

NWN Script Debugger

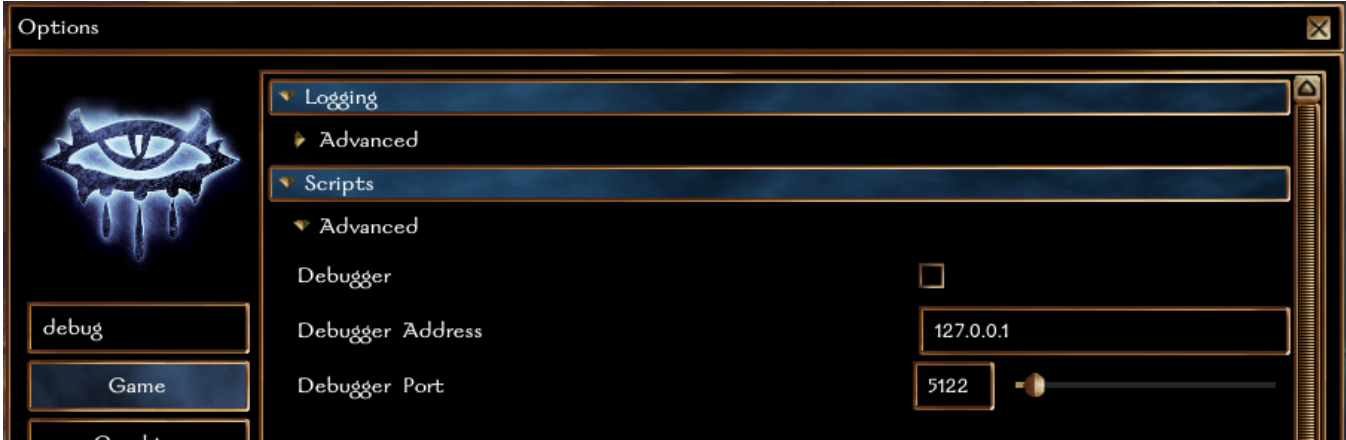
The NWN script debugger is used to...debug scripts! well it's kinda an old Windows-only utility (that runs in WINE) that can be used with SpawnScriptDebugger to see the variables a script have and step through a bit of code step by step.

Debugger

You can debug NWN nss scripts. It's sure not pretty, and for most users [PostString](#) will be a lot easier to debug things, but may be useful to some.

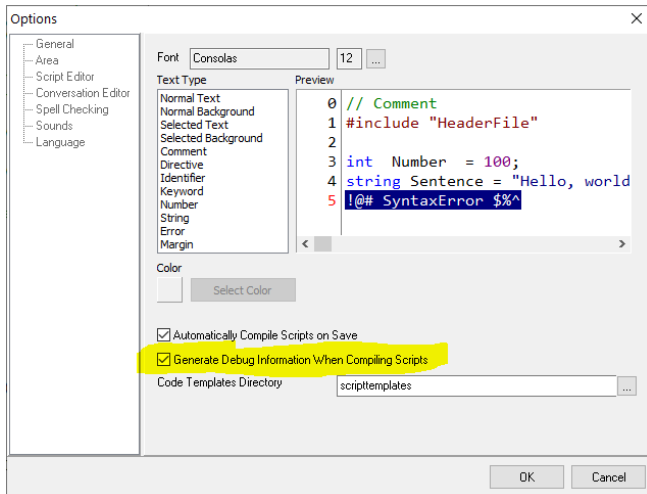
NWN:EE 8193.23 and higher now includes an updated DebugServer.exe file (which works with newer types of thing like [sqlquery](#)), but if your install missing it use [this NWVault copy](#). If you use this copy you need to create a folder called "utils" in the path just above the main game exe files. For instance: **D:\Steam\steamapps\common\Neverwinter Nights\bin\win32\utils**

Next step: go to your NWN:EE's [options](#) and enable the debugger:



You can adjust the port and address here too. Obviously we're not...in 2001 so we can use both applications on the same device so 127.0.0.1 is fine.

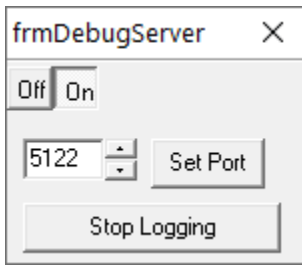
Then we need to generate the debug files. Use the script compiler referenced in this guide with the flag **-g** which is "Enable generation of .ndb debug symbols file". Or you can enable it in the toolset using this checkbox:



Once ticked it will generate **.ndb** files alongside the **.nss** versions. Both are needed.

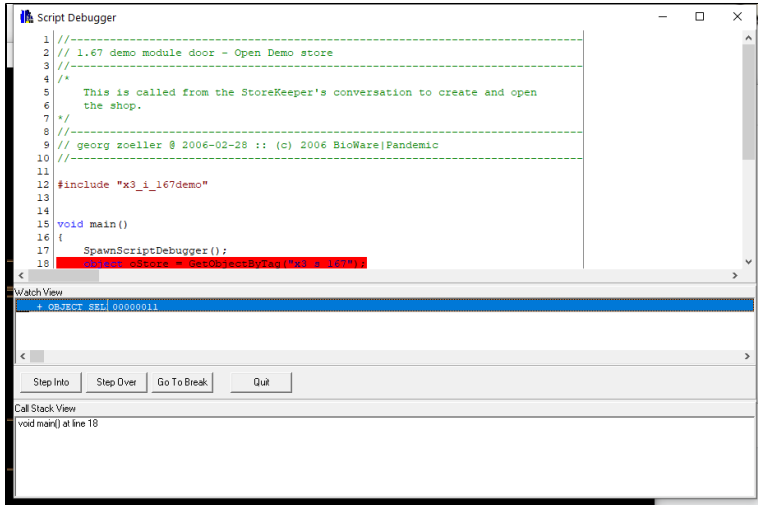
You then need a way to spawn in the debugger! This is the function [SpawnScriptDebugger](#). As noted in the description of the function do not use this needlessly or without the debugger present, it just slows things down.

Now start the debugger application, and use the port suggested above. It looks like this when opened and must be set to "On". The game *should* be able to open it by itself mind you when it hits [SpawnScriptDebugger](#).



Now run your module, from the toolset or in game, with the debugger in the background. I recommend running it in windowed mode. On 1.69 set the Fullscreen=0 and AllowWindowed=1 options. In NWN:EE you can press ALT+Enter or change it in the options.

Once it hits the given [SpawnScriptDebugger](#) start point the window should pop up with the debugger allowing you to view all the variable and step through the script.



Once closed the script will continue as usual.

API/Protocol

If you want to write your own version of DebugServer.exe it'll need the API info, which is below:

The protocol is token-based, with the following tokens:

```

#define CVMDEBUGGER_FIELD_BYTE_INTEGER      'i'
#define CVMDEBUGGER_FIELD_BYTE_STRING      's'
#define CVMDEBUGGER_FIELD_BYTE_ENDOFMESSAGE 'e'
//-----
#define CVMDEBUGGER_MAJOR_BYTE_FROM_INTERFACE 'D'
#define CVMDEBUGGER_MAJOR_BYTE_TO_INTERFACE  'G'
//-----
#define CVMDEBUGGER_MINOR_BYTE_FULL_WINDOW  '0'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_SHUTDOWN 'X'
//-----
#define CVMDEBUGGER_MINOR_BYTE_CODE_WINDOW  '1'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CLEAR_SCRIPT 'C'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_LINE_OF_SCRIPT 'L'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_DEBUGGER_FOCUS 'D'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_KEYWORD_TO_HIGHLIGHT 'K'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_NO_REDRAW 'N'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_RESUME_REDRAW 'U'
//-----
#define CVMDEBUGGER_MINOR_BYTE_CALLSTACK_WINDOW '2'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CLEAR_CALL_STACK 'C'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_LINE_OF_CALL_STACK 'L'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_FOCUS_LINE_CALL_STACK 'F'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CAPTION_CALL_STACK 'T'
//-----
#define CVMDEBUGGER_MINOR_BYTE_WATCH_WINDOW '3'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CLEAR_WATCH_WINDOW 'C'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_LINE_WATCH_WINDOW 'L'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_FOCUS_LINE_WATCH_WINDOW 'S'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_TOGGLE_LINE_WATCH_WINDOW 'F'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CAPTION_WATCH_WINDOW 'T'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CAPTION_OBJECTS 'O'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_CAPTION_VALUES 'V'
//-----
#define CVMDEBUGGER_MINOR_BYTE_BUTTON_PANEL '4'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_BUTTON_EXIT '1'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_BUTTON_STEP_INTO '2'
#define CVMDEBUGGER_COMMAND_FROM_INTERFACE_BUTTON_STEP_OVER '3'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_BUTTON_EXIT_TEXT 'E'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_BUTTON_STEP_INTO_TEXT 'I'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_BUTTON_STEP_OVER_TEXT 'O'
#define CVMDEBUGGER_COMMAND_TO_INTERFACE_BUTTON_GO_TO_BREAK 'G'

```

Integers are always 32bit in network order.

Strings are 32bit-network order size prefixed. (x02\x00x00x00hi)

Example commands as sent by the debugger UI to the game:

D2Li\x05\x00\x00\x00e Display the current display location at callstack entry/line 5. (The hex encoding needs to be sent as binary on the wire).