Hiding the Cancel Button on the MSI Progress Dialog

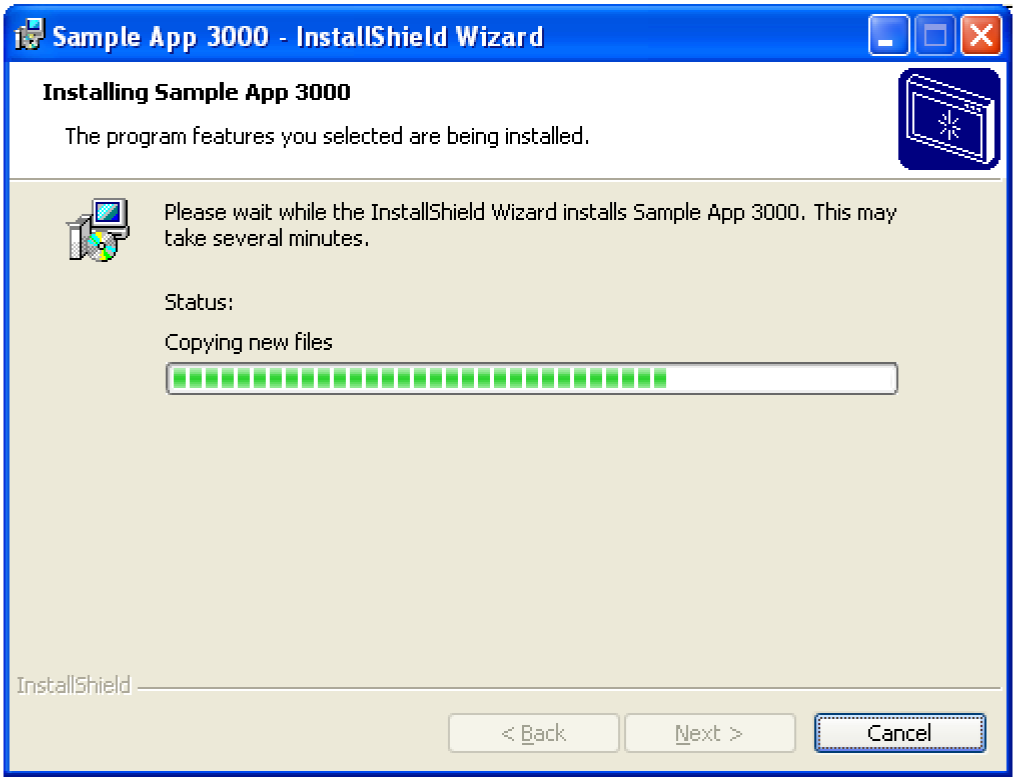
Robert Dickau
Senior Technical Content Manager

Abstract

This article discusses various techniques for hiding the Cancel button on the various progress dialog boxes displayed by a Windows Installer (Basic MSI) installation.

Default Progress Dialog Boxes

While a Windows Installer installation makes changes to a target system, by default it displays feedback to the user on a progress dialog box. For a full-user interface installation or uninstallation, the SetupProgress dialog box is displayed.



Sample App 3000 - InstallShield Wizard

Installing Sample App 3000

The program features you selected are being installed.

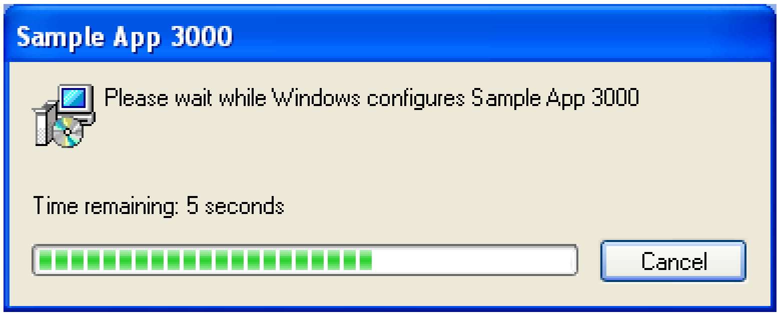
Please wait while the InstallShield Wizard installs Sample App 3000. This may take several minutes.

Status:

Copying new files

For a Basic UI installation—one performed by running the command **msiexec /i product.msi /qb** or by launching an uninstallation from the Remove button on the Add or Remove Programs tool—a smaller progress dialog box similar to the following is displayed:

©2010 Flexera Software, Inc. and/or InstallShield Co. Inc. All rights reserved.



Sample App 3000

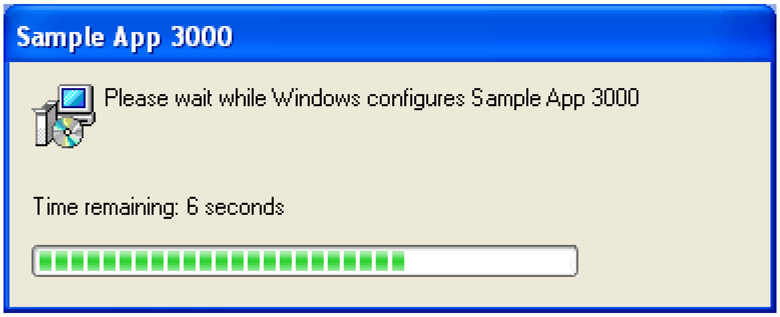
Please wait while Windows configures Sample App 3000

Time remaining: 5 seconds

In both cases, the progress dialog displays a Cancel button, with which the user can exit the installation. If the installation is running in deferred mode, any rollback actions contained in the execution script will be performed.

Disabling the Cancel Button

In some cases, it is desirable to disable the Cancel button. For a Basic UI installation, the Cancel button can be disabled by adding the exclamation point to the /qb switch, as in **msiexec /i product.msi /qb!**. In this case, the progress dialog box appears as follows.



Sample App 3000

Please wait while Windows configures Sample App 3000

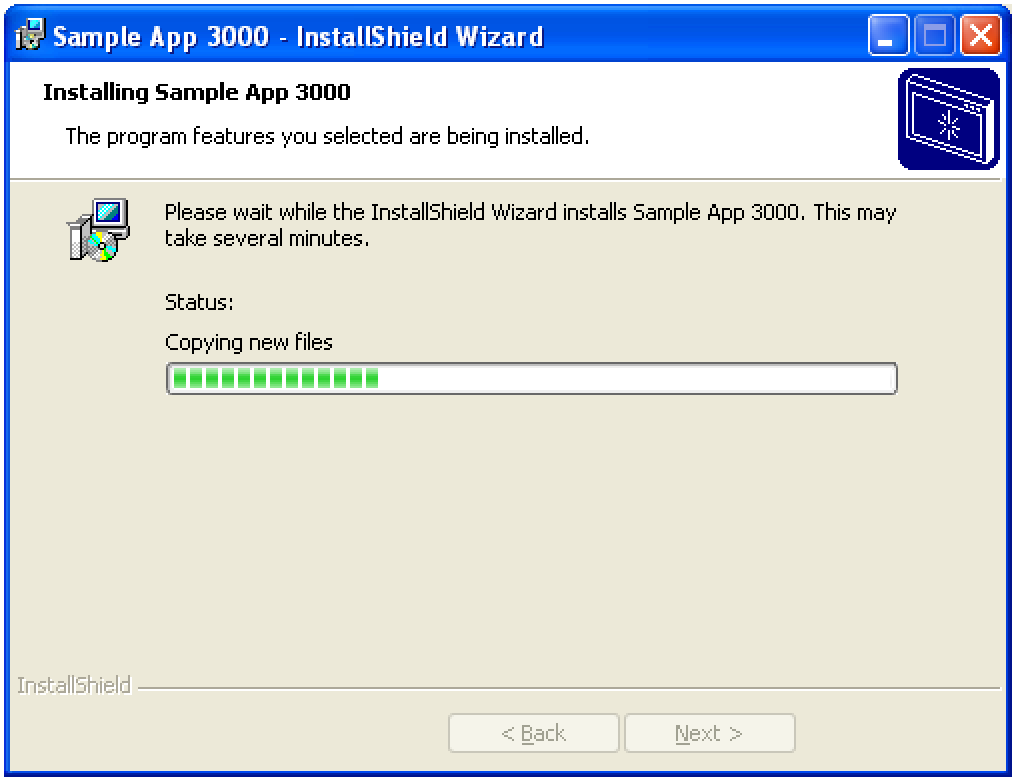
Time remaining: 6 seconds

Disabling the Cancel button for a full-UI installation requires a custom action. The MsiProcessMessage function and Session.Message method support constants that can suppress the Cancel button while data transfer is taking place. In a VBScript custom action, the code might appear as follows:

```
Function HideCancel( )

Const msiMessageTypeCommonData = &H0B000000

Set rec = Installer.CreateRecord(2)

rec.IntegerData(1) = 2
rec.IntegerData(2) = 0

Message msiMessageTypeCommonData, rec

HideCancel = 1 ' return success to MSI

End Function
```

©2010 Flexera Software, Inc. and/or InstallShield Co. Inc. All rights reserved.

If you call this code in an immediate-mode custom action scheduled in the Execute sequence after InstallInitialize, the SetupProgress dialog box appears as follows.



Sample App 3000 - InstallShield Wizard

Installing Sample App 3000

The program features you selected are being installed.

Please wait while the InstallShield Wizard installs Sample App 3000. This may take several minutes.

Status:

Copying new files

For a reduced-UI or basic-UI installation, the Cancel button is similarly hidden.

Code for an MSI DLL custom action that performs the same task might appear as follows.

```
#pragma comment(lib, "msi.lib")

#include <windows.h>

#include <msi.h>
#include <msiquery.h>

UINT __stdcall HideCancelButton(MSIHANDLE hInstall)
{
    PMSIHANDLE hRecord = MsiCreateRecord(2);

    if (!hRecord)
        return ERROR_INSTALL_FAILURE;

    MsiRecordSetInteger(hRecord, 1, 2);
    MsiRecordSetInteger(hRecord, 2, 0);

    MsiProcessMessage(hInstall, INSTALLMESSAGE_COMMONDATA, hRecord);

    return ERROR_SUCCESS;
}
```

©2010 Flexera Software, Inc. and/or InstallShield Co. Inc. All rights reserved.

By default, because the custom action is called for first-time installations and maintenance operations (including uninstallation), the Cancel button will also be hidden during maintenance mode and uninstallation. If you want control over the circumstances during which the Cancel button is hidden, you can attach a condition to the custom action. For example, to hide the Cancel button only during uninstallation, you can use the condition **REMOVE="ALL"** in the Execute sequence (somewhere after the InstallValidate action). For more information about this type of condition, see "Installation and Uninstallation Conditions".

To re-display the Cancel button, you can use a similar custom action, in which the second field of the record passed to Message or MsiProcessMessage is set to 1 instead of 0.

For more information, see the Windows Installer Help Library topics "MsiProcessMessage" and "Session.Message".

©2010 Flexera Software, Inc. and/or InstallShield Co. Inc. All rights reserved.