INSIDE THE ULTIMA ONLINE CLIENT
- ALPHA CLIENT LEFTOVERS, HAVING FUN WITH MENU'S

GOAL

I'm going to describe and try to understand some unused code found in the Ultima Online 2D clients (and even the Ultima Online Demo).

This article is similar to the "Pre-Alpha Client Leftovers, The Cursors" article but goes a few steps further. I will explain to you how to copy the menu from the Alpha Client to the Ultima Online Demo. I will also tell you how to look for secret menu options and what steps you can take to re-enable them. Have fun!

UTILITIES USED

IDA Pro, a very professional utility, definitely worth buying, Standard version is affordable
HxD, a very neat hex editor and above all, it's free
Resource Hacker, a free utility to fool around with a program's resources

INSIDE THE CLIENT

NOTE: the client analyzed here is version 5.0.8.3

The Ultima Online Client contains a call to the LoadMenu function at startup. The resource file does not contain a menu thus no menu will be displayed in the game window.

```
00536336                mov     ecx, [esp+6B4h+var_69C]
0053633A                mov     edx, GLOBAL_hInstance
00536340                push    6Ah             ; lpMenuName
00536342                push    edx             ; hInstance
00536343                lea     esi, [eax+ecx*2]
00536346                mov     eax, [edi]
00536348                add     esi, eax
0053634A                call    ds:LoadMenuA
00536350                mov     GLOBAL_hMenu, eax
```

The raises a few questions, can we add a menu and where does this menu come from in the first place?

INSIDE THE DEMO

The Ultima Online Demo contains a similar LoadMenu call:

```
004FCCDA push    6Ah ; 'j'                      ; lpMenuName
004FCCDC mov     ecx, [ebp+var_BC]
004FCCE2 mov     edx, [ecx+438h]
004FCCE8 push    edx                            ; hInstance
004FCCE9 call    ds:LoadMenuA
004FCCEF mov     ecx, [ebp+var_BC]
004FCCF5 mov     [ecx+544h], eax
```

Just like in the modern client, the menu itself is missing in the resource section of the EXE.

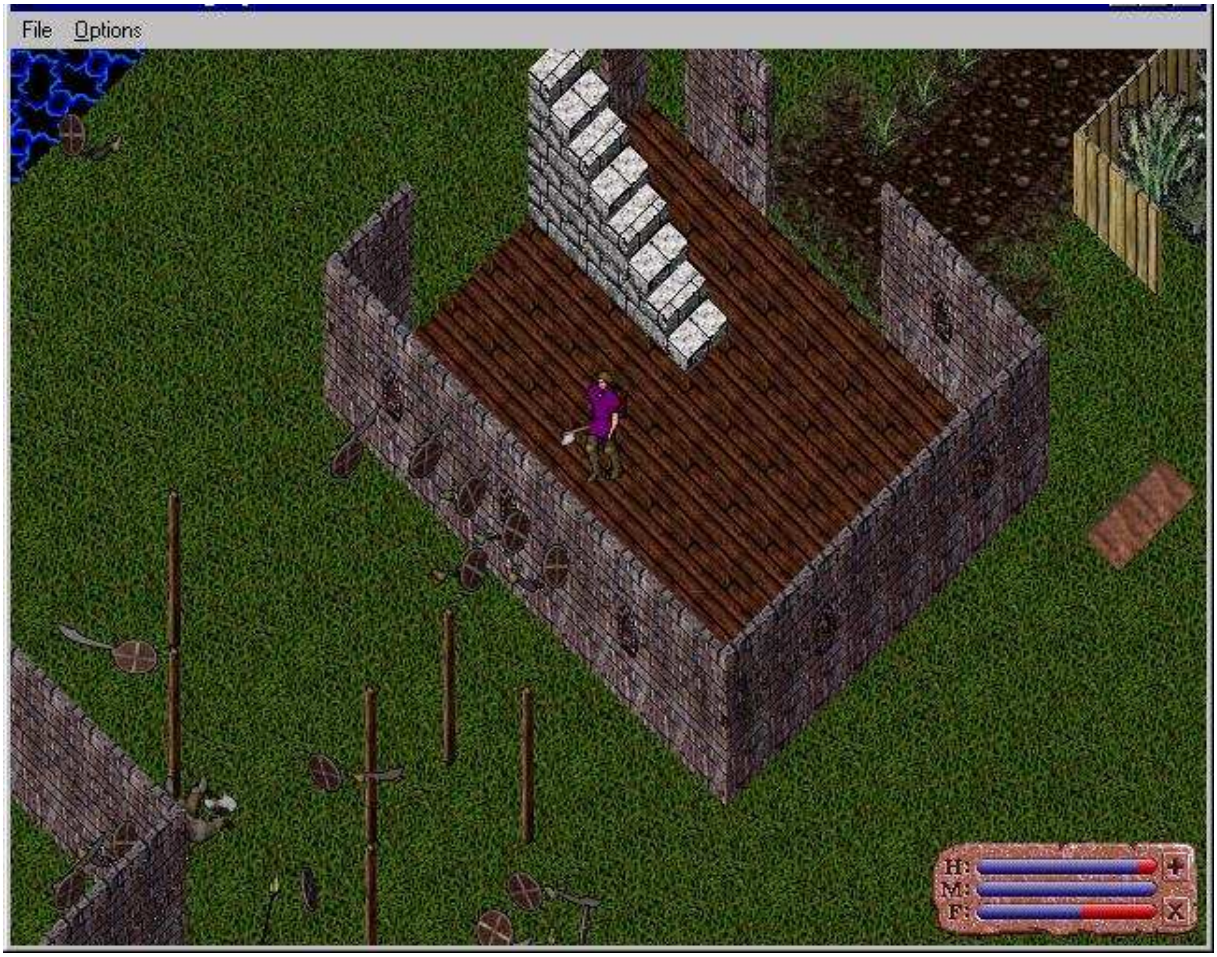
So, whatever this menu is, it must be older than 1998 and has been removed for an unknown reason.

PRE-ALPHA

Raph Koster's website, one of the original UO designers, has posted a screenshot of the original client:

http://www.raphkoster.com/2006/06/24/random-uo-anecdote-1/

As you can see, that screenshot has a menu on it:



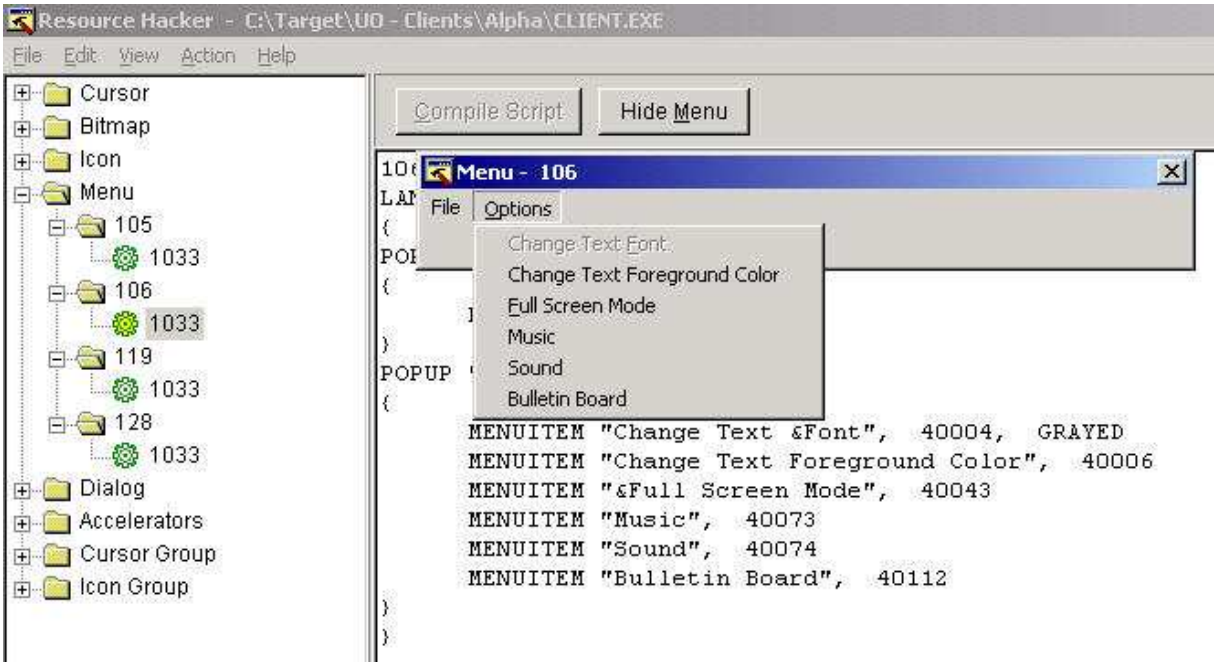
Recently, the client has become available (again) on the net, so we can delve into it and do some research.

INSIDE THE PRE-ALPHA CLIENT

As expected, the client also contains the already famous LoadMenu call:

```
00422483        push    6Ah             ; lpMenuName
00422485        mov     ebp, ds:LoadMenuA
0042248B        push    esi             ; hInstance
0042248C        call    ebp ; LoadMenuA
0042248E        push    0               ; lpParam
00422490        mov     [ebx+8AFh], eax
```

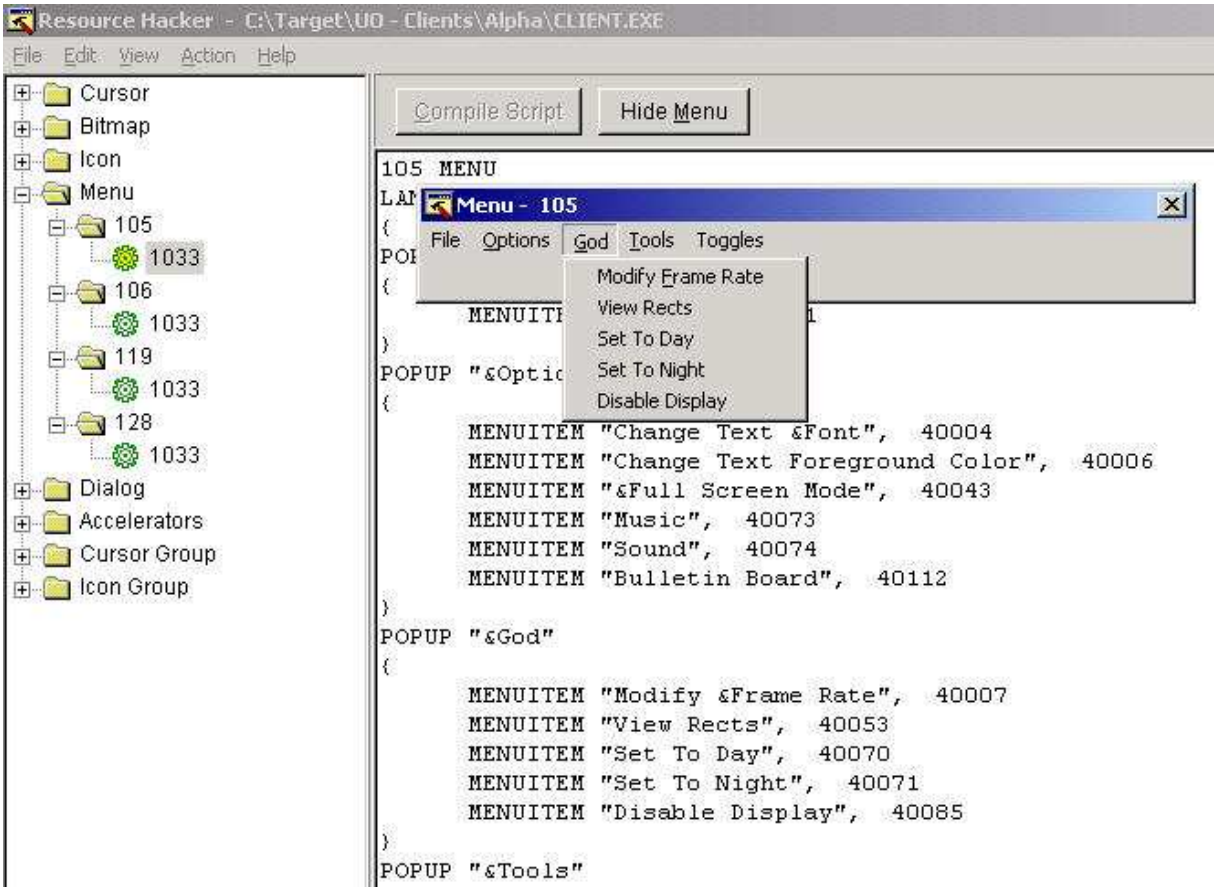
But there is a difference; the Pre-Alpha Client contains the menu lay-out as shown here (using Resource Hacker):



NOTE: 6Ah (see assembler screenshot) equals 106, 1033 is a language identifier

IS THE PRE-ALPHA CLIENT GOD?

There is another difference; the Pre-Alpha Client contains a GOD menu too, with ID 105 (69h):



Let's investigate if the Pre-Alpha Client supports (handles/can execute) this GOD menu.

When dealing with menus you need to investigate the WindowProc function and look for WM_COMMAND handling:

```
00424052 LOCAL_HandleWMCOMMAND:                  ; CODE XREF: FUNC_WindowProc+6F↑j
00424052                 mov     ebx, [esp+158h+wParam]
00424059                 mov     eax, ebx
0042405B                 and     eax, 0FFFFh
00424060                 sub     eax, 40001
00424065                 cmp     eax, 73h        ; switch 116 cases
00424068                 ja      loc_424517      ; default
00424068                                         ; jumptable 00424076 cases 1,2,4,
0042406E                 xor     ecx, ecx
00424070                 mov     cl, ds:byte_42458C[eax]
00424076                 jmp     ds:off_424560[ecx*4] ; switch jump
```

A jump-table is used (see the previous screenshot), so we need to look at this table:

```
00424560 off_424560      dd offset loc_42407D, offset loc_424098, offset loc_4240A9
00424560                                       ; DATA XREF: FUNC_WindowProc+466↑r
00424560                 dd offset loc_4240DA, offset loc_424236, offset loc_424292 ;
00424560                 dd offset loc_4240DA, offset loc_4242EE, offset loc_42431E
00424560                 dd offset loc_4243AA, offset LOCAL_HandleWMCOMMAND_DoNothing
0042458C byte_42458C     db      0,    0Ah,    0Ah,     1
0042458C                                       ; DATA XREF: FUNC_WindowProc+460↑r
0042458C                 db    0Ah,      2,    0Ah,    0Ah ; indirect table for switch st
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
0042458C                 db    0Ah,    0Ah,      3,    0Ah
0042458C                 db    0Ah,    0Ah,    0Ah,    0Ah
```

Most resource ID's will default to the eleventh (0Ah+ 1) handler, this handler will jump directly to DefWindowProc. This means nothing else happens for those menu options, they are unhandled!

This is a screenshot of the handler going to DefWindowProc:

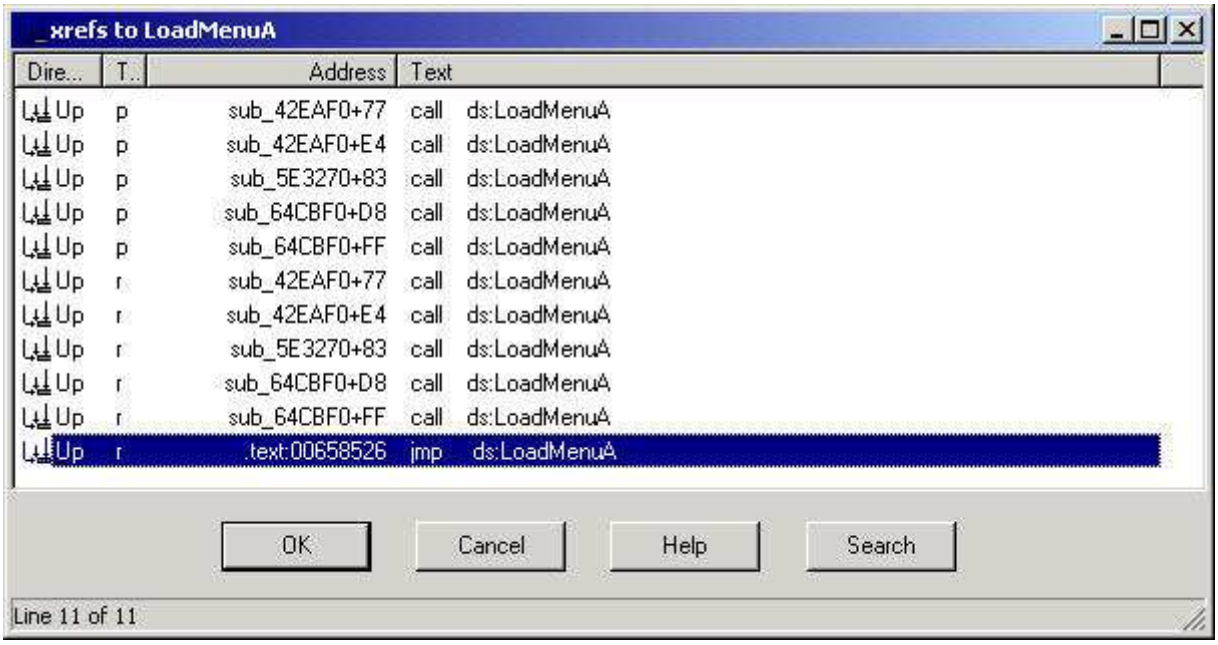
```
00424517 LOCAL_HandleWMCOMMAND_DoNothing:        ; CODE XREF: FUNC_WindowProc+458↑j
00424517                                         ; FUNC_WindowProc+466↑j
00424517                                         ; DATA XREF: ...
00424517                 mov     esi, [esp+158h+hWnd] ; default
00424517                                         ; jumptable 00424076 cases 1,2,4,6-41,
0042451E                 jmp     short LOCAL_GoDoDefWindowProc
00424520 ; ---------------------------------------------------------------------------
00424520
00424520 loc_424520:                             ; CODE XREF: FUNC_WindowProc+3E2↑j
00424520                 mov     esi, [esp+158h+hWnd]
00424527                 jmp     short loc_424530
00424529 ; ---------------------------------------------------------------------------
00424529
00424529 LOCAL_GoDoDefWindowProc:                ; CODE XREF: FUNC_WindowProc+35↑j
00424529                                         ; FUNC_WindowProc+327↑j ...
00424529                 mov     ebp, [esp+158h+lParam]
00424530
00424530 loc_424530:                             ; CODE XREF: FUNC_WindowProc+8E2↑j
00424530                                         ; FUNC_WindowProc+905↑j ...
00424530                 push    ebp             ; lParam
00424531                 push    ebx             ; wParam
00424532                 push    edi             ; Msg
00424533                 push    esi             ; hWnd
00424534                 call    ds:DefWindowProcA
```

ANSWER

The Pre-Alpha Client is not GOD but the available GOD menu at least gives us insight in what the Pre-Alpha GOD client was capable of during the early development in 1996.

INSIDE THE GOD CLIENT

The leaked GOD client from the year 2000 also contains multiple menus and a few LoadMenu calls:



Two menus are the same as we saw in the Pre-Alpha Client, namely menu 69h and 6Ah. I'll provide two screenshots of the actual code loading them.

69h:

```
0042EB5E                push    69h             ; lpMenuName
0042EB60                mov     edx, GLOBAL_hInstance
0042EB66                push    edx             ; hInstance
0042EB67                call    ds:LoadMenuA
0042EB6D                mov     [ebp+hMenu], eax
```

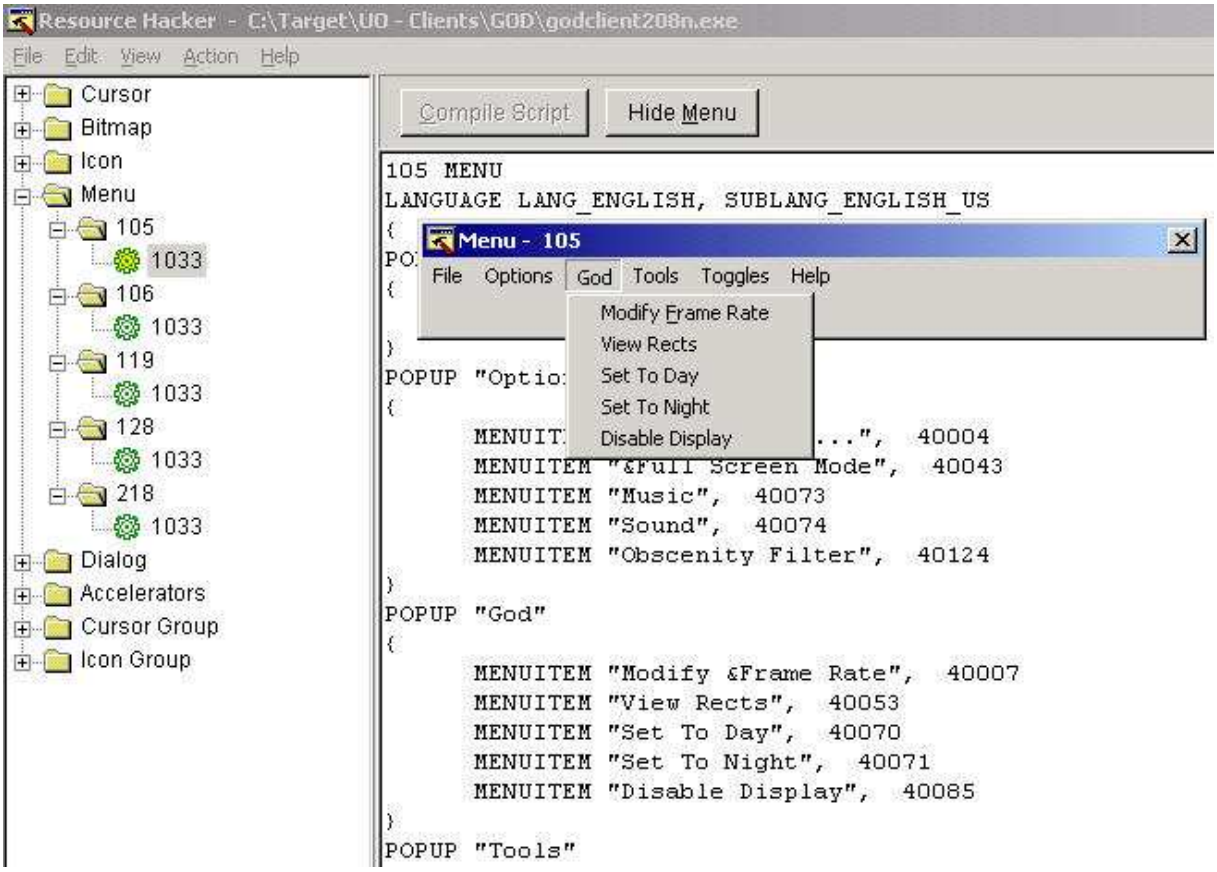
6Ah:

```
0042EBCB                push    6Ah             ; lpMenuName
0042EBCD                mov     edx, GLOBAL_hInstance
0042EBD3                push    edx             ; hInstance
0042EBD4                call    ds:LoadMenuA
0042EBDA                mov     [ebp+hMenu], eax
```

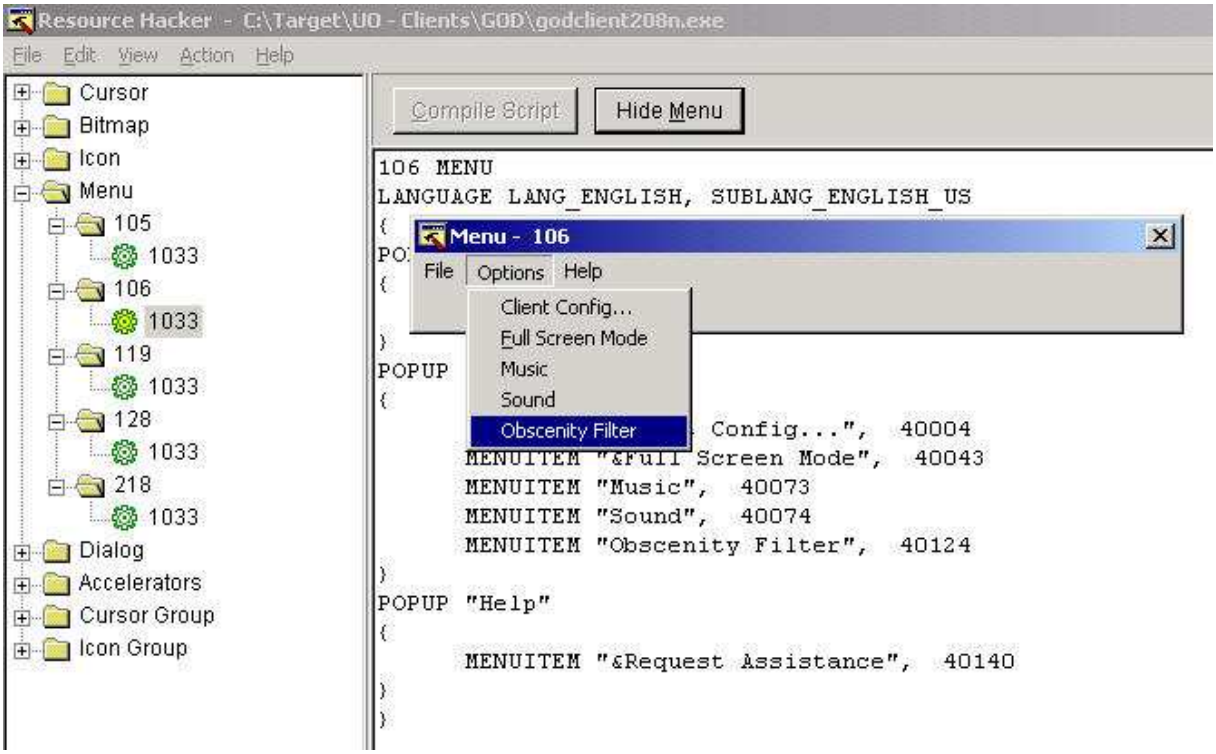
It's logical that a GOD client would load the GOD menu (69h). It's interesting to note that the menu identifiers from 1996 are still used in the year 2000. This means the client is clearly a continuation of the original pre-alpha client.

Screenshots of the GOD menu and the Normal menu using Resource Hacker:

69h:



6A:



LOADMENU (GOD MODE ON/OFF)

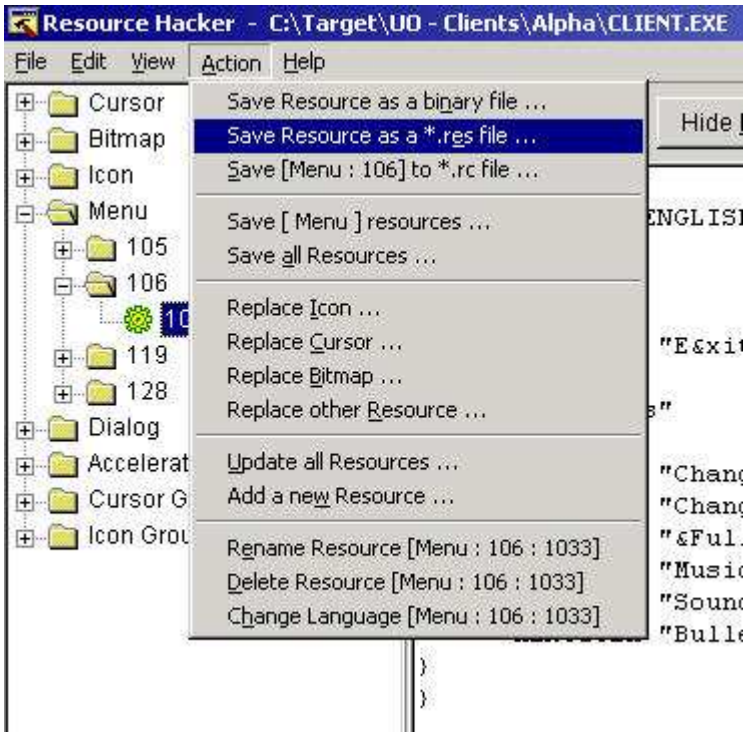
```
0042EB21                mov     ecx, [ebp+VAR_FlagGOD]
0042EB24                and     ecx, 0FFh
0042EB2A                test    ecx, ecx
0042EB2C                jz      LOCAL_GODoff
0042EB32
0042EB32 LOCAL_GODon:
0042EB32                cmp     dword_85D6E4, 0
0042EB39                jz      short loc_42EB5E
0042EB3B                mov     dword_85D6E4, 0
0042EB45                cmp     dword_B52140, 0
0042EB4C                jz      short loc_42EB59
0042EB4E                mov     ecx, dword_B52140
0042EB54                call    sub_404F11
0042EB59
0042EB59 loc_42EB59:                            ; CODE XREF: sub_42EAF0+5C↑j
0042EB59                call    sub_403E72
0042EB5E
0042EB5E loc_42EB5E:                            ; CODE XREF: sub_42EAF0+49↑j
0042EB5E                push    69h             ; lpMenuName
0042EB60                mov     edx, GLOBAL_hInstance
0042EB66                push    edx             ; hInstance
0042EB67                call    ds:LoadMenuA
0042EB6D                mov     [ebp+VAR_hMenuLoaded], eax
0042EB70                cmp     dword_18A3FB4, 0
0042EB77                jnz     short loc_42EB8D
0042EB79                mov     eax, [ebp+VAR_hMenuLoaded]
0042EB7C                push    eax             ; hMenu
0042EB7D                mov     ecx, dword_18A4B84
0042EB83                mov     edx, [ecx+4]
0042EB86                push    edx             ; hWnd
0042EB87                call    ds:SetMenu
0042EB8D
0042EB8D loc_42EB8D:                            ; CODE XREF: sub_42EAF0+87↑j
0042EB8D                cmp     GLOBAL_hMenu, 0
0042EB94                jz      short LOCAL_DoNotDestroyMenu69
0042EB96                mov     eax, GLOBAL_hMenu
0042EB9B                push    eax             ; hMenu
0042EB9C                call    ds:DestroyMenu
0042EBA2
0042EBA2 LOCAL_DoNotDestroyMenu69:              ; CODE XREF: sub_42EAF0+A4↑j
0042EBA2                mov     ecx, [ebp+VAR_hMenuLoaded]
0042EBA5                mov     GLOBAL_hMenu, ecx
0042EBAB                push    offset aGodModeIsNowOn ; "God mode is now on."
0042EBB0                push    3
0042EBB2                push    0
0042EBB4                call    sub_409B3D
0042EBB9                add     esp, 0Ch
0042EBBC                call    sub_408E13
0042EBC1                call    sub_4019E2
0042EBC6                jmp     LOCAL_GoReturn
0042EBCB ; ---------------------------------------------------------------------------
0042EBCB
0042EBCB LOCAL_GODoff:                          ; CODE XREF: sub_42EAF0+3C↑j
0042EBCB                push    6Ah             ; lpMenuName
0042EBCD                mov     edx, GLOBAL_hInstance
0042EBD3                push    edx             ; hInstance
0042EBD4                call    ds:LoadMenuA
0042EBDA                mov     [ebp+VAR_hMenuLoaded], eax
0042EBDD                cmp     dword_18A3FB4, 0
0042EBE4                jnz     short loc_42EBFA
0042EBE6                mov     eax, [ebp+VAR_hMenuLoaded]
0042EBE9                push    eax             ; hMenu
0042EBEA                mov     ecx, dword_18A4B84
0042EBF0                mov     edx, [ecx+4]
0042EBF3                push    edx             ; hWnd
0042EBF4                call    ds:SetMenu
0042EBFA
0042EBFA loc_42EBFA:                            ; CODE XREF: sub_42EAF0+F4↑j
0042EBFA                cmp     GLOBAL_hMenu, 0
0042EC01                jz      short LOCAL_DoNotDestroyMenu6A
0042EC03                mov     eax, GLOBAL_hMenu
0042EC08                push    eax             ; hMenu
0042EC09                call    ds:DestroyMenu
0042EC0F
0042EC0F LOCAL_DoNotDestroyMenu6A:              ; CODE XREF: sub_42EAF0+111↑j
0042EC0F                mov     ecx, [ebp+VAR_hMenuLoaded]
0042EC12                mov     GLOBAL_hMenu, ecx
0042EC18                push    offset aGodModeIsNowOf ; "God mode is now off."
0042EC1D                push    3
0042EC1F                push    0
0042EC21                call    sub_409B3D
0042EC26                add     esp, 0Ch
0042EC29                mov     [ebp+var_10], 1
0042EC30                lea     edx, [ebp+var_10]
0042EC33                push    edx
0042EC34                push    offset dword_141BB78
0042EC39                call    sub_404854
0042EC3E                add     esp, 8
0042EC41                and     eax, 0FFh
0042EC46                test    eax, eax
0042EC48                jz      short LOCAL_GoReturn
0042EC4A                call    sub_4073BF
0042EC4F
0042EC4F LOCAL_GoReturn:                        ; CODE XREF: sub_42EAF0+D6↑j
0042EC4F                                        ; sub_42EAF0+158↑j
```

MENU FUN

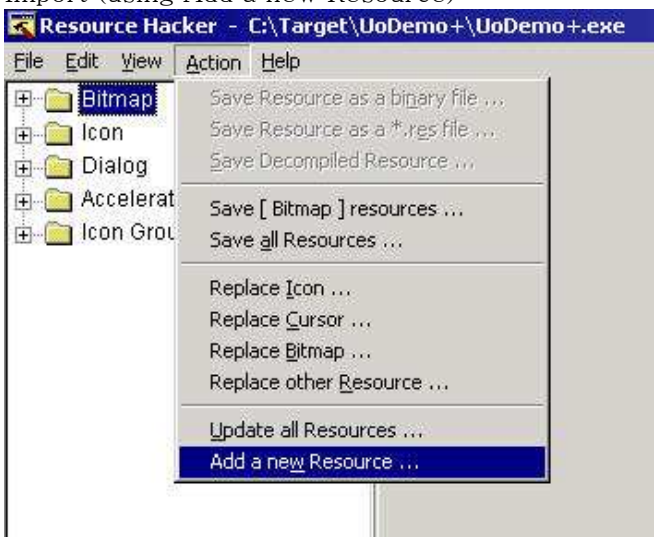
Since all clients we investigated so far (5.0.8.3 / Demo / Pre-Alpha / GOD) support the menu 6Ah (106) we will now investigate if the support is actually implemented beyond the LoadMenu call.

To do so, open Resource Hacker, save the menu (from either the Pre-Alpha or GOD client) to a RES file and import it again in to the target client (5.0.8.3 or the Demo or any other client you desire to test for menu support).

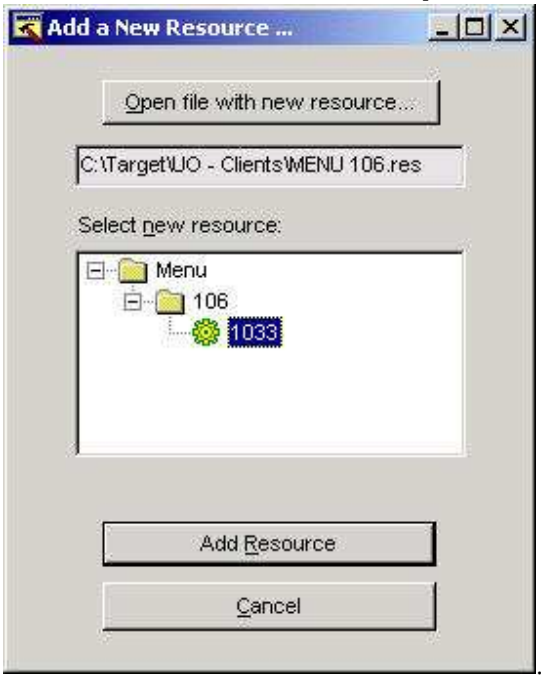
1. Open Resource Hacker
2. Open your Source Client
3. Go To Menu 106 (1033)
4. Save as a RES file:



5. Open the Target Client
6. Import (using Add a new Resource):

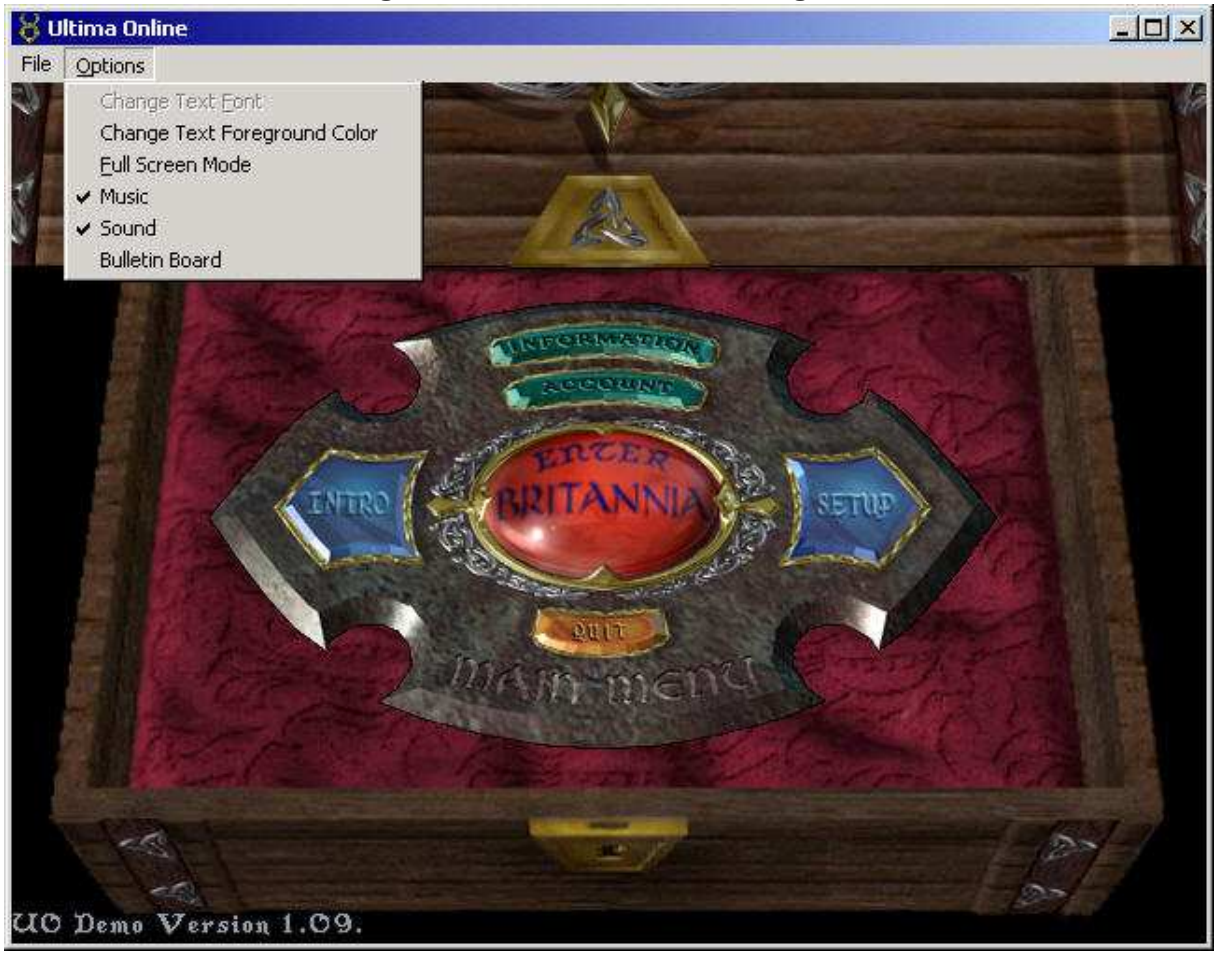


7. Select - Menu 106 (1033) – and press Add Resource:



8. When done, save the executable (make sure you have backup somewhere).

9. Do a test run; make sure the game is in Windowed Mode though



On the previous screenshot we can see that the demo will load the menu from the Pre-Alpha Client, if you would repeat the steps and actually play with the menu you will find that the demo (and client 5.0.8.3) is reacting to the menu options.

Conclusion: the game is responding to a menu that is initially not available. This makes me angry! This means there is some secret code inside the client, bad programming that is!

UNCOVERING SOME SECRETS

According to the Windows API programs have to respond to menu options through the WM_COMMAND handler inside a WindowProc. The Ultima Online client has only one WindowProc function so locating WM_COMMAND becomes easy.

I've done the research for you and the demo will arrive here when responding to a menu option:

```
004FDBFA loc_4FDBFA:                                   ; CODE XREF: FUNC_WindowProc+5D5↑j
004FDBFA mov     ecx, [ebp+var_60]
004FDBFD sub     ecx, 40004
004FDC03 mov     [ebp+var_60], ecx
004FDC06 cmp     [ebp+var_60], 139
004FDC0D ja      loc_4FDDF8
004FDC13 mov     eax, [ebp+var_60]
004FDC16 xor     edx, edx
004FDC18 mov     dl, ds:byte_4FE31F[eax]
004FDC1E jmp     ds:off_4FE2EF[edx*4]
```

The menu is handled in the original C language with a switch/case statement. The compiler translated it to a jump-table. The jump-table for the demo is here:

```
004FE2EF off_4FE2EF dd offset loc_4FDCCE         ; DATA XREF: FUNC_WindowProc+613↑r
004FE2F3 dd offset loc_4FDDB4
004FE2F7 dd offset loc_4FDC3C
004FE2FB dd offset loc_4FDC85
004FE2FF dd offset loc_4FDD4A
004FE303 dd offset loc_4FDD9F
004FE307 dd offset loc_4FDD7A
004FE30B dd offset loc_4FDCF3
004FE30F dd offset loc_4FDCCE
004FE313 dd offset loc_4FDD3E
004FE317 dd offset loc_4FDD62
004FE31B dd offset loc_4FDDF8
004FE31F byte_4FE31F db 0,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F                                         ; DATA XREF: FUNC_WindowProc+60D↑r
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,1,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,2,3
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,4,5,0Bh,0Bh,0Bh,6,0Bh,0Bh,0Bh,7,8,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,9,0Bh,0Bh
004FE3AA db  0Ah
```

This is similar to the jump-table we saw earlier of the Pre-Alpha Client, one option has been added. Most GOD commands are unhandled (0Bh) so the Ultima Online Demo does not contain the GOD client in it. Note: The server side contains GOD code but that's another story for another day.

If you compare the menu of the Demo with the menu of the Pre-Alpha Client you will notice they are almost equal in design:

Pre-Alpha:

```
40001 - 0 - Exit
40004 - 1 - Change Text Font
40006 - 2 - Change Text Foreground Color
40043 - 3 - Full Screen Mode
40073 - 4 - Toggle Music
40074 - 5 - Toggle Sound
40089 - 6 - Unknown
40112 - 7 - Bulletin Board
40115 - 8 - Unknown
40116 - 9 - Unknown
```

Demo:

```
40001 - / - Exit
40004 - 0 - Open Client Config
40043 - 1 - Full Screen Mode
40073 - 2 - Toggle Music
40074 - 3 - Toggle Sound
40115 - 4 - Unknown
40116 - 5 - Unknown
40120 - 6 - Unknown
40124 - 7 - Unknown
40125 - 8 - Unknown
40140 - 9 - Unknown
40143 - A - Unknown
```

Menu ID 40043 was changed from “Change Text Font” to “Open Client Config” in the demo. This is also seen in the GOD client’s menu definition. The Bulletin Board has been removed. Menu option 40089 seems to be unique for the Pre-Alpha Client but the menu is not defined in that client’s resource, so it’s a hidden menu option! Menu 40115 and 40116 are also not defined in both the demo and the Pre-Alpha Client (and the GOD client). Secrets!

Menu ID 40124 also appears in the GOD client and equals “Obscenity Filter” and menu ID 40140 equals “Request Assistance”. For the understanding the others more hacking is required.

An easy way to enable these options is to edit the menu with the Resource Hacker and add them.

MENU EDITING

Edit the menu so it looks like this:

Compile Script | Hide Menu

```
106 MENU
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
{
POPUP "File"
{
      MENUITEM "E&xit",  40001
}
POPUP "&Options"
{
      MENUITEM "Client Config...",  40004
      MENUITEM "&Full Screen Mode",  40043
      MENUITEM "Music",  40073
      MENUITEM "Sound",  40074
}
POPUP "&Hacker"
{
      MENUITEM "Unknown 40115",  40115
      MENUITEM "Unknown 40116",  40116
      MENUITEM "Unknown 40120",  40120
      MENUITEM "Unknown 40125",  40125
      MENUITEM "Unknown 40143",  40143
}
}
```

When you're done, press the Compile Script button and save the executable, then run it. If all went okay your game will look like this:

It now becomes a process of trial and error, press the menu options and see what happens.

The menu design became this:

Pre-Alpha:
```
40001 - 0 - Exit
40004 - 1 - Change Text Font
40006 - 2 - Change Text Foreground Color
40043 - 3 - Full Screen Mode
40073 - 4 - Toggle Music
40074 - 5 - Toggle Sound
40089 - 6 - Unknown
40112 - 7 - Bulletin Board
40115 - 8 - Toggle Combat
40116 - 9 - Open Spellbook
```

Demo:
```
40001 - / - Exit
40004 - 0 - Open Client Config (see 8)
40043 - 1 - Full Screen Mode
40073 - 2 - Toggle Music
40074 - 3 - Toggle Sound
40115 - 4 - Toggle Combat
40116 - 5 - Open Spellbook
40120 - 6 - Open Radar
40124 - 7 - Obscenity Filter
40125 - 8 - Open Client Config (see 0)
40140 - 9 - Request Assistance
40143 - A - Toggle Footsteps
```

Menu ID 40116 was a bit trickier for me to understand; this turned out to be the Open Spellbook packet (thanks Garret & Derrick for aiding me here).

A screenshot of the same uodemo jump-table with all menu handlers renamed to something more meaningful based on the done analysis:

```
004FE2EF off_4FE2EF dd offset LOCAL_Menu_OpenSettings
004FE2EF                                        ; DATA XREF: FUNC_WindowProc+613↑r
004FE2F3 dd offset LOCAL_Menu_ToggleFullScreen
004FE2F7 dd offset LOCAL_Menu_ToggleMusic
004FE2FB dd offset LOCAL_Menu_ToggleSound
004FE2FF dd offset LOCAL_Menu_ToggleCombat
004FE303 dd offset LOCAL_Menu_SendUnknownPacket12type43
004FE307 dd offset LOCAL_Menu_OpenRadar
004FE30B dd offset LOCAL_Menu_ObsenityFilter
004FE30F dd offset LOCAL_Menu_OpenSettings
004FE313 dd offset LOCAL_Menu_RequestAssistence
004FE317 dd offset LOCAL_Menu_ToggleFootsteps
004FE31B dd offset LOCAL_Menu_Ignore
004FE31F byte_4FE31F db 0,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F                                        ; DATA XREF: FUNC_WindowProc+60D↑r
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,1,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,2,3
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,4,5,0Bh,0Bh,0Bh,6,0Bh,0Bh,0Bh,7,8,0Bh,0Bh,0Bh
004FE31F db 0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,0Bh,9,0Bh,0Bh
004FE3AA db  0Ah
```

MENU COMPARISON

Client 5.0.8.3 (2007) is still responding to the menu options similar to the ones found in the Ultima Online Demo from 1998. This is very interesting, especially since both clients come without any menu.

I made a comparison of the 3 client menus (GOD client excluded).

Pre-Alpha:
(the original menu, with 3 secrets and 1 item disabled)

```
40001 - 0 - Exit
40004 - 1 - Change Text Font
40006 - 2 - change Text Foreground Color
40043 - 3 - Full Screen Mode
40073 - 4 - Toggle Music
40074 - 5 - Toggle Sound
40089 - 6 - Unknown
40112 - 7 - Bulletin Board
40115 - 8 - Toggle Combat
40116 - 9 - Open spellbook
```

Demo:
(new item but their ID's indicate they were added soon after Pre-Alpha)

```
40001 - / - Exit
40004 - 0 - Open Client Config (see 8)
40043 - 1 - Full Screen Mode
40073 - 2 - Toggle Music
40074 - 3 - Toggle Sound
40115 - 4 - Toggle Combat
40116 - 5 - Open Spellbook
40120 - 6 - Open Radar
40124 - 7 - Obscenity Filter
40125 - 8 - Open Client Config (see 0)
40140 - 9 - Request Assistance
40143 - A - Toggle Footsteps
```

Client 5.0.8.3:
(almost equal to the demo but with 40087 (re-?)added, why?)

```
40001 - 0 - Exit
40004 - 1 - Open Client Config (see between 8-9)
40043 - 2 - Full Screen Mode (see between 4-5)
40073 - 3 - Toggle Music
40074 - 4 - Toggle Sound
40087 - 2 - Full Screen Mode (see 2)
40115 - 5 - Toggle Combat
40116 - 6 - Open Spellbook
40120 - 7 - Open Radar
40124 - 8 - Obscenity Filter
40125 - 1 - Open Client Config (see 1)
40140 - 9 - Request Assistance
40143 - A - Toggle Footsteps
```

THE FINAL MENU

The following menu definition is the final menu for the Ultima Online Demo client (also for client 5.0.8.3 and most likely, for the many other clients out there). I renamed most options and reorganized the layout:

```
106 MENU
LANGUAGE LANG_ENGLISH, SUBLANG_ENGLISH_US
{
POPUP "&Game"
{
      MENUITEM "Toggle &Combat Mode",  40115
      MENUITEM SEPARATOR
      MENUITEM "Open Spell&book",  40116
      MENUITEM "Open &Radar",  40120
      MENUITEM SEPARATOR
      MENUITEM "E&xit",  40001
}
POPUP "&Options"
{
      MENUITEM "&Full Screen Mode",  40043
      MENUITEM SEPARATOR
      MENUITEM "Footste&ps",  40143
      MENUITEM "&Music",  40073
      MENUITEM "&Sound",  40074
      MENUITEM SEPARATOR
      MENUITEM "&Obscenity Filter",  40120
      MENUITEM SEPARATOR
      MENUITEM "Client &Config",  40004
}
POPUP "&Help"
{
      MENUITEM "Request &Assistance",  40140
}
}
```

FIXING THE DEMO

When you go in full screen mode with the menu enabled you will see the menu is active in full screen mode and when you return to windowed mode the menu is gone (forever).

Let's fix this behavior and turn it around; going in full screen mode will make the menu go away and when you enter windowed mode it will return.

Locate the code where switching between full screen and windowed mode is handled. Because we have renamed the menu handlers this becomes an easy task:

```
004FDDB4 LOCAL_Menu_ToggleFullScreen:            ; CODE XREF: FUNC_WindowProc+A2↑j
004FDDB4                                         ; FUNC_WindowProc+13E↑j ...
004FDDB4 cmp     dword_82246C, 0
004FDDBB jnz     short loc_4FDDD8
004FDDBD mov     dword_7022A8, 1
004FDDC7 call    sub_4FD41E
004FDDCC mov     dword_7022A8, 0
004FDDD6 jmp     short loc_4FDDF1
004FDDD8 ; ---------------------------------------------------------------------------
004FDDD8
004FDDD8 loc_4FDDD8:                             ; CODE XREF: FUNC_WindowProc+7B0↑j
004FDDD8 mov     dword_7022A8, 1
004FDDE2 call    sub_4FD4E7
004FDDE7 mov     dword_7022A8, 0
004FDDF1
004FDDF1 loc_4FDDF1:                             ; CODE XREF: FUNC_WindowProc+7CB↑j
004FDDF1 xor     eax, eax
004FDDF3 jmp     loc_4FE2BC
```

The trick is to understand the function and modify its design. This is not an easy task and probably takes some practice if you are new to this.

On the following pages I will give screenshots of the original code and the code I changed. Compare and learn from it.

This is not an assembler tutorial nor a patching tutorial so I will not go in too much detail. Let the code speak for itself.

FULL SCREEN: ON – ORIGINAL CODE

```
004FD439 6A F0              push    0FFFFFFF0h                      ; nIndex
004FD43B A1 18 20 70 00     mov     eax, GLOBAL_ClientWindow
004FD440 50                 push    eax                             ; hWnd
004FD441 FF 15 0C 58 9A 00  call    ds:GetWindowLongA
004FD447 89 45 EC           mov     [ebp+dwNewLong], eax
004FD44A 8B 4D EC           mov     ecx, [ebp+dwNewLong]
004FD44D 81 E1 FF FF 39 FF  and     ecx, 0FF39FFFFh
004FD453 89 4D EC           mov     [ebp+dwNewLong], ecx
004FD456 8B 55 EC           mov     edx, [ebp+dwNewLong]
004FD459 81 CA 00 00 00 80  or      edx, 80000000h
004FD45F 89 55 EC           mov     [ebp+dwNewLong], edx
004FD462 8B 45 EC           mov     eax, [ebp+dwNewLong]
004FD465 50                 push    eax                             ; dwNewLong
004FD466 6A F0              push    0FFFFFFF0h                      ; nIndex
004FD468 8B 0D 18 20 70 00  mov     ecx, GLOBAL_ClientWindow
004FD46E 51                 push    ecx                             ; hWnd
004FD46F FF 15 5C 57 9A 00  call    ds:SetWindowLongA
004FD475 C7 45 F4 00 00 00+mov      [ebp+Rect.top], 0
004FD47C C7 45 F0 00 00 00+mov      [ebp+Rect.left], 0
004FD483 C7 45 FC E0 01 00+mov      [ebp+Rect.bottom], 1E0h
004FD48A C7 45 F8 80 02 00+mov      [ebp+Rect.right], 280h
004FD491 6A EC              push    0FFFFFFECh                      ; nIndex
004FD493 8B 15 18 20 70 00  mov     edx, GLOBAL_ClientWindow
004FD499 52                 push    edx                             ; hWnd
004FD49A FF 15 0C 58 9A 00  call    ds:GetWindowLongA
004FD4A0 50                 push    eax                             ; dwExStyle
004FD4A1 A1 18 20 70 00     mov     eax, GLOBAL_ClientWindow
004FD4A6 50                 push    eax                             ; hWnd
004FD4A7 FF 15 C0 57 9A 00  call    ds:GetMenu
004FD4AD F7 D8              neg     eax
004FD4AF 1B C0              sbb     eax, eax
004FD4B1 F7 D8              neg     eax
004FD4B3 50                 push    eax                             ; bMenu
004FD4B4 6A F0              push    0FFFFFFF0h                      ; nIndex
004FD4B6 8B 0D 18 20 70 00  mov     ecx, GLOBAL_ClientWindow
004FD4BC 51                 push    ecx                             ; hWnd
004FD4BD FF 15 0C 58 9A 00  call    ds:GetWindowLongA
004FD4C3 50                 push    eax                             ; dwStyle
004FD4C4 8D 55 F0           lea     edx, [ebp+Rect]
004FD4BD FF 15 0C 58 9A 00  call    ds:GetWindowLongA
004FD4C3 50                 push    eax                             ; dwStyle
004FD4C4 8D 55 F0           lea     edx, [ebp+Rect]
004FD4C7 52                 push    edx                             ; lpRect
004FD4C8 FF 15 58 57 9A 00  call    ds:AdjustWindowRectEx
```

FULL SCREEN: ON –CHANGED CODE

```
004FD4A1 A1 18 20 70 00     mov     eax, GLOBAL_ClientWindow
004FD4A6 6A 00              push    0                               ; hMenu
004FD4A8 50                 push    eax                             ; hWnd
004FD4A9 FF 15 24 58 9A 00  call    ds:SetMenu
004FD4AF 90                 nop
004FD4B0 90                 nop
004FD4B1 90                 nop
004FD4B2 6A 00              push    0                               ; bMenu
```

FULL SCREEN: OFF – ORIGINAL CODE

```
004FD5D0 8B 4D EC          mov     ecx, [ebp+cy]
004FD5D3 51                push    ecx                             ; cy
004FD5D4 8B 55 E4          mov     edx, [ebp+var_1C]
004FD5D7 52                push    edx                             ; cx
004FD5D8 6A 00             push    0                               ; Y
004FD5DA 6A 00             push    0                               ; X
004FD5DC 6A FE             push    0FFFFFFFEh                      ; hWndInsertAfter
004FD5DE A1 18 20 70 00    mov     eax, GLOBAL_ClientWindow
004FD5E3 50                push    eax                             ; hWnd
004FD5E4 FF 15 20 58 9A 00 call    ds:SetWindowPos
004FD5EA 6A 00             push    0                               ; bErase
004FD5EC 6A 00             push    0                               ; lpRect
004FD5EE 8B 0D 18 20 70 00 mov     ecx, GLOBAL_ClientWindow
004FD5F4 51                push    ecx                             ; hWnd
004FD5F5 FF 15 1C 58 9A 00 call    ds:InvalidateRect
004FD538 51                push    ecx                             ; hWnd
004FD539 FF 15 5C 57 9A 00 call    ds:SetWindowLongA
004FD53F 6A 20             push    20h ; ' '                       ; nIndex
004FD541 FF 15 20 57 9A 00 call    ds:GetSystemMetrics
004FD547 8D 94 00 80 02 00+lea     edx, [eax+eax+280h]
004FD54E 89 55 E4          mov     [ebp+var_1C], edx
004FD551 6A 21             push    21h ; '!'                       ; nIndex
004FD553 FF 15 20 57 9A 00 call    ds:GetSystemMetrics
004FD559 8B F0             mov     esi, eax
004FD55B 6A 04             push    4                               ; nIndex
004FD55D FF 15 20 57 9A 00 call    ds:GetSystemMetrics
004FD563 8D 84 70 E0 01 00+lea     eax, [eax+esi*2+1E0h]
004FD56A 89 45 EC          mov     [ebp+cy], eax
004FD56D C7 45 F4 00 00 00+mov     [ebp+Rect.top], 0
004FD574 C7 45 F0 00 00 00+mov     [ebp+Rect.left], 0
004FD57B 8B 4D EC          mov     ecx, [ebp+cy]
004FD57E 89 4D FC          mov     [ebp+Rect.bottom], ecx
004FD581 8B 55 E4          mov     edx, [ebp+var_1C]
004FD584 89 55 F8          mov     [ebp+Rect.right], edx
004FD587 6A 00             push    0                               ; hMenu
004FD589 A1 18 20 70 00    mov     eax, GLOBAL_ClientWindow
004FD58E 50                push    eax                             ; hWnd
004FD58F FF 15 24 58 9A 00 call    ds:SetMenu
004FD595 6A EC             push    0FFFFFFECh                      ; nIndex
004FD597 8B 0D 18 20 70 00 mov     ecx, GLOBAL_ClientWindow
004FD59D 51                push    ecx                             ; hWnd
004FD59E FF 15 0C 58 9A 00 call    ds:GetWindowLongA
004FD5A4 50                push    eax                             ; dwExStyle
004FD5A5 6A 00             push    0                               ; bMenu
004FD5A7 6A F0             push    0FFFFFFF0h                      ; nIndex
004FD5A9 8B 15 18 20 70 00 mov     edx, GLOBAL_ClientWindow
004FD5AF 52                push    edx                             ; hWnd
004FD5B0 FF 15 0C 58 9A 00 call    ds:GetWindowLongA
004FD5B6 50                push    eax                             ; dwStyle
004FD5B7 8D 45 F0          lea     eax, [ebp+Rect]
004FD5BA 50                push    eax                             ; lpRect
004FD5BB FF 15 58 57 9A 00 call    ds:AdjustWindowRectEx
004FD5C1 B9 60 24 82 00    mov     ecx, offset dword_822460
004FD5C6 E8 CE 34 02 00    call    sub_520A99
004FD5CB 68 12 01 00 00    push    112h                            ; uFlags
004FD5D0 8B 4D EC          mov     ecx, [ebp+cy]
004FD5D3 51                push    ecx                             ; cy
004FD5D4 8B 55 E4          mov     edx, [ebp+var_1C]
004FD5D7 52                push    edx                             ; cx
004FD5D8 6A 00             push    0                               ; Y
004FD5DA 6A 00             push    0                               ; X
004FD5DC 6A FE             push    0FFFFFFFEh                      ; hWndInsertAfter
004FD5DE A1 18 20 70 00    mov     eax, GLOBAL_ClientWindow
004FD5E3 50                push    eax                             ; hWnd
004FD5E4 FF 15 20 58 9A 00 call    ds:SetWindowPos
004FD5EA 6A 00             push    0                               ; bErase
004FD5EC 6A 00             push    0                               ; lpRect
004FD5EE 8B 0D 18 20 70 00 mov     ecx, GLOBAL_ClientWindow
004FD5F4 51                push    ecx                             ; hWnd
004FD5F5 FF 15 1C 58 9A 00 call    ds:InvalidateRect
```

FULL SCREEN: OFF – CHANGED CODE

```
004FD57B 8B 55 E4          mov     edx, [ebp+var_1C]
004FD57E 89 45 FC          mov     [ebp+Rect.bottom], eax
004FD581 89 55 F8          mov     [ebp+Rect.right], edx
004FD584 A1 5C 21 70 00    mov     eax, hMenu
004FD589 8B 35 18 20 70 00 mov     esi, GLOBAL_ClientWindow
004FD58F 50                push    eax                             ; hMenu
004FD590 56                push    esi                             ; hWnd
004FD591 FF 15 24 58 9A 00 call    ds:SetMenu
004FD597 6A EC             push    0FFFFFFECh                      ; nIndex
004FD599 56                push    esi                             ; hWnd
004FD59A FF 15 0C 58 9A 00 call    ds:GetWindowLongA
004FD5A0 50                push    eax                             ; dwExStyle
004FD5A1 A1 5C 21 70 00    mov     eax, hMenu
004FD5A6 50                push    eax                             ; bMenu
004FD5A7 6A F0             push    0FFFFFFF0h                      ; nIndex
004FD5A9 56                push    esi                             ; hWnd
004FD5AA FF 15 0C 58 9A 00 call    ds:GetWindowLongA
004FD5B0 50                push    eax                             ; dwStyle
004FD5B1 8D 45 F0          lea     eax, [ebp+Rect]
004FD5B4 50                push    eax                             ; lpRect
004FD5B5 FF 15 58 57 9A 00 call    ds:AdjustWindowRectEx
004FD5BB 90                nop
004FD5BC 90                nop
004FD5BD 90                nop
004FD5BE 90                nop
004FD5BF 90                nop
004FD5C0 90                nop
```

THE PATCH MORE OR LESS EXPLAINED

What will happen after the patch?

Well, when you go to Full Screen Mode the fix will call “SetMenu(hWnd, NULL)” to remove the menu. “AdjustWindowRectEx” will be called with “hMenu” set to “FALSE”.

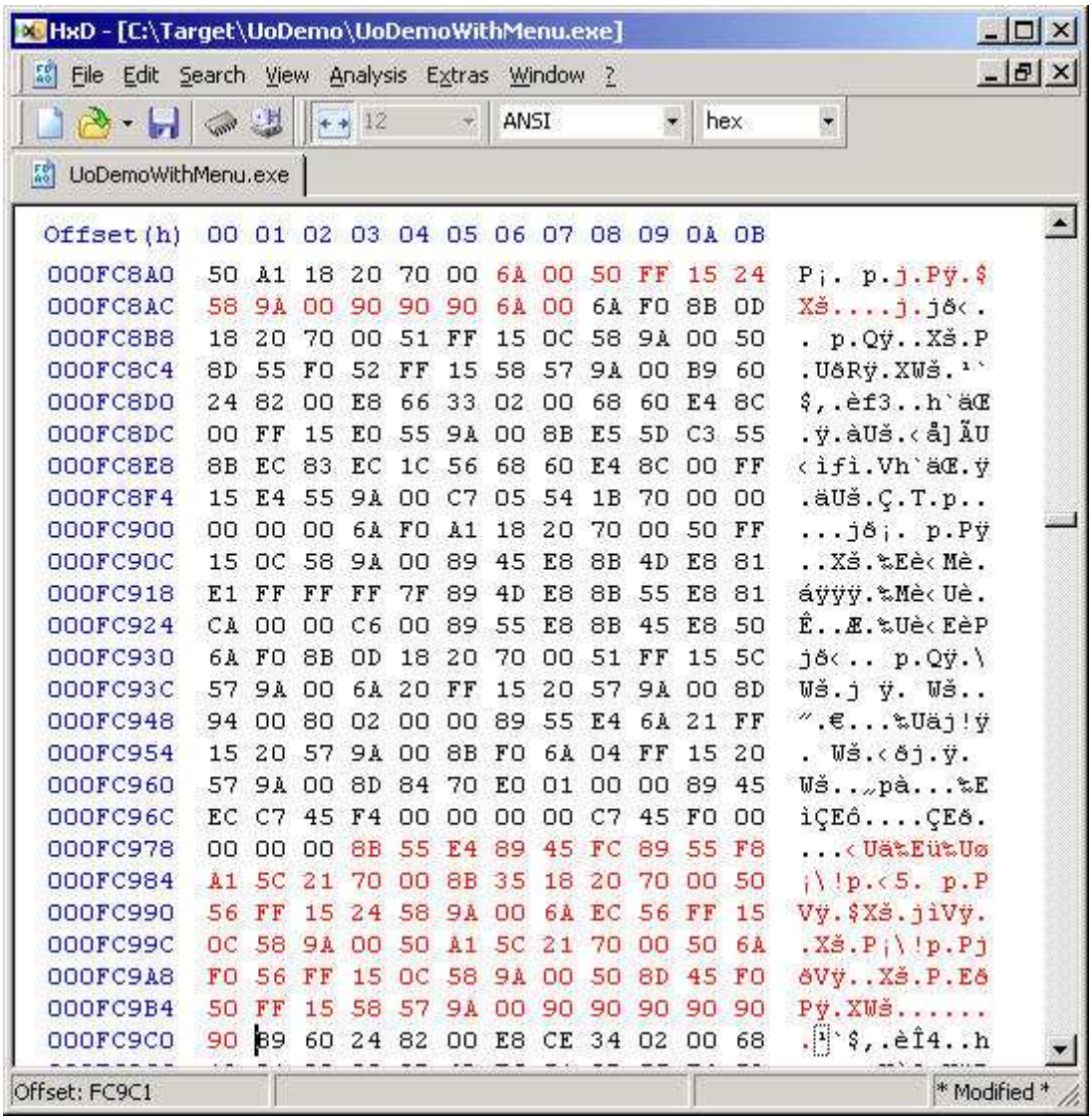
The opposite will happen when you enter Windowed Mode. The fix will call “SetMenu(hWnd, hMenu)” and will call “AdjustWindowRectEx” with “hMenu” converted to a “BOOL”.

Previously the client was not doing anything when going from Windowed Mode to full Screen mode. Strangely enough, “SetMenu” was being called to remove the menu when entering Windowed Mode.

NOTE: “hMenu” is coming from the call to “DestroyMenu”:

```
004FCB4C                   loc_4FCB4C:
004FCB4C 83 3D 5C 21 70 00+cmp     hMenu, 0
004FCB53 74 0D             jz      short loc_4FCB62
004FCB55 8B 15 5C 21 70 00 mov     edx, hMenu
004FCB5B 52                push    edx
004FCB5C FF 15 C8 57 9A 00 call    ds:DestroyMenu
```

A screenshot of a hex editor with the newly patched bytes:



Apply these changes and go play the Ultima Online Demo with a working menu :-).