

Writing your First Script Xtra

By Noisecrime 2002

www.noisecrime.com

Attachment: myFirstXtra.zip

Summary

This document has been written in an attempt to help new developers to the world of Director Xtras. This guide is focused purely for the PC using MSVC++.

Although I would not classify myself as an Xtra guru, I think everything included in this document should be correct. If you find any errors, omissions, or poor xtra coding practice please email me at noise@noisecrime.com

How to begin

First you'll need the Director XDK 8.5 from MM website. Download it from ?

Of course you'll also need a compiler ideally for PC's that would be MSVC++ v6 as this has been used to create and compile all the examples that come with the XDK.

One final item is a GUID generator. This will generate a unique ID everytime its used, and is required for every Xtra . If any two xtras have the same GUID, Director will complain about duplicate xtras.

Setting up

Its best to be organised from the beginning so create a new directory on your hard drive to keep all your xtra development work in. For example in this Doc will call it "Xtra_development". Unzip the D8.5 XDK into this folder, making sure the contents of the XDK are a level down (xtra_development /xdk85_win/contents).

Keeping things tidy, create another directory within 'Xtra_development called 'Xtras', this is where you can keep all the xtras that you are working on.

The main reason I'm stating this is that you have to add an include path to MSVC to the XDK include folder. With an outline of the directory structure above, it should be clearer to describe.

Finally make a copy of the Skeleton scripting Xtra from Xtra_development/xdk85_win/examples/script/ to Xtra_development/Xtra/Skeleton

Make sure you use skeleton, not skeleton 2. The only difference is that in skeleton2 the registration and xtrascript sections have been combined into one, and whilst this is better, its not so easy to learn from.

Skeleton to myFirstXtra

When I open a MSVC workspace on a project that has been copied or moved to another folder I find all the links to the files retain their original directory path. So if I was to edit script.cpp in my copied skeleton xtra folder, the original file not the copied one would get updated. In addition I found other similar little problems with MSVC, so I have developed my own practice when starting a new xtra. Because the skeleton xtra is already set up for MSVC I don't want to lose its project settings, but basically I clear out everything else and start from scratch.

Changing names

1 Select the 'Skeleton' folder, and change its name to 'myFirstXtra'

2. In myFirstXtra/winproj folder delete the following file types .ncb, .opt. MSVC only needs .dsw, .dsp, .def.

3. Change the name of every file left to myFirstXtra (e.g myFirstXtra.dsw)

4. Open (double click) the myFirstXtra.dsw file in MSVC

It will complain that it can't find script.dsp, simply browse and find myFirstXtra.dsp and select o.k.

5. In MSVC select the 'fileview' tab, and delete all the files shown (not folders).

6. At the top of the 'fileview' window, right click the Icon/directory called 'script files'.

From the popup menu select 'Add files to project'

A browse box will pop up, navigate to the source folder of myFirstXtra, select all the files and import. In 'fileview' you should now find all the .cpp files in the source folder, and all .h files in the header folder.

7 Bring up the Add files to project browse box again, change the type to any (*.*), go into the winproj folder and select myFirstXtra.def.

Once imported, double click myFirstXtra.def and change the LIBRARY name from SCRIPT to MYFIRSTXTRA – keep the CAPS!

Once changed save it.

8. In MSVC go to the 'Project/settings' menu item and from the pop up window, select the 'Link'

Tab, under the 'general' Category, change the 'Output file name' from script.x32 to myFirstXtra.x32 in both Debug and Release modes (keep the directory just change the name).

9. In the same window select the 'C/C++' tab, and choose the Preprocessor category.

Under Additional include directories change the include link to the XDK include folder. In our Directory structure this should be (..\..\xdk85_win\include)

Note: If running the Skeleton xtra original this Include path should be correct.

10. Open the cregister.cpp file, and scroll down to the message table.

Where it says "xtra ScriptingSkeleton\n" change it to the name of your xtra so it reads 'xtra myFirstXtra\n'.

You'll see the skeleton xtra (now myFirstXtra) template already includes 3 handlers, a global, parent and child. Although we could leave them as they are, it would be better to give them some different names. The final message table should look like this – just copy and paste it.

```
static char msgTable[] = {
```

```

"extra myFirstXtra\n" \
"new object me\n" /* standard first handler entry in all message tables */

"-- Template handlers --\n"
"* mfx_globalHandler          -- prints Hello message\n"
"+ mfx_parentHandler object xtraRef -- prints parent handler message\n"
"mfx_childHandler  object me      -- prints child handler message\n"
};

```

The addition of 'mfx_' (My First Xtra) is just to ensure that the globalhandler call doesn't conflict with any other xtras, as its unlikely anyone else would have used this naming convention. In the future you should come up with your own naming convention, for example I use NCP_handlername.

11. Open cregister.h, scroll down to find the line

```
#error PLEASE DEFINE A NEW CLSID
```

Comment this line out, open the GUID generator and get a new GUID. Copy this to below the commented line.

```

// {4CA5E661-A9F9-11d6-BFFC-000102F413ED}
DEFINE_GUID(<<name>>,
0x4ca5e661, 0xa9f9, 0x11d6, 0xbf, 0xfc, 0x0, 0x1, 0x2, 0xf4, 0x13, 0xed);

```

This is what I get from Guidgen.exe, but we don't need all of it. Delete the first line, then replace <<name>> with the Class ID of this class. In this case its CLSID(CRegister) so it now reads

```

// #error PLEASE DEFINE A NEW CLSID
DEFINE_GUID(CLSID(CRegister),0x4ca5e661, 0xa9f9, 0x11d6, 0xbf, 0xfc, 0x0, 0x1, 0x2, 0xf4,
0x13, 0xed);

```

12. Open cscript.h. scroll down and repeat step 11 for defining the GUID.

Remember you need to create another new GUID for this class, don't use the same one! However Note that the class name is now Cscript

```

// #error PLEASE DEFINE A NEW CLSID
DEFINE_GUID(CLSID(CScript),0x4ca5e662, 0xa9f9, 0x11d6, 0xbf, 0xfc, 0x0, 0x1, 0x2, 0xf4,
0x13, 0xed);

```

Then scroll down to the very end of the file, where it says 'enum' and replace the names with the updates you made to the message table in cregister.cpp, so it reads.

```

enum
{
m_new = 0, /* standard */

m_mfx_globalHandler,
m_mfx_parentHandler,
m_mfx_childHandler,

m_XXXX
};

```

13. open cscript.cpp, scroll down to
STDMETHODIMP CScript_IMoaMmXScript::Call(PMoaMmCallInfo callPtr)
And change the 'case' statement names to match the enum names

```
case m_mfx_globalHandler:  
    err = XScrpGlobalHandler(callPtr);  
    break;
```

14. Take a deep breath and select build from the Build menu (or press F7)
If everything compiles, you've just made you're first xtra, although it still includes the generic handlers from the skeleton version.

Running the Xtra

- 1 Minimise MSVC and locate the myFirstXtra.x32 file in
Xtra_development/Xtra/Skeleton/winproj/debug or release depending on your compile setting.
2. Make a shortcut to the file, and place the shortcut into your Macromedia/Director85/xtra folder.
3. Run Director

Tips:

Don't have Director open when you build a new version of an xtra. MSVC will complain because the file is already in use, and can't be overwritten.

Q.1 Why doesn't my Xtra work?

Have you placed it, or a shortcut in the Director/xtra folder?
Does it appear in the 'put the xtralist' list?

Check you're messageTable – THOROUGHLY.
Its amazing how little errors such as a missing comma between handler arguments can creep in when witing it. Doing so will cause the xtra to fail.

Q.2 Why doesn't the updates I made to my xtra appear to work?

Be sure you're using the correct version of the xtra, and the right build, its easy to get mixed up between Debug and Release builds.