

InstallShield Setup.exe Command-Line Parameters

InstallShield setup.exe files can accept a number of command-line parameters. Using these parameters, administrators can specify details like where to install, if it should reboot the system, or what (if anything) should be displayed during installation.

Command-line options that require a parameter must be specified with no space between the option and its parameter. Quotation marks around an option's parameter are required only if the parameter contains spaces.

Several of the command-line options apply only to MSI projects or InstallScript projects. What is the difference? InstallScript MSI setups are Windows Installer setups driven by InstallScript, whereas InstallScript setups are written purely in InstallShield Script. This is often referred to as a "legacy setup".

Setup.exe accepts the following command-line parameters regardless of if it is an InstallScript or InstallScript MSI project.

Parameter	Action	InstallScript Supported?	InstallScript MSI Supported?
/uninst	Uninstalls the product	■	■
Setup.exe /uninst			
/removeonly	Also uninstalls the product, but depending upon how authored it may display the welcome/maintenance dialog	■	■
Setup.exe /removeonly			
/r	Record the installation to create an ISS answer file which may be used in a silent installation command	■	■
Setup.exe /r Setup.exe /r /f1"c:\temp\my-answer-file.iss"			
/s	Run the setup silently using a previously created ISS answer file (with the /r parameter).	■	■
Setup.exe /s Setup.exe /s /SMS Setup.exe /s /f1"c:\temp\my-answer-file.iss" Setup.exe /s /f1"c:\temp\my-answer-file.iss" /f2"c:\temp\my-log-file.iss"			

/f1	Specifies an alternative response file name and path; by default /s will look for setup.iss in the same directory as the setup executable and /r will create setup.iss in the Windows directory (%windir%).	■	■
<pre>Setup.exe /r /f1"c:\temp\my-answer-file.iss" Setup.exe /s /f1"c:\temp\my-answer-file.iss"</pre>			
/f2	Specifies an alternative log file name and path; by default the log file will be created in the same directory as the setup executable with the name setup.log.	■	■
<pre>Setup.exe /s /f2"c:\temp\my-log-file.iss" Setup.exe /s /f1"c:\temp\my-answer-file.iss" /f2"c:\temp\my-log-file.iss"</pre>			
/SMS	Instructs a silent setup not to release the current session until the installation is complete. Without this switch a silent installation will immediately appear complete and run in the background. Use this switch to have any additional installations or script actions wait for installation to complete (such as in a batch file). The parameter was introduced to support MS Systems Management Server which would temporarily map a drive during installation to access source files (where it would immediately complete, and SMS would unmap the drive to the source files before the installation was complete.)	■	■
<pre>Setup.exe /s /SMS Setup.exe /s /SMS /f1"c:\temp\my-answer-file.iss" Setup.exe /s /SMS /f1"c:\temp\my-answer-file.iss" /f2"c:\temp\my-log-file.iss"</pre>			
/m	Generate a .MIF file (used by SMS and some other management solutions to report status information).	■	■
<pre>Setup.exe /m"MyMIF"</pre>			
/m1	Specifies the serial number to be written to a .MIF file (used with /m parameter)	■	■
<pre>Setup.exe /m"MyMIF" /m1"1111-2222-" Setup.exe /m"MyMIF" /m1"1111-2222-" /m2"ENU"</pre>			
/m2	Specifies the locale string to be written to a .MIF file (used with /m parameter)	■	■

<pre>Setup.exe /m"MyMIF" /m2"ENU" Setup.exe /m"MyMIF" /m1"1111-2222-" /m2"ENU"</pre>			
/d	<p>Debug InstallScript portion of the setup with the InstallScript debugger (ISDbg.exe). This is intended for setup authors to troubleshoot a setup and not for admins- the action requires a setup.dbg file to be available.</p>	■	■
<pre>Setup.exe /d"<path to folder containing setup.dbg>" Setup.exe /v"ISSCRIPTDEBUG=1 ISSCRIPTDEBUGPATH=\"<path to folder containing setup.dbg>\""</pre>			
/f	<p>Specify alternative compiled script. Setup.exe looks for a compiled script file named Setup.inx, this parameter may be used to specify a different name for the compiled script file. Note that in earlier versions of InstallShield (InstallShield - Windows Installer Edition), this parameter served to initiate a repair of the installation.</p>	■	■
<pre>Setup.exe /f"AlternateSetup.inx"</pre>			
/L	<p>Specifies the language for the setup for those setups that support multiple languages. Language must be identified by decimal (LCID).</p>	■	■
<pre>Setup.exe /L1031 Setup.exe /L0007</pre>			
/delayedstart	<p>Delay initialization of the installation. With this parameter, specify the amount of time (in seconds) by which initialization of the installation is to be delayed after Setup.exe is launched.</p>	■	□
<pre>Setup.exe /delayedstart:10 Setup.exe /s /delayedstart:900</pre>			
/deleter	<p>Do not clone a second Setup.exe process when debugging. By default, Setup.exe clones a second process so that it properly uninstalls applications from machines on which no InstallScript installation had previously been run. Use this option when debugging a DLL function that is called from your own authored installation script.</p>	■	□

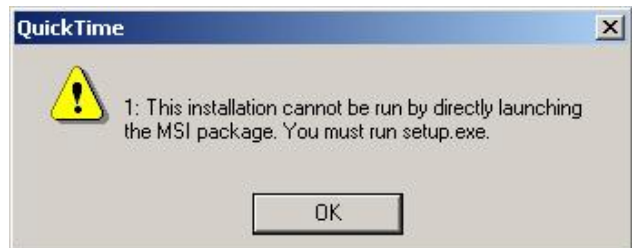
Setup.exe /deleter			
/extract_all	Extract a package's files but do not run the setup.	■	□
Setup.exe /extract_all:"c:\temp"			
/hide_usd	Suppress update dialog box for multiple installations. When this parameter is used and an update-enabled installation detects multiple previous installations, the installation updates the first previous installation that it finds without prompting the user to select which product installation to update (default).	X	
Setup.exe /uninst			
/ig	Specify the value of the system variable INSTANCE_GUID. If this option is not used, the installation automatically assigns a value to INSTANCE_GUID (for multi-instance installations, this value is a newly generated GUID; for standard installations, this value is the same as the value of PRODUCT_GUID).	X	
Setup.exe /uninst			
/z	This parameter may be used to pass data to the InstallScript system variable CMDLINE.	X	
Setup.exe /uninst			
/a	Perform an administrative installation. An administrative installation copies (and decompresses) your data files to a directory specified by the user, but it does not create shortcuts, register COM servers, or create an uninstall log.	X	X
Setup.exe /uninst			
/j	Advertise the installation. An advertised installation creates shortcuts, registers COM servers, and registers file types, but does not install your product's files until the user invokes one of these installed entry points.	X	X
Setup.exe /uninst			
/x	Uninstall the installation (if previously installed).	X	X

Setup.exe /uninst			
/p:<password>	For a password protected setup, this option may be used to supply the password at the command line (normally in support of a silent installation).	X	X
Setup.exe /uninst			
/s	Install silently. By default the silent installation is based on the response file called Setup.iss in the same directory. (Response files are created by running Setup.exe with the /r option.) To specify an alternative file name or location of the response file, use the /f1 parameter.	X	X
Setup.exe /uninst			
/ua:<URL to InstMsiA.exe>	If download locations are specified in the setup, those paths may be overridden using this parameter. Note that only the path is required.	□	X
Setup.exe /uninst			
/uw:<URL to InstMsiW.exe>	If download locations are specified in the setup, those paths may be overridden using this parameter. Note that only the path is required.		X
Setup.exe /uninst			
/us:<URL to ISScript.msi>	If download locations are specified in the setup, those paths may be overridden using this parameter. Note that only the path is required.		X
Setup.exe /uninst			
/um:<URL to .MSI package>	If download locations are specified in the setup, those paths may be overridden using this parameter. Note that only the path is required.		X
Setup.exe /uninst			
/b<local directory>	If a path to locally cache setup files is specified, this path may be overridden using this parameter.		X
Setup.exe /uninst			

/v:<MSIEXEC parameters>	Pass command-line options and values of public properties through to Msiexec.exe.		X
Setup.exe /uninst			
/w	Forces Setup.exe to wait until the installation is complete before exiting (similar to /SMS).		X
Setup.exe /uninst			
/h	By default, the build engine automatically creates an installation that supports Setup.exe cloning in cases where cloning is required (such as in multi-disk installations). Specify this parameter to have Setup.exe clone itself to a temporary location and run from that location.		X
Setup.exe /uninst			

For more on how to work with InstallScript setups, see "Dealing with InstallScript for Administrators" in the AppDeploy Library (www.appdeploy.com/library). In this video presentation you will learn what InstallScript is, why it is still used and other general information on the topic before explaining the three ways of avoiding the need for running the setup from the provided setup.exe. Including video demonstrations of each:

- Removing the custom action that checks for the setup.exe call by creating a transform (MST) file
- Tricking the MSI into thinking it is being called by the setup.exe using the ISSETUPDRIVEN property
- Using the InstallScript Scan feature of AdminStudio to migrate the InstallScript to native Windows Installer (so InstallScript might be removed as a dependency)



For more details, visit <http://www.appdeploy.com/video/installscript.asp>