
Detecting a Pending Reboot in the User Interface Sequence

One way Windows Installer minimizes the number of reboots required by a setup program is by prompting the user to exit any open applications before data transfer takes place. However, if the user chooses to ignore the running application, or the running application does not have a window associated with it, a reboot may still be required.

It is common to want give the user the option to launch an installed executable, by using a custom action attached to a checkbox appearing on the SetupCompleteSuccess dialog box. A problem arises, however, if the offer to launch the application should be suppressed if a reboot is going to be required: the value of ReplacedInUseFiles is set in the Execute sequence, while SetupCompleteSuccess appears in the User Interface sequence, and properties are not passed "backward" from the Execute sequence back to the User Interface sequence.

Looking through the Windows Installer help, it turns out there is another technique for determining if a reboot is required: calling the MsiGetMode API with the MSIRUNMODE_REBOOTATEND constant (or the Mode property in the MSI Automation interface) will return TRUE if a reboot is required.

To disable a checkbox appearing on the SetupCompleteSuccess dialog box, we can define a property called REBOOT_REQUIRED, to be set near the end of the User Interface sequence if a reboot is required. To call a custom action in the User Interface sequence after the Execute sequence has completed, we insert it after the ExecuteAction action in the Sequences view of the IDE

In VBScript, the custom action to be called after ExecuteAction might appear as follows.

```
If Mode(6) Then ' i.e., msiRunModeRebootAtEnd
    Property("REBOOT_REQUIRED") = "1"
End If
```

We can then insert REBOOT_REQUIRED as a Hide control condition to the checkbox appearing on the SetupCompleteSuccess dialog box.