

# Performing Silent Installations and Uninstallations

By Robert Dickau  
Principal Technical Training Writer

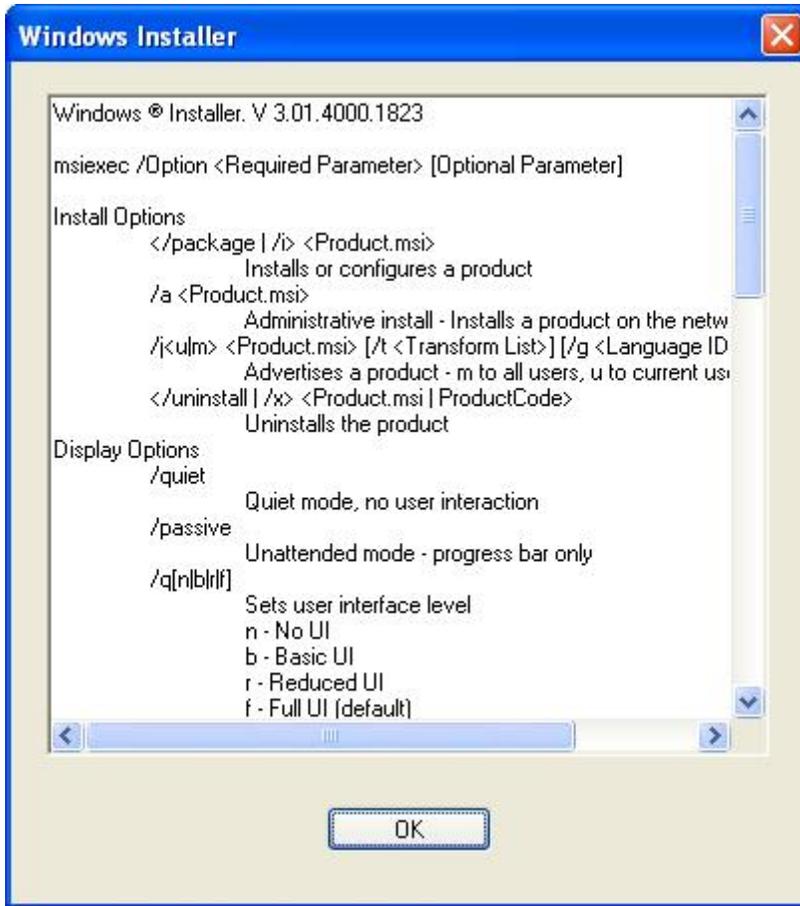
## Abstract

A silent, or unattended, installation is one that displays no feedback to the end user. This article provides an overview of how to perform silent installation and uninstallation of products installed with Windows Installer-based installers.

**Please note:** This article also includes information on how to perform silent installation and uninstallation of InstallScript-based installers, although **it is strongly recommended that you repackage InstallScript-based installers into MSIs before deploying them to your environment.**

## Windows Installer (MSI) Installations

Part of every Windows Installer-based installation is the MSI database. (In your repackaging settings in AdminStudio, you can specify whether data files should be compressed within the MSI database or stored externally to it as compressed cabinet files or an uncompressed directory structure.) The Windows command-line executable for processing MSI databases is *msiexec.exe*, which is located in a target system's System folder. To see a list of command-line switches supported by the Windows Installer executable, you can enter the command *msiexec* in a Command Prompt window or the Run box. The following figure shows the help window for a recent version of the Windows Installer engine.



As described in the msiexec.exe help message, you can use the **/i** switch to perform an installation. A typical command for performing an installation is the following:

### **msiexec /i SampleApp.msi**

This is the same command used by the operating system when a user double-clicks an MSI file icon.

To perform a silent installation, you can add the **/q** switch to the msiexec command. For example, the following command runs a silent installation:

### **msiexec /i SampleApp.msi /q**

(Starting with Windows Installer version 3.0, you can also use the **/quiet** switch to perform the same action.)

During a normal graphical installation, the installer's dialog boxes are used to specify the settings for the installation. Running an installation in silent mode performs the installation with default settings for the registered user name, installation directory, and so forth. To modify the settings used for a silent installation, you can pass in the values of *public properties*, which are Windows Installer properties with all-uppercase names. Public properties that are commonly modified include:

- USERNAME: the name of the user registered to use the software
- COMPANYNAME: the name of the organization registered to use the software
- INSTALLDIR: the directory into which the software is installed
- ALLUSERS: indicates whether to perform a per-machine or per-user installation
- TRANSFORMS: names of one or more transforms (.mst files) to be applied to the base MSI database

The Windows Installer Help Library describes these and other predefined public properties used by the Windows Installer engine. It is also possible for installation authors to define custom properties.

A typical command for a silent installation that modifies the values of some public properties might appear as follows:

```
msiexec /i SampleApp.msi /q INSTALLDIR="C:\Sample App"
COMPANYNAME="Sample Customer Org." ALLUSERS=2
```

(If a property value contains any spaces, as in the values of INSTALLDIR and COMPANYNAME in this example, you must surround it with quotation marks.)

You can use the Tuner and Editor tools in AdminStudio to generate a *transform*, which is a file containing the differences between the original MSI database from a vendor and a modified database with your desired changes to properties and other settings. In particular, creating a *response transform* will run the user interface of your Windows Installer installation, and populate a transform with the modified property values changed by the user interface.

To silently deploy a database along with a transform created with Tuner or Editor, you can use a command similar to the following:

```
msiexec /i SampleApp.msi /q TRANSFORMS=OurSettings.mst
```

To uninstall a product, you can use the `/x` command to msiexec, specifying either the MSI database or the desired product code:

```
msiexec /x SampleApp.msi
```

(Starting with Windows Installer version 3.0, you can use the equivalent `/uninstall` switch.)

As with installation, you can add the `/q` switch to perform a silent uninstallation:

```
msiexec /x SampleApp.msi /q
```

## InstallScript Installations

For an InstallScript-based installer, the user interface populates the values of script variables. Instead of directly passing the values of variables to the installer, you perform the following steps:

1. On a test system, run the installation in record mode, which generates a *response file*
2. On an end user's system, run the installation in silent mode, using the data in the

response file in place of user input

To generate the response file, you run the InstallScript setup.exe with the /r switch. This displays the installer's dialog boxes, and additionally records the data you enter in the dialog boxes to a text response file. By convention, a response file uses the .iss (for "InstallShield silent") file name extension; you can specify the name and location of the response file using the /f1 switch.

**setup /r /f1"C:\sample\setup.iss"**

(Note there is no space between /f1 and the quoted path to the response file to create.)

The response file is in the form of an INI file. Contents of a typical response file appear similar to the following:

```
[InstallShield Silent]
Version=v7.00
File=Response File
[File Transfer]
OverwrittenReadOnly=NoToAll

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-DlgOrder]
Dlg0={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdWelcome-0
Count=8
Dlg1={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdLicense2Rtf-0
Dlg2={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdRegisterUser-0
Dlg3={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SetupType2-0
Dlg4={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdAskDestPath2-0
Dlg5={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdComponentTree-0
Dlg6={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdStartCopy2-0
Dlg7={9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdFinish-0

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdWelcome-0]
Result=1

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdLicense2Rtf-0]
Result=1

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdRegisterUser-0]
szName=RMD
szCompany=Sample Customer
Result=1

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SetupType2-0]
Result=303

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdAskDestPath2-0]
szDir=C:\Program Files\Macrovision\Sample App
Result=1

[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdComponentTree-0]
szDir=C:\Program Files\Macrovision\Sample App
Component-type=string
```

```
Component-count=1
Component-0=MainProgram
Result=1
```

```
[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdStartCopy2-0]
Result=1
```

```
[Application]
Name=Sample App
Version=1.00.0000
Company=Macrovision
Lang=0009
```

```
[{9F9CA66B-138C-48FF-80E8-2C767B855C20}-SdFinish-0]
Result=1
bOpt1=0
bOpt2=0
```

The response file contains a list (in the DlgOrder section) of all the dialog boxes used, followed by a section for each dialog box. The section for each dialog box contains the variable names and values used by that dialog box, followed by a "Result" value indicating which button the user clicked to dismiss the dialog box (typically 1 for the Next button of a dialog box). For example, the section for SdRegisterUser lists the szName and szCompany variables, along with the value you entered for each when you ran the installer in record mode.

To deploy the installer silently based on the response file, you launch setup.exe with the /s switch, again using the /f1 switch to specify the location of the response file.

**setup /s /f1"C:\sample\setup.iss"**

(If you intend to perform the silent installation on the same system used to generate the response file, you must uninstall the product first.)

Note that a silent installation displays no dialog boxes, even if an error occurs. To help you determine if a silent installation was successful, setup.exe generates a log file called setup.log. By default this file is created in the same directory as the .iss response file, and you can specify a different location using the /f2 switch.

Part of a typical log file appears similar to the following:

```
[InstallShield Silent]
Version=v7.00
File=Log File
[ResponseResult]
ResultCode=0
```

The highlighted ResultCode line indicates whether the silent installation was successful: a ResultCode value of 0 indicates success, and different negative numbers indicate different types of failure. Common error codes are -3, indicating a missing value from the response file; -5, meaning the response file could not be found; and -12, indicating that the dialog boxes in the response file are different from those encountered during the silent installation. For a complete list of error codes and their interpretations, see the "Setup.log" topic in the

online help.

Because a response file must describe the same sequence of dialog boxes actually used by a particular installation, to perform a silent uninstallation you must generate a response file for the uninstall sequence of dialog boxes. To do this, you can run the installer when the product is already installed, generating responses for the uninstall option of maintenance mode:

```
setup.exe /r /f1"C:\sample\uninstall.iss"
```

To perform the silent uninstallation on another system that already has the product installed, you can specify the appropriate response file:

```
setup.exe /s /f1"C:\sample\uninstall.iss" /f2"C:\sample\uninstall.log"
```

Again, the above information on InstallScript-based installers was meant for educational purposes. **It is strongly recommended that you repackage InstallScript-based installers into MSIs before deploying them to your environment.**