



TIDYING UP APP INSTALLS



IF YOU LONG FOR NEAT APPLICATION INSTALLATIONS,
MSI SOFTWARE REPACKAGERS ARE JUST THE TICKET.
WISE PACKAGE STUDIO BEAT TWO RIVALS IN OUR TESTS

By James E. Drews

» Any sys admin who's tired of messing with proprietary software-installation methods, has probably been eyeing Microsoft Software Installer lately. With MSI, Microsoft is attempting to bring consistency to application installations. That's good for sys admins—it's easy to publish an application in an MSI format out to an Active Directory setup, and many popular desktop-management packages, including Altiris Client Management Suite, Microsoft Systems Management Server and Novell ZENworks, support MSI for software distribution.

That's all well and good for new software, but what about older applications? Although it's best to get the MSI file from the software developer, several new products let you convert legacy software installations to the MSI format. To test them, we invited gInstall Corp., InstallShield Software Corp., OnDemand Software, Wise Solutions and Zero G Software to submit their software-repackaging products for testing at our University of Wisconsin-Madison Real-World Labs®. Unfortunately, Zero G's and gInstall's products turned out to be geared toward performing standalone application

installs and couldn't generate an MSI; see our impressions in "More Install Options" on page 79.

Inside MSI

Within an MSI file are database tables describing the application. In many cases, the MSI file also contains the software files. This gives you some flexibility, for example, with large applications that don't fit on one CD. The MSI comprises three basic parts: product, features and components.

The product is a collection of features that make up the software program to be installed. End users can select the application features they wish to install. Examples of components are registry settings and COM (Component Object Model) objects. Applications with shared components will have the same component in their MSI files.

Before MSI, reference counts in a registry setting kept track of shared components; these counts changed when a shared component was added or removed. With MSI, each product using a shared component has a unique product ID or GUID (global unique identifier)

added to the reference counter for the component. This lets administrators see exactly which applications are using a shared component.

The biggest benefit from MSI is standardization: MSI could make proprietary installation formats history. Here are some other advantages:

» **WIS (Windows Installer Service)** performs application installs transactionally. That is, it keeps track of files and registry settings that are added, replaced or removed. If an unrecoverable error occurs at any time during installation, MSI lets you roll back the entire process to avoid disabling your system with half-installed components.

» **When used with Active Directory**, an MSI can be associated with a group policy to let end users install the application on their systems even if they don't have rights to modify the file system or registry. The request is passed on to WIS for the actual install. Because the package was approved by the administrator using association with the AD group policy, for example, the install will continue.

» **The MSI format lets you easily modify or customize** the software install by creating a transform. An MSI transform is a file (.mst) that describes how WIS should install an MSI package. The transform may indicate which package features to install, provide a custom installation location or designate responses to the user-information screens commonly presented by software installers. Rather than try to modify the original MSI package, it's better to create an MSI transform so that different transforms can be created, all using the same base MSI.

» **Software developers choosing the MSI format** can use advanced MSI capabilities, like self-healing or specifying on-first-access feature installs. These capabilities, however, require the use of MSI programming interfaces, so don't expect this level of functionality from legacy applications that are repackaged into an MSI.

I'll Take It!

All three products we tested require some kind of database. All could use Microsoft SQL server or its run-time component, MSDE, but only OnDemand's WinInstall could use Oracle. However, WinInstall was also the only entry unable to use a Microsoft Access database file and the only one to do the MSDE install and configuration as part of its setup process.

Databases let each product track multiple MSI applications for conflict resolution. This feature let us compare a set of applications and/or the information about components installed on a workstation for possible DLL conflicts, for example. To make things simple, we used an Access file for both Wise Package Studio and InstallShield AdminStudio, and the MSDE database for WinInstall. (For more testing details, see "How We Tested Software Repackagers," page 77.)

All three products did a good job of converting our legacy Mozilla Firefox application to an MSI. They all detected the same set of files and registry settings. However, when the resulting MSI files were run through the products' validation check, both Wise Package Studio

and InstallShield AdminStudio had several errors, whereas WinInstall had none in the resulting MSI. Bear in mind, however, that some errors are more significant than others. Many applications ship with hundreds of errors and warnings. Many of them can be ignored, and the MSI will still work (see www.installshield.com/consulting/services/ice.asp).

AdminStudio's and Wise Package Studio's user interfaces were more polished than WinInstall's. WinInstall was the only package, in contrast, to include a built-in application-distribution capability. The other two require Microsoft SMS, Active Directory or Altiris Client Management to distribute the MSI packages.

We ran our test applications against the sample database of applications each vendor included. Our test applications didn't contain any DLL conflicts. But Wise Package Studio and AdminStudio purposely created DLL conflicts and provided tutorials on how to resolve them. For WinInstall, we performed conflict assessment against two copies of our created Firefox MSI; this resulted in a long list of conflicts for each identical file installed to the same location, but gave us an idea of its conflict-assessment piece.

As more vendors adopt MSI, system admins will want to customize software installs for their systems. Creating

Executive Summary

MSI SOFTWARE REPACKAGERS

Microsoft Software Installer offers system administrators and programmers many benefits, including a standard way to distribute applications, customizable installations, integration with Active Directory and desktop-management packages, and rollback options for failed attempts.

Most newer software supports MSI, but legacy apps are left out in the cold ... unless you have a nifty software repackager that can retrofit MSI capabilities. We invited gInstall Corp., InstallShield Software Corp., OnDemand Software, Wise Solutions and Zero G Software to submit their repackaging products to our University of Wisconsin Real-World Partner Labs for testing. Zero G and gInstall did not meet the MSI requirement, but we did examine their products outside of our test scenario (see "More Install Options," page 79).

We tested the remaining three packages and were quite pleased. All easily converted our legacy Mozilla Firefox application to an MSI and avoided the dreaded DLL conflicts. Still, only one could win, and we chose Wise Package Studio as our Editor's Choice. However, those needing desktop management should consider OnDemand's WinInstall MSI Packager, and InstallShield's AdminStudio can save experienced admins a few bucks.

an MSI transform will be the preferred method. Each product we looked at could create transforms, though WinInstall had a harder time than its rivals. Wise Package Studio and AdminStudio simulated an MSI setup and recorded our selections and responses, but didn't install the product. Wise Package Studio let us hide wizard pages running with the transform. All three products could hide the dialog windows, but AdminStudio and WinInstall required an extra step to do so.

We gave Wise Package Studio our Editor's Choice award. We liked its task-based project interface, and the typical systems admin should find the product easy to use. Moreover, its integration with a number of third-party distribution systems made it stand out from the pack.

Wise Solutions Wise Package Studio Professional Edition Wise Package Studio is a great tool for application repackaging, MSI validation, DLL conflict

A resolution and the creation of standalone application installs.

At the start, Workbench, the program's interface, gives you a set of tasks necessary for completing the setup. The agenda includes installing a wizard to help find file and registry exclusions for repackaging captures; capturing a standard operating system to check for conflicts; and configuring various templates.

Package Studio is the only product we tested to provide a wizard for finding standard file and registry exclusions. The wizard let us determine the appropriate capture settings for each software-install snapshot. In addition, it helped us build an exclusion list, look for deleted files in the capture and decide which directories to watch.

Building the exclusion list involved Package Studio performing a snapshot of the system and having us run the applications, such as Internet Explorer, Regedit or Notepad, we typically open during a software install. Then it rebooted the system to catch any file and registry changes and exclude them from future software-install captures. We liked this feature, and wish the other two packages offered something similar.



MSI or transforms were created as projects in Workbench. For each project, we selected a process, or set of tasks to perform in the project. We were most interested in "Customize MSI Using Transform" and "Repackage for Windows Installer."

Choosing a process populated the project with a set of tasks we should perform to create the MSI or transform. Each task had a checkbox to indicate if it was completed and a "run" link that would launch the appropriate application. For example, the "Create Transform" task's run link launched the InstallTailor program, which stepped us through creating the initial transform.

Package Studio let us select from a number of methods to create an application snapshot. It has a conventional snapshot process, which scans and records the state of the system before and after the application install and picks out any change, and a SmartMonitor that tracks what the install does at a low API level. Or, we could take a "virtual OS" snapshot of a clean system, then simulate the install.

First we tried the standard snapshot method with the addition of SmartMonitor. This resulted in an MSI package that failed to run. The problem was that the

REAL-WORLD LABS [®] REPORT CARD		Software Repackagers		
	Wise Solutions Wise Package Studio Professional Edition 5.1	InstallShield Software AdminStudio 5.5 Professional	OnDemand Software WinInstall MSI Packager Professional Edition & WinInstall 8	
EASE OF MSI CREATION				
Creating an MSI transform (15%)	5	4.5	4	
Creating MSI manually (15%)	4.5	4	3.5	
FEATURES (5%)	4.5	4.5	5	
ADDITIONAL FEATURES				
Accuracy of capture (20%)	5	5	4.5	
Creating valid MSI files from capture (20%)	4	4.5	5	
Ease of capturing install (20%)	5	5	5	
PRICE (5%)	3.5	4	4.5	
TOTAL SCORE (100%)	4.63	4.60	4.50	
<small>A≥4.3, B≥3.5, C≥2.5, D≥1.5, F<1.5 A-C GRADES INCLUDE + OR - IN THEIR RANGES. TOTAL SCORES AND WEIGHTED SCORES ARE BASED ON A SCALE OF 0-5.</small>	A	A	A-	
EASE OF MSI CREATION applies not only to MSIs, but also to MSI transforms, which are files that let admins easily modify or customize the software install. For ease of creating an MSI by hand, we rated how simple it was to create an MSI manually.				
FEATURES include wizards, integration with DM suites, conflict management and breadth of database support.				
Customize the results of this report card using the Interactive Report Card [®] , a Java applet, at www.nwc.com .				

setup.exe for Firefox had multiplatform elements—it attempted to create the *bin/components* directory (note the forward slash). This failed, but SmartMonitor picked up the directory-creation attempt and figured it had to be included. Because this didn't work, we redid the snapshot, which didn't use SmartMonitor the second time around. The result: a working MSI.

In examining the MSI, we didn't find any extra registry entries. We also didn't notice any missing items when compared with what was picked up by the other repackaging programs. However, the MSI had the largest number of errors and warnings—a total of seven errors and one warning when using the included Package Validation program. Most errors arose from creating per-machine and per-user settings in the same install. That is, there were file components put on the machine for all users, and shortcuts created for the user running the install.

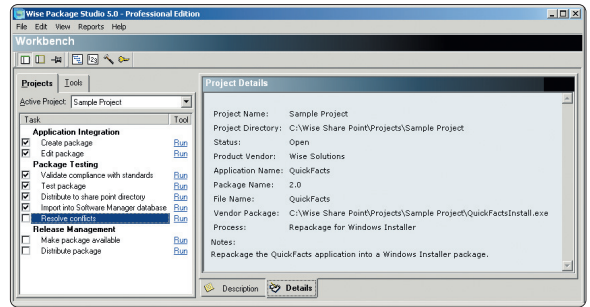
For errors that are easy to fix, the validation program provides a "correct" button. Unfortunately, none of the glitches we encountered fell in that category. Ironing out those kinks would have required us to suck it up and edit the MSI ourselves.

Using Wise's InstallTailor application, we had no trouble creating a transform for the ActiveState Perl MSI, which we could further customize. One unique feature is a small checkbox on each page of the simulated install to indicate that the page should not be presented to the user when running the install with the transform. The other vendor packages also let us mark pages as hidden, but had to do that as a separate step.

Another handy component of Package Studio is the Conflict Manager. This lets administrators compare workstation snapshot and/or application installs to check for DLL conflicts *before* the application gets deployed to end users. We ran the Conflict Manager and compared our Firefox install with the included sample database. No conflicts there.

When putting together a package for our custom Atropos application, we used the Windows Install Editor to create an MSI. Had our application been written in Visual Basic, C# or J#, the Install Editor could have imported settings from the Visual Studio project to jump-start MSI creation. But because our app was written in C++, we had to select the standard template and create the package by hand. Fortunately, the creation process was painless using Installation Expert's high-level editing screens, which made it easy to put in all the file, registry and shortcuts we needed and select and customize a standard set of dialog windows—like welcome and license screens—that would be presented to the user during install. Customizing each screen was a breeze.

Wise Package Studio includes a deployment wizard to help take the newly created MSI and put it into one of several popular distribution systems, including LAN-Desk, Microsoft SMS and Novell ZENworks. The wizard detected that we had ZENworks installed on our system and enabled that deployment option. When selected, it



The Wise Package Studio Workbench guided us through the process of creating app installs. And the install editor—with Installation Expert—made it easy to modify MSI packages.

prompted us for the NDS tree, context and application object name to create. It then created the ZENworks application object in our NDS tree with the settings needed to deploy our MSI application.

Overall, Wise Package Studio is a solid application repackager. It has the high-level package creation/editing tools needed to create and modify MSI packages automatically, but it also lets experienced software developers who know the ins and outs of MSI get their hands dirty. And it can be used to create standalone application installers (.exe). Our only gripe is the errors in the created MSIs when run against the validation tests.

Wise Package Studio Professional Edition 5.1, \$4,599 for a single-user license; one year of support and upgrade protection included. Wise Solutions, a wholly owned subsidiary of Altiris. (800) 554-8565, (734) 456-2100. www.wise.com

HOW WE TESTED SOFTWARE REPACKAGERS

We installed the software packages onto identical Windows 2000 Professional VMWare images and used each package to create two MSI installs and a transform for a third MSI package. In the snapshot process, we used the default set of file and registry excludes provided by the vendor. We used the following software packages:

- » Atropos, a custom in-house application used at the University of Wisconsin-Madison
- » The Mozilla Firefox 0.8 Web browser setup.exe
- » The ActiveState Perl MSI 5.8.3.809

We evaluated each of the software-repackaging applications on how easy it was to create the MSIs and transforms. We also examined how much information was captured in the snapshot process of the Firefox install and compared each of the MSIs to determine if any added extra registry settings and files to the MSI package. As for conflict resolution, two packages had demo walk-throughs in which they created conflicts, so we did see conflict finding in action.

InstallShield AdminStudio Like rivals, AdminStudio requires the use of a database; for our tests, we chose



Microsoft Access. To create projects for transforms, repackaging applications, identifying conflicts and package evaluation, AdminStudio gave us a series of tasks. In comparison, AdminStudio gave us 28 tasks to look at, while Wise Package Studio gave us four. AdminStudio's list was more detailed, of course, containing items an administrator should check, such as confirming that the application isn't installed on a workstation. Although all these tasks may seem like overkill, it's often a good thing to have reminders.

When creating the MSI transform for our ActiveState Perl MSI, AdminStudio simulated install and recorded the responses entered in each step, like Package Studio. To get the simulated install, however, we had to remember to select the Response Transform checkbox in the InstallShield Tuner application.

One thing the Tuner app has that Wise doesn't is a pre- and post-validation utility. The prevalidation check let us know if the MSI for which we were about to create a transform had any problems we needed to fix. Our Perl MSI generated three warnings and one error. The error dealt with version-checking of the package. We didn't run into any problems creating the transform, and had the same three warnings and one error in the post-validation.

We used the Repackager tool to capture our Firefox install. Repackager offered us two options: the conventional Snapshot method and an Installation Monitoring

WebLinks

▶ **"Desktop Management Enlightenment,"**
www.nwc.com/showArticle.jhtml?articleID=15000598

▶ **Most popular MSI downloads,** www.microsoft.com/downloads/results.aspx?productID=&freetext=msi&DisplayLang=en

approach. We took the Installation Monitoring route and found it an easy ride. The monitoring program located all our files, shortcuts and registry entries except one. Our missing entry turned up on the default exclude list, as it was a pointer for the applications uninstaller.

MSI editing and verification is done using the included DevStudio application. When we ran the newly created MSI file through the verification tools, we found one error and one warning. The MSI did work as expected, however, when installed on a clean workstation.

We used DevStudio to create an MSI for our custom application. Like Wise Package Studio, DevStudio has a number of templates with which to start new MSIs, including importing settings from a Visual Basic or C# project. We went with the basic template and used DevStudio's Project Assistant to walk us through creating our MSI. We provided the basic application information, such as name of the company and application name; the platforms we were targeting; and the files, registry and shortcuts we wanted to create. Piece of cake.

AdminStudio includes a ConflictSolver application that will take a set of MSI packages, check for potential

Software Repackager Features

	InstallShield Software AdminStudio 5.5 Professional	OnDemand Software WinInstall MSI Packager Professional Edition & WinInstall 8	Wise Solutions Wise Package Studio Professional Edition 5.1
MSI packaging			
Snapshot conversion of legacy Installs	Y	Y	Y
Other snapshot methods	Y (Installation Monitor)	N	Y (SmartMonitor)
Create a baseline set of snapshot excludes	Manually	Manually	With wizard
Create transforms	Y	Y	Y
Edit existing MSI files	Y	Y	Y
Edit raw MSI tables	Y	Y	Y
Easy editing interface (not raw tables)	Y	Y	Y
MSI package-validation checking	Y	Y	Y
MSI Conflict Management			
Compare multiple MSI applications	Y	Y	Y
Include OS baseline snapshot	Y	Y	Y
Deployment wizard	Altiris Client Management Suite; LANDesk Management Suite; Marimba Client Management; Microsoft Active Directory, Systems Management Server; Tivoli	Microsoft Active Directory, Systems Management Server; Novell ZENworks; OnDemand WinInstall	Altiris Client Management Suite; LANDesk Management Suite; Microsoft Active Directory, Systems Management Server; NetSupport NetInstall; Novadigm Radia; Novell ZENworks; OnCommand MSI Package Wizard; Tivoli
Import projects supported	InstallShield; Microsoft SMS; Novell ZENworks; WinInstall; Wise WSE	Novell ZENworks; WinInstall	InstallShield; Microsoft SMS; Novell ZENworks; WinInstall; Wise WSE
Database support	Microsoft Access, SQL/MSDE	Microsoft SQL/MSDE; Oracle	Microsoft Access, SQL/MSDE

Y=Yes, N=No

conflicts and help resolve them. We ran our Firefox install against the included set of sample applications and didn't find any problems. We also liked the Distribution Wizard, which helps deploy MSIs to distribution systems. Unfortunately, ZENworks wasn't supported (though it should be in the next release), so we were unable to test this feature.

We liked this product, but it lagged behind Wise Package Studio in ease of use. On the other hand, the price was lower. If you have the chops, AdminStudio is worth considering.

AdminStudio 5.5 Professional, \$3,299. InstallShield Software Corp., (800) 809-5659, (847) 466-6000. www.installshield.com

OnDemand Software WinInstall MSI Packager Professional Edition WinInstall goes beyond software repackaging to include an entire desktop-management

A system. It let us distribute our applications to workstations, track inventory and run reports. These are all great features that the other two products don't include, but they're also beyond the scope of this review.

Like the other packages we tested, WinInstall gave us a single console from which we could perform all administrative tasks. And most of the tasks were performed from the console, not from separate programs launched from the console (the tack the other two programs take). The notable exception to this was the Discover tool, which takes the application snapshot. The downside to the WinInstall console is that its interface isn't as polished as those of the other packages. For example, for examining files in the MSI package, it provided a simple Windows multicolumn list box showing all the files and the directory into which each would be installed. We preferred the other packages' tree-list windows. And unlike its rivals, WinInstall has no Smart Monitoring tools.

An application install is converted to an MSI using the Discover wizard. This wizard performs the conventional before/after snapshot of the system. One nice feature is that the "before" scan can be saved and reused. For example, if our "clean machine" was created with a Ghost image, we could run the discover process once and save it for later use instead of having to wait for the "before" scan to complete again. Although WinInstall can't revert the machine back to the saved state, most administrators will just re-Ghost the machine or revert the VMWare image.

When we did our Firefox application snapshot, WinInstall missed capturing the Quick Launch icon. However, when it came to creating a valid MSI file, WinInstall was the only repackaging program to be error-free with our Firefox application test. It did issue five warnings, but these were minor.

When creating an MSI transform, we weren't offered the option of a simulated install. Instead, WinInstall watched our modifications to the MSI package. It didn't

make any of the changes to the package, but created the transform file instead. Compared with the other two packages, it took a bit more effort on our part to create the transform, but the end result was the same—a working transform file.

WinInstall also provides tools for MSI package validation and conflict resolution. Like the other two packages, an OS baseline snapshot can be imported into the database and included in checking for any possible problems before deploying the application. We imported our base OS and Firefox MSI and didn't find any conflicts.

Overall, WinInstall is a good, inexpensive app-repackaging system. The user interface could use work, but WinInstall could be a perfect fit if you're in need of a desktop-management system.

WinInstall MSI Packager Professional Edition: OnDemand Software, \$2,500; WinInstall 8 license, \$2,000 for 100 nodes. (866) 495-0541, (239) 495-0541. www.ondemandsoftware.com **NWC**



JAMES E. DREWS is a network administrator for the CAE Center of the University of Wisconsin-Madison. Write to him at jdrews@nwc.com. Post a comment or question on this story at www.nwc.com/go/ask.html.

MORE INSTALL OPTIONS

Looking for a more general-purpose application-install creator? We didn't include products from Zero G Software and gInstall in our review because they do not create MSI-based installs, but we did find them interesting. Both are geared toward creating standalone application installations.

» Zero G Software InstallAnywhere; www.zerog.com/
If you need to create application installers for multiplatform systems—where your definition of *multiplatform* is more than just multiple versions of Windows—InstallAnywhere fits the bill. This Java-based package supports a wide variety of platforms, including Windows, Hewlett-Packard HP-UX, IBM AIX, Linux, Mac OS, Novell NetWare and Sun Solaris.

» gInstall Ghost Installer; www.ginstall.com/
Ghost Installer was designed to create application installers for Windows (creating MSI installs is on the company's road map). Ghost Installer includes wizards for importing more than just .Net projects—it also imports C and C++ projects. In addition, it can protect the application install with an RSA public/private key, a feature not found in the installation-generation products we tested. Each customer is given a unique key to decrypt the installer, which can be traced back to the legal owner if necessary.