

## Working with Single File Installers

Install Solutions ([www.InstallSolutions.net](http://www.InstallSolutions.net))

In the past most software installation “packages” were not actually packaged at all. A batch file or custom setup.exe file was created to copy files from a local media, such as a CD or floppy disk, and register any necessary dll or exe files. When the internet came along much of that changed because of the unreliability of downloading multiple files necessary for an installation and the increased bandwidth and time necessary for downloading files in an uncompressed version. Now it is very common to download a single MSI or EXE installation file that contains all of the necessary files and setup information to install the software product. This same approach is a best practice for deploying software internally throughout your corporation. It helps to maintain the integrity and reliability of your installation package.

While many vendors now ship their products in an MSI format for installation, some still deliver their software on CD's containing several file and directories necessary for the installation. For the ones that are already in a single file installer format, typically an MSI, some common switches are necessary to make the installation run silently or unattended which is the first requirement for any software deployment package. For MSI's it is generally a /qn command line. It would look like this in most scenarios: `myInstallerFile.msi /qn`. You can see more MSI command line switches [here](#) at Microsoft's site. For the multi-file installation packages it is necessary to bundle them up in a single file installer format. There are several options for doing this.

### ***Option 1: The easiest way, creating a self-extracting executable***

The first step is to determine if the install package will install unattended using any command line switches. Most vendors will have this information, or if you know what type of installer package is being used in the setup, i.e. InstallShield or Wise etc., then you may be able to find the correct method for running them silently. InstallShield setups typically require an answer file which will be called something.iss. These are created by starting the installation using the proper command line which tells the installer to record the options you select. Then you can specify the created answer file (.iss) the next time you run setup for an unattended install. Go [here](#) to see more about InstallShield installers. If the installer is Wise then usually a /S will suffice.

If the installer can install silently then you can bundle it into a self-extracting zip file. Most self-extracting executable creators will allow you to specify a setup file to call once the files have been extracted. In short a self-extracting executable is a zipped up version of your CD installation. Instead of creating a .zip file it will be a .exe file that will automatically extract all contents to a location on the hard drive, then run the setup file, with any command line parameters, that kicks off the install just like from the CD. Install Solutions has a free [Quick Install Maker](#) that does just this.

### ***Option 2: Repackaging the installation***

In some cases it may be necessary to repackage an existing installation even if it is already a single file installation. Most of the time it is because the installation package has not been designed to run unattended. This is usually accomplished by using a packaging application like Altiris's Wise Package Studio or Macrovision's InstallShield. These tools perform a snapshot method of creating an installation package. This is accomplished by creating a before snapshot of the computer prior to the application being installed followed by an after snapshot of the system once the application has been installed and any configurations have then been made. The difference is automatically bundled up into an installation package. In this case it is necessary to also decide on the format of the installer package, MSI or EXE.

### ***MSI or EXE based installations?***

*MSI* is Microsoft's Installer standard. It first came on the scene with Office 2000. Unlike exe installations in the past MSI's are a database driven install. Inside each MSI is a small data table of information that correlates files to directories and versions to files and files to GUID's etc. The best thing about MSI packages is that they follow a standard for command line switches for silent installs, patches, upgrades, and have several features for advertising the application to a desktop without actually installing it. The drawback to MSI's installations is that even for a small application they can write a great deal of information to the registry and are rather complex in their install routine. Upgrades to existing applications, especially multi-version updates, can be difficult to create. Since each file is tied to a GUID in the MSI and local registry these have to be updated before files can be replaced. So a simple 1 file patch can be a complex process.

*EXE* is a script based installation. This means that it runs top down just like reading a book or running a VBScript. What you see is what you get which is what makes this installation format a top choice for many companies that are constantly rolling out updates or changes to their environment. Also, because of powerful scripting engine, these packages are often used for much more than application installation. The drawback is that outside of an internal business environment there really is no standard for exe based installers. Especially when it comes to running them unattended, each vendor adopts their own method.

***More information can be found at these resources:***

[Command line switches for unattended installations](#) (AppDeploy.com)

[Silent installations for specific packages](#) (AppDeploy.com)

[Remote Software Installation Utility](#) (Install Solutions)