


---

[**Editor’s Note:** The following excerpt is from Chapter 5 of the free eBook *The Definitive Guide to Windows Installer Technology for System Administrators* (Realtimepublishers.com) written by Jeremy Moskowitz and Darwin Sanoy and available at <http://www.wise.com/ebook/>.]


## Source Lists—the Good and the Bad

In the last chapter, source list management was mentioned as a necessary best practice. As mentioned in that chapter, each package installed on a computer keeps a list of places where the package source files can be found. Package source files include the package file itself (MSI), any needed transforms (MST), and the actual software application files. The software application files can be embedded in the MSI, external from the MSI in .cab files, or external from the MSI and uncompressed. Aside from the installation of the application itself, they are also used for self-healing, install-on-demand, patching, and for server-based client applications. Source locations are also used to re-cache missing transforms if they are unavailable in the local Windows Installer file cache.

 Microsoft has enhanced Windows Installer 2.0 so that it does not need to touch the source files as frequently as previous versions do. In addition, Windows Installer 2.0 supports building patches that do not require access to the original package source files.


Source lists are intended to provide fault tolerance in finding source files, so they are processed sequentially until a valid source location is found. Source lists do not inherently provide any type of load balancing. A poor man’s load balancing can be accomplished by randomizing the source list when the package is installed.

Source lists can be managed more loosely in a by-the-book enterprise IntelliMirror implementation because packages are generally installed from a DFS share. The DFS share handles fault tolerance as well as several valuable features. A by-the-book implementation will need to manage source lists if the overall approach includes any mobile machines that will be installing from removable media.

 When no efforts are made to manage the source list for a package, the source defaults to the location of the package file when the initial installation begins. A casual approach to installation locations was the norm with setup.exe technologies, and most administrators do not discover this painful Windows Installer behavior until they have quite a mess on their hands.


There are three types of source lists—*removable media sources* for packages located on CD-ROMs, floppy drives, and so on; *network-based sources* for packages located on network drives or UNC’s; and *URL-based sources* for packages located on Web servers.

---

 Drive imaging is a popular method of deploying ready-made desktop system configurations. If Windows Installer packages are deployed on the desktop image, special care must be taken to ensure that the embedded Windows Installer source lists will be relevant in all the physical locations where the image will be used. Because decisions to use drive imaging might occur in a different part of your IT organization, this nuance can be lost in the shuffle of desktop engineering.


The following list provides the basic actions that can be taken to manage the source list:

- Manage which type of source lists are scanned first—removable media-based, network-based, and URL-based (SearchOrder policy).
- Add source list entries one or more at a time in a specified order (SOURCELIST public property or scripting API call).
- Force the SourceDir private property value so that the *recorded* initial location is different from the *actual* initial location (custom action). Doing so can only be accomplished if the forced location is available to the client at the time of installation.
- Use a custom action to make MSI API calls to clear the source list and add new sources.

 WindowsInstallerTraining.com publishes a script called MSICAResetSources (MSI Custom Action Reset Sources) that inserts a custom action into existing MSI files that will implement the approach in the last bullet point. This script uses the default public property SOURCELIST to populate the source list—essentially overriding the normal behavior of making the install location the first source location. If SOURCELIST does not exist, the custom action does nothing, which ensures that your package defaults to the normal expected behavior. You can download this script from <http://windowsinstallertraining.com/msiebook>.

You can combine these methods to do things such as remove the original install location and add managed locations or add network sources and reorder scanning so that network sources are preferred over media installs (such as with laptops that load software from CD-ROMs).

Environment variables can also be used in source lists to make the lists dynamically point to site-level repository locations. If you are using an environment variable strategy rather than a managed UNC or drive location, the source list must also be cleared of the initial installation location. The reason is that any environment variables used in the initial installation location will be resolved to a literal location before Windows Installer processes the package. This literal location then becomes the first source list location—which does not include the environment variable to make it dynamic. This occurrence is in contrast to a managed UNC or drive location, which does not require source list management because the embedded literal location is made dynamic through DFS or dynamic drive mapping.

 Using environment variables causes the “dynamic” nature of your source lists to be distributed by being embedded in every package configuration on every computer. Using a managed UNC or drive letter leaves the “dynamic” nature of source lists in the hands of a centrally manageable infrastructure.

---


## **Trickle Services, CD-ROM Distribution, and Source Lists**

Many distribution systems, including SMS 2003 (code-named Topaz), support the ability to trickle a package down to a client slowly over time, then execute it from the local drive. These types of services generally do not attempt to manage the source list for your package, so the first and only source list entry will point to the locally cached copy. Locally cached copies can be deleted by the distribution system on some clean up interval. In some companies, mobile users who are always remote use a CD-ROM for software distribution. In both of these cases, it is important to manage the source lists for installed packages. You can use several approaches to manage the source lists.

One approach is to include a couple of network-based source locations in the SOURCELIST property when launching the package. This approach causes the client to search for network locations first and then for the media it was originally installed from—if the mobile user keeps the media handy, they can use it. If they do not have it, they can connect to the network and attempt the Windows Installer operation again.

It might also be worthwhile to set the SearchOrder policy to *nm* to ensure that Windows Installer searches network sources (*n*) before media sources (*m*). *nmu* is the default behavior for this policy when it is not configured (*u* being URL-based sources), but setting the policy explicitly helps others to know that it is a managed policy value. It should be set high in your OU hierarchy to prevent overriding at lower levels.

Another approach is to clear the source list of all sources, then add only network sources. Doing so forces mobile users to connect to the network to gain access to package files. You might want to do so if you have multiple versions of packages and want to ensure that the mobile user does not attempt to use an old CD-ROM as the source for an installed package. You can implement this setup by using in-house maintenance scripts or the MSICAResetSources script mentioned earlier.


 If you've always wanted to use trickle-type distribution, you might already have it for free! Windows XP and Win2K SP3 include the Background Intelligent Transfer Service (BITS) for supporting trickle down of Windows Update files. BITS works very well—detecting when to resume, limiting bandwidth usage, and so on. By using the BITSADMIN.EXE utility on the Windows XP CD-ROM, you can schedule your own files to deploy from a Web server (using HTTP) to a client through BITS. For more information about how to use BITSADMIN, visit <http://desktopengineer.com/bitsadmin>.

## **Fixing Existing Unmanaged Sources**

Many of you might be feeling a sense of doom in regard to package installation locations because you have not been managing them for the past 2 years that installations have been occurring. You might have thousands of computers that have installed packages from many different locations that might no longer be valid. Or perhaps you are going through a process to consolidate locations or migrate servers—in all of these instances, package source lists might be pointing to incorrect locations.

---

MSISources is a script written for just this purpose. It allows you to identify which packages you want to manage source locations for in a control file. The control file is then used by MSISources to find packages that should be managed and point them to new locations. MSISources allows drive letters or UNC's and allows absolute paths to be used (for example, \\Server\Share\acrobat\version5). In addition, MSISources allows re-rooting of existing source list paths. This feature facilitates easily consolidating or moving existing source locations without manually coding all the paths into the control file. When source lists are re-rooted, each existing source list is retrieved, and the drive letter or UNC is replaced with the new root location.

 MSISources allows flexible remapping of the source lists of multiple installed packages at one time. For example, by running it in the logon script, you could use it to ensure that every installed copy of Visio on every computer in your company has its source list pointed to a managed location. You can download MSISources from <http://windowsinstallertraining.com/msiebook>.

**[Editor's Note:** This content was excerpted from the free eBook *The Definitive Guide to Windows Installer Technology for System Administrators* (Realtimepublishers.com) written by Jeremy Moskowitz and Darwin Sanoy and available at <http://www.wise.com/ebook/>.]