

Debugging C Code (Business Functions)

By Jeff Geiger

Introduction

You can use Microsoft Visual C++ to debug business functions, including C business functions and NER business functions (by debugging the NER generated C code). Starting with OneWorld® version B7331 you must have version 6 of Microsoft Visual C++ installed on the machine you wish to use for debugging purposes.

You can debug business functions attached to interactive applications or batch applications. The business functions MUST be executed locally. If launching a UBE, this generally means running it on the local workstation.

Determine whether to build in DEBUG or OPTIMIZE mode.

- 1) Open up windows explorer, go into b7/system/bin32, and find jdbodbc.dll.
- 2) Right click on the file, choose properties, choose the versions tab, and select build type. If the window to the right shows RELEASE, build all BSFNs in optimize mode. If it shows DEBUG, build all BSFNs in debug mode. This will ensure the Business DLLs are built in the same mode as the Base DLLs. “Cross building” might cause memory violations during BSFN execution.

Note: The default build mode for BusBuild is set in the [JDE_CG] section of the jde.ini. Change TARGET = *debug mode*, where *debug mode* can be Optimize or Debug. You can change this manually, or by selecting the appropriate mode while in BusBuild (which will automatically update the jde.ini settings.)

Build the Business function.

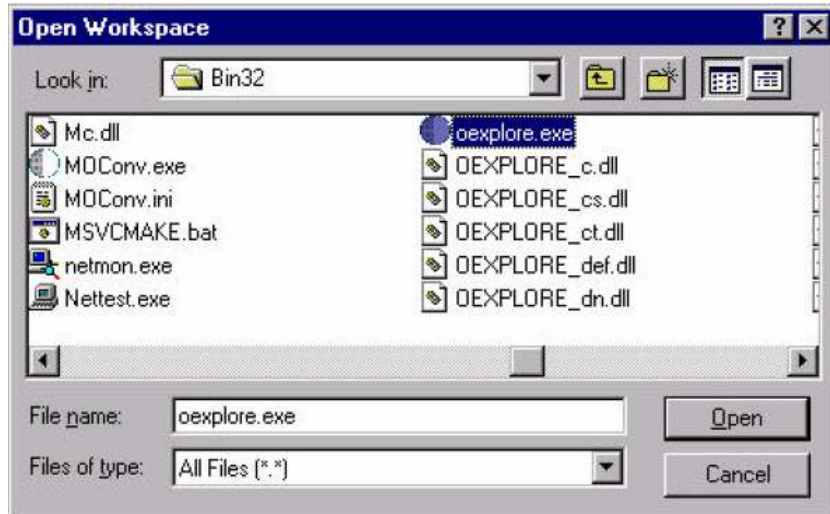
- 1) Sign on to OneWorld, go to object management workbench (OMW), find the BSFN that you want to debug and put it into your project. Highlight the object and perform a “Get” (unless the current specs are what you want to build), choose design, go to the Design Tools Tab and then click on “Build Business Function”.

(for more information on the OMW, see the eGuide – Technical – Development Tools – Object Management Workbench)

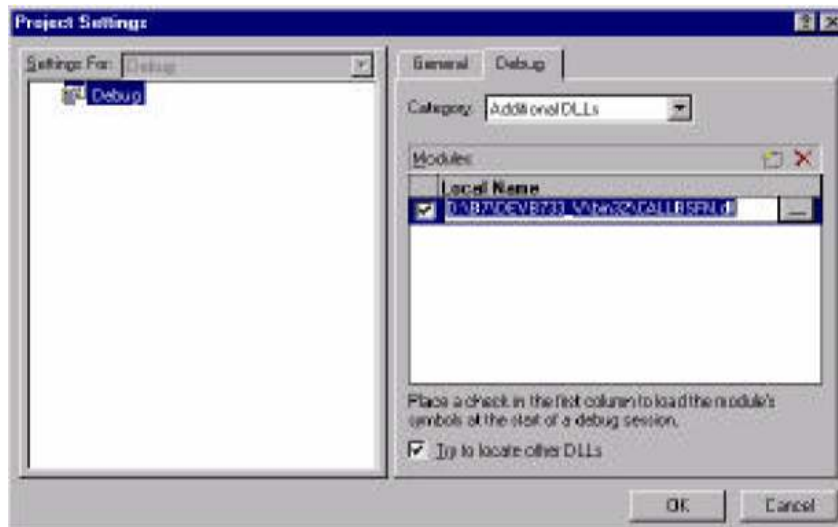
- 2) In the BusBuild window, make sure that under build tool bar the debug info has a check mark by it, and the dropdown box has the appropriate build mode of OPTIMIZE. If debug info is not checked, let OneWorld finish the build, check the debug info option, close BusBuild, and hit build again. If the dropdown box has the wrong build mode, to avoid memory violations, let OneWorld finish the build, select the correct mode in the BusBuild mode window, close BusBuild, and hit build again.
- 3) Remember the parent DLL of the BSFN.
- 4) Close out of OneWorld.

Debug the Business Function.

- 1) Make sure that OneWorld (Oexplore, UTB, BusBuild, etc.) is not running. OneWorld must be closed to correctly debug using C++.
- 2) Open Microsoft Visual C++ and make sure that all workspaces are closed.
- 3) From the file menu, choose Open Workspace.
- 4) Make sure to choose all file types so you can choose the correct executable.
- 5) Select OEXPLORE.EXE on path\b7\System\bin32.



- 6) Go to the project menu at the top of the screen.
- 7) Choose settings.
- 8) Make the debug tab active.



- 9) From the drop down box for category choose additional DLLs.
- 10) Click under local names and press the ... three dots button.
- 11) Choose b7\path code\bin32\xxxx.dll and choose the DLL where the business functions resides (parent DLL) and choose OK.
- 12) From the File menu choose open to open the .c file related to the BSFN (B7\path code\source\xxxxxx.c)
- 13) Set your break points in the code (make sure to set your break points after main processing).
- 14) Go to the build menu option then start debug and then go.
- 15) If "Microsoft Developer Studio" message boxes appear click ok and continue.
- 16) The OneWorld Login box will appear.
- 17) Sign into OneWorld.

Note: If you are debugging the application's event rule and C business functions, you can use the OneWorld debugger and visual C++ debugger together. Follow the steps above until you log into OneWorld. At this point follow the step for OneWorld debugger. Program execution stops if C code is accessed. You can then use Visual C++ to continue debugging. This is useful if you are trying to locate a problem and you are not sure whether the problem is in a C business function or the application calling the business function

Launch the application or submit the UBE (locally) as you normally would (Menu, Interactive Versions, Batch Versions, Object Librarian, etc). When the application or UBE reaches the business function to Debug, it will display the C code in Visual C++ so that you can step through it.