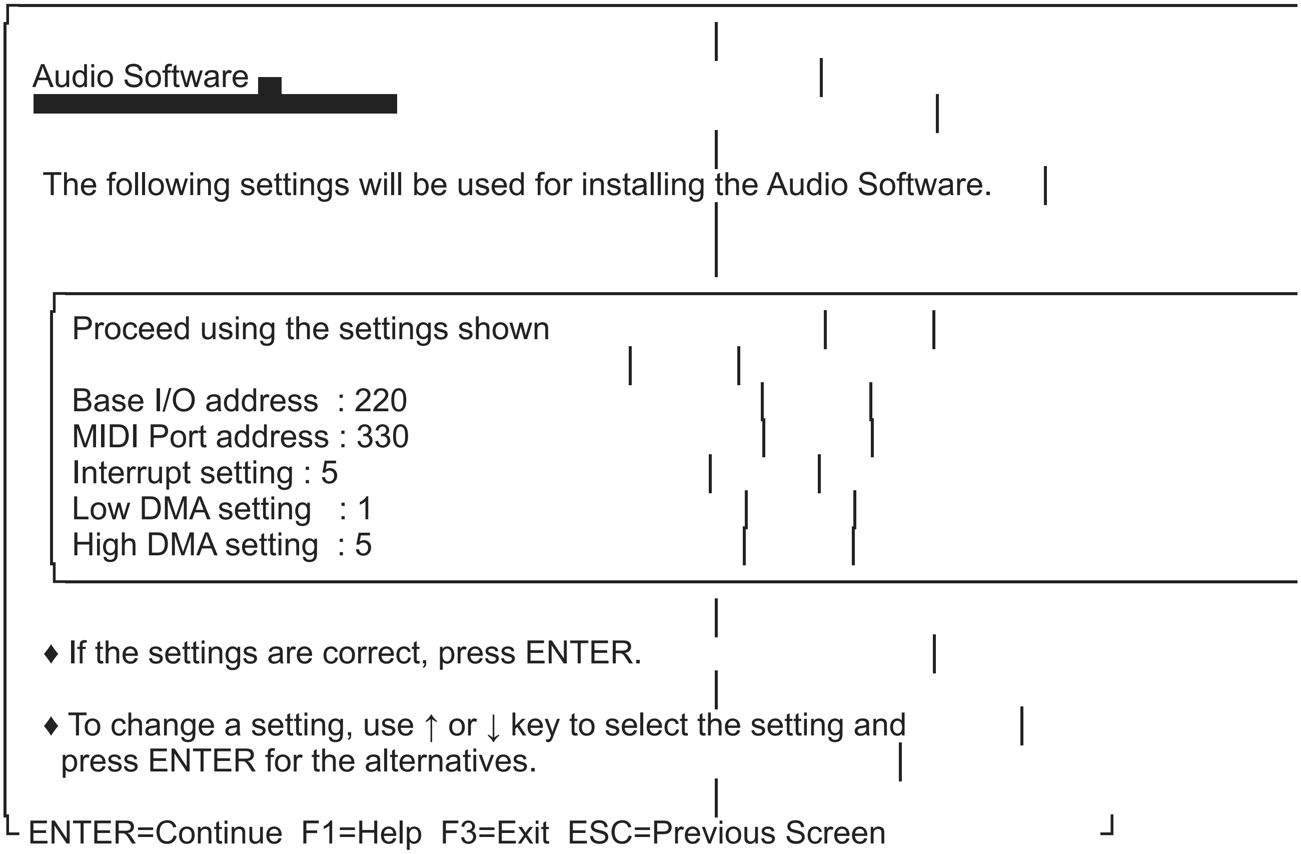
For one internal build, Windows 95 contained an evil message

devblogs.microsoft.com/oldnewthing/20191031-00

October 31, 2019

Raymond Chen

One of the many significant accomplishments of Windows 95 was the widespread adoption of Plug and Play, the idea (borderline heretical at the time) that devices could not only be software-configurable, but that the operating system could automatically resolve resource conflicts among hardware devices. Prior to Plug and Play, hardware devices were configured by setting jumpers or DIP switches, and users were expected to configure their devices so that no two of them used the same I/O ports, IRQ channels, etc., and then they were expected to configure the driver in order to tell it which I/O ports, IRQ channels, etc., to use so that it could access the hardware device.



```
Audio Software

 The following settings will be used for installing the Audio Software.

   Proceed using the settings shown

   Base I/O address  : 220
   MIDI Port address : 330
   Interrupt setting : 5
   Low DMA setting   : 1
   High DMA setting  : 5

 ♦ If the settings are correct, press ENTER.

 ♦ To change a setting, use ↑ or ↓ key to select the setting and
   press ENTER for the alternatives.

ENTER=Continue  F1=Help  F3=Exit  ESC=Previous Screen
```

As I noted some time ago, an early design for the `WM_ DEVICEBROADCAST` message required applications to return `TRUE` to allow the operation to proceed, or `FALSE` to block the operation. One consequence of this design was that any window procedure that simply returned `FALSE` in response to all unknown messages would unwittingly prevent any device configuration from occurring.

To help identify programs that improperly returned `FALSE` in response to the `WM_ DEVICEBROADCAST` message, debug builds of Windows 95 periodically sent out a message saying, "I'm reconfiguring this device that you've never heard of. Are you okay with that?" If anybody said, "No, I'm still using that device," then the Plug and Play system knew that it had found a buggy window procedure. (Sound familiar?)

The Plug and Play team nickname for "the device node (*devnode*) you never heard of" was *the Hell devnode*. As I recall, it got this name because this particular devnode had every possible thing wrong with it, in order to stress the rest of the Plug and Play system.

The message that was displayed when one of these buggy window procedures was found went something like "BUG! BUG! BUG! A window procedure blocked the removal of the Hell devnode." As I recall, the message was not particularly clear about what you should do if you saw this message, and as a result, people tended to file bugs against the `MessageBox` function for displaying such an unhelpful message in the first place.

For one internal build, however, the message changed. When a buggy window procedure was found, the message was "Muhahahahaha! 666! Plug and Play is the devil! Prepare for eternal damnation!"

This message caused quite a bit of consternation. A member of the user interface team confessed to being the one who changed the message. It was partly mocking the Plug and Play team, playing on the whole "Hell devnode" name, and I suspect partly out of frustration for the bugs that kept being filed against the user interface team as a result of this dialog box.

The message was replaced with something a bit less frightening for the next build.

Raymond Chen

Follow