;
```

```

        break;

    case kEventWindowActivated:
        ::CallNextEventHandler(nextHandler, theEvent);
        result = noErr;
        break;

    case kEventWindowDeactivated:
        ::CallNextEventHandler(nextHandler, theEvent);
        result = noErr;
        break;

}

return result;

}

void Initialize(void)
{
    // Do one-time-only initialization

    WindowRef          theWindow;
    WindowAttributes    windowAttrs;
    Rect                contentRect;
    EventHandlerUPP     handlerUPP;

    windowAttrs = kWindowStandardDocumentAttributes |
                  kWindowStandardHandlerAttribute;

    SetRect(&contentRect, kWindowLeft, kWindowTop,
            kWindowRight, kWindowBottom);

    CreateNewWindow(kDocumentWindowClass, windowAttrs,
                    &contentRect, &theWindow);

    SetWindowTitleWithCFString(theWindow, CFSTR("U++ Carbon Example"));

    handlerUPP = NewEventHandlerUPP(MyWindowEventHandler);

    InstallWindowEventHandler(theWindow, handlerUPP,
                             GetEventTypeCount(eventList), eventList,
                             theWindow, NULL);

    ShowWindow(theWindow);

    InitCursor();
}

```

```
}
```

```
void Finalize(void)
{
}

}
```

```
using namespace UPP;
```

```
CONSOLE_APP_MAIN
{
    Initialize(); // Do one-time-only initialization

    RunApplicationEventLoop(); //Process events until time to quit

    Finalize(); // Do one-time-only finalization
}
```

Extra link flags, and post-link step added:

UPPCarbon.upp:

```
flags(OSX11) "-framework carbon";
```

```
uses
    Core;
```

```
link(OSX11) "-framework carbon";
```

```
file
    UPPCarbon.h,
    UPPCarbon.cpp;
```

```
mainconfig
    "" = "",
```

```
custom(OSX11) "post-link",
    "mkdir -p $(OUTTITLE).App/Contents/MacOS\ncp $(OUTPATH) $(OUTTITLE).App"
    "/Contents/MacOS",
    "",
    ,
```

And yes, creating the UPPCarbon.App/Contents/MacOs/ is required for it to fully function.

Looks like:

<http://netbsd.interq.or.jp/~lundman/carbon.jpg>

File Attachments

1) [carbon.jpg](#), downloaded 951 times
