InstallAware 2005 Reviewer's Guide

The Scripting Aware, Web Aware, MSI Aware

Installation Solution

Produced for InstallAware Software Corporation by Sinan Karaca

March 2005

INSTALLAWARE

Contents

Overview

InstallAware©
2005



Overview

Welcome to the InstallAware 2005 Reviewer's Guide. This document will familiarize you with InstallAware 2005, the newest version of InstallAware Software Corporation's flagship setup development environment, culminating more than nine years of technological innovation.

The InstallAware 2005 Reviewer's Guide is organized into two parts. In this first part, the Overview, you will find a general introduction to InstallAware 2005.

The second part of this guide takes you on a tour of InstallAware 2005. It is organized by the major areas of installation development and support in InstallAware 2005, providing you with an overview of each area, and is followed by a description of the many updates, enhancements, and additions introduced in this release. If you are already an InstallAware enthusiast, you may want to quickly scan the overview section, concentrating instead on the updates that make this the most important version of InstallAware yet.

InstallAware: Advancing the Art of Setup Authoring

InstallAware's legacy began with a simple vision: to make setup authoring simple and enjoyable again. Since the debut of Windows Installer in 1999, setup authoring has become an incredibly complex endeavor. Developers are forced to work within the constraints of a relational database structure when building their setups, and they can no longer use branching script flow to direct their setup logic. Even creating a very simple setup that installs a single file entails populating multiple interconnected database tables. While a large variety of third party setup authoring tools have emerged since 1999, automating or visualizing most of the tedious work, Windows Installer still remains a setup development platform that is very hard to develop for and customize.

InstallAware has been designed from the ground up to solve today's setup authoring challenges. InstallAware targets the Windows Installer platform, because despite its being hard to use, Windows Installer is rapidly becoming the corporate standard worldwide for developing and deploying application installations, and offers unprecedented features. Therefore, instead of ignoring Windows Installer, InstallAware embraces and extends Windows Installer and strives to make it more accessible. With InstallAware, you have everything you need to create modern setups that address today's development needs:

- Genuine Scripting for Windows Installer: InstallAware introduces branching, script-driven execution logic into Windows Installer. Yet, InstallAware does this without requiring any external scripting engine or language that needs to be installed as a scripting runtime before-hand. InstallAware generates 100% pure MSI installations while also allowing them to execute like the traditional setup scripts used before 1999. This provides developers with unprecedented levels of flexibility in coding their setup logic. Such flexibility was unavailable since the initial launch of Windows Installer in 1999.
- Web Deployment with Web Media Blocks: InstallAware allows splitting the data contained inside a setup package into multiple online and offline parts. While web deployment traditionally involves putting the entire setup online, thus mandating an active web connection to install any part of the product, InstallAware sidesteps this artificial limitation and allows developers to precisely choose which parts of their setup are to be downloaded, and which are to be provided with the main setup program. This saves time and bandwidth not only for end users installing the applications, but also for companies hosting the downloads. Application runtimes which may already be present on end user systems, and typically bloat setup downloads, are ideal candidates to benefit from the web media blocks system. With web media blocks, they will only be downloaded if required, and the main setup is wholly self-contained, capable of executing an installation without requiring an active Internet connection.
- Full Support for all Windows Installer Standards: Every setup that is generated by InstallAware is in full compliance with all Windows Installer standards and provides uncompromised support for all Windows Installer features. If you are already familiar with Windows Installer, you will recognize such features as install-on-demand, self-healing setups, advertising, binary differential patching, merge modules, silent/command-line installations, and so on. InstallAware fully supports each of these Windows Installer features while continuing to extend Windows Installer to add its own.

What developers love about InstallAware is that they can rapidly develop a setup without having to worry about any of the internals of Windows Installer. They may use either of the visual/code views to build their setup and fine tune its operation, and when the setup is compiled, the InstallAware compiler will ‘magically’ create a fully logo compliant, legal MSI database that can install applications on any number of systems without any runtime/scripting requirements whatsoever.

What Developers Say About InstallAware:

I'm continually impressed by the depth of power that InstallAware offers me. I know absolutely nothing about creating Windows MSI based installers, but with InstallAware I don't need to. It's easy to quickly get up and running with InstallAware, but there is so much more to explore once you start getting more adventurous with your scripts. And the support for custom dialogs is truly awesome! *David Clegg, eBetOnline*

InstallAware has enabled me to fully support Windows Installer, yet remain in full control of what's going on. InstallAware's Web Media Blocks help to ensure that my customers only download what's necessary, and the excellent compression ensures that it's small as possible. I take quality very seriously, and I've been impressed with what I've seen. *Troy Wolbrink, TntWare*

InstallAware is a one-of-a-kind product. I searched high and low for a solution to provide the ability to download and install runtimes from the Internet only if they are needed. InstallAware was the only program I could find that offered this for all the runtimes needed. Add to that the dialog editing/customization, and all the other great features, InstallAware is the best solution available to my knowledge. *Chris Hoffman, Mystik Media*

A first class Installer using the Microsoft Windows Installer Engine. You are not limited to standard MSI actions; you can even author your own plug-ins. Combined with an extensive selection of pre-built themed installers, or even the option to totally construct your own, InstallAware becomes one of the most flexible packages out in the market today. *Andrew Neillans, ABCC Computers*

InstallAware is the best MSI based installer available at this time. It will quickly handle small jobs but will grow to meet the more complex requirements of larger jobs. As software developers we really like the ability to drop down to the InstallAware scripting level which is still quite easy to use yet flexible enough for our complex tasks. Most MSI based installers that we looked at were difficult to use because of a lack of ability to drop down to a scripting style model and were expensive to boot. Our applications that are released today and in the future are dependent upon new Microsoft technologies such as the .NET framework, MDAC and many others. It's nice to have the benefits of an MSI based installer to make sure these pieces are installed while still providing enough power and flexibility to complete the main job of installing our product. *Jim Gunkel, Nevrona Systems*

The remaining sections of this guide are organized into related topics associated with installation authoring. Each section begins with a general overview, and then continues with a description of the new and enhanced features introduced in InstallAware 2005.

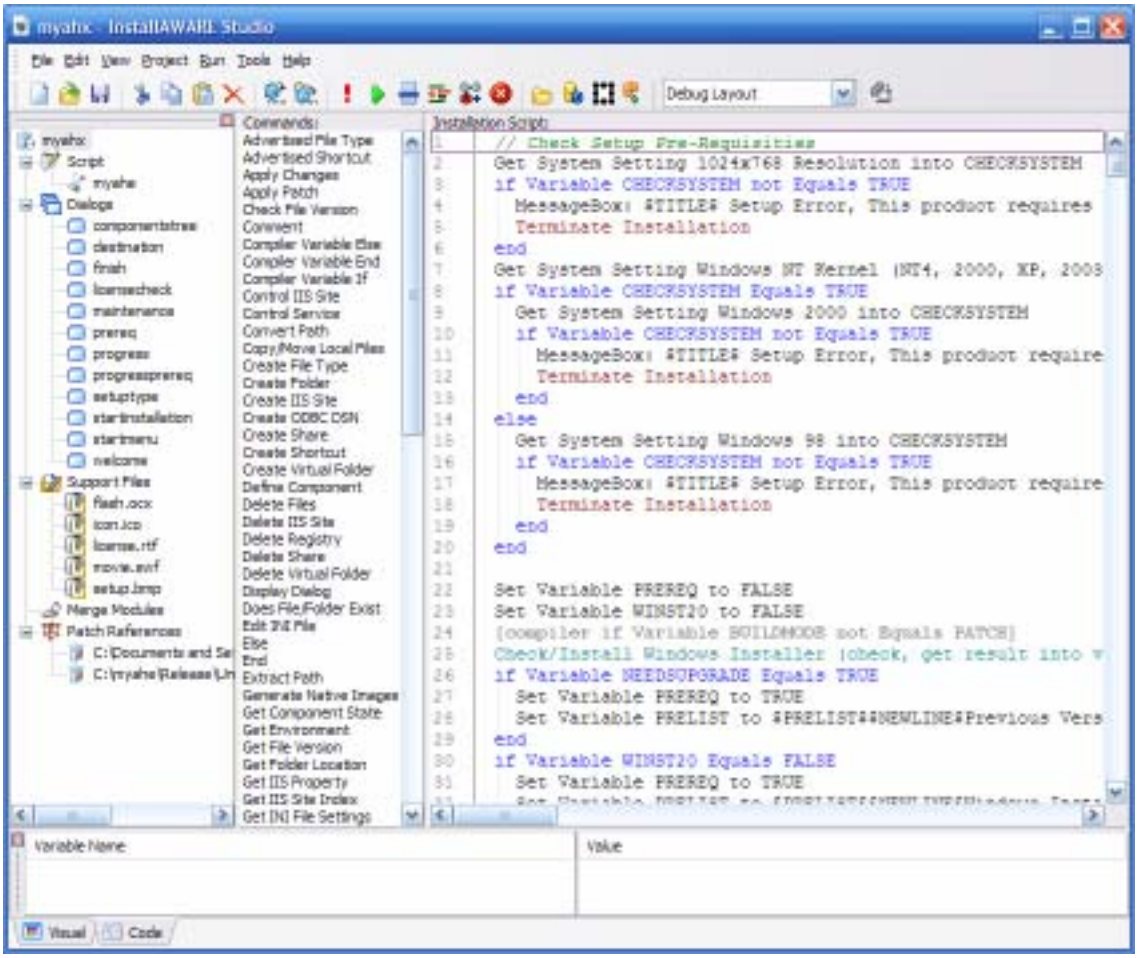
The InstallAware 2005 IDE

InstallAware©
2005



The Integrated Development Environment

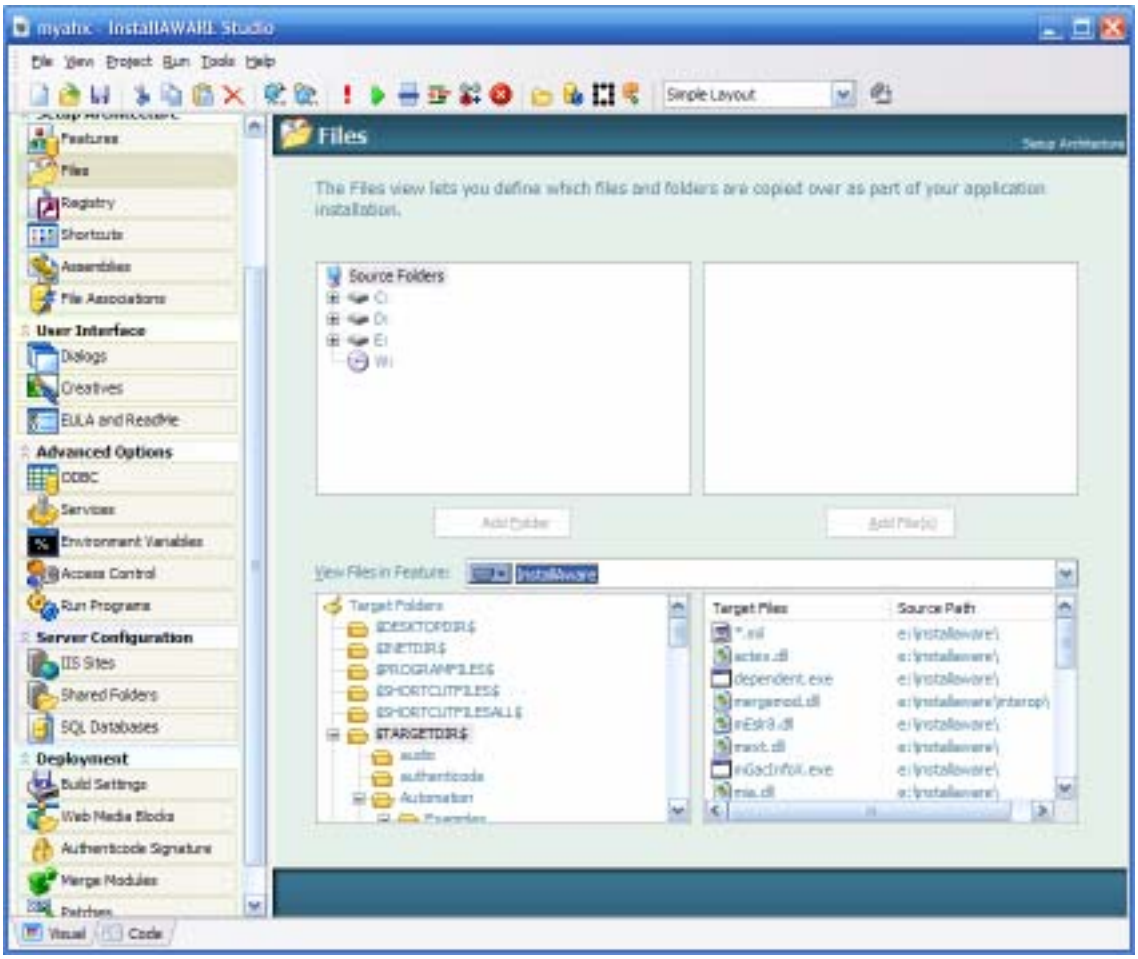
The InstallAware 2005 IDE (integrated development environment) represents state-of-the-art in setup authoring tools. Fast, clean, and visually appealing, InstallAware 2005's IDE enables you to develop setups faster and better.



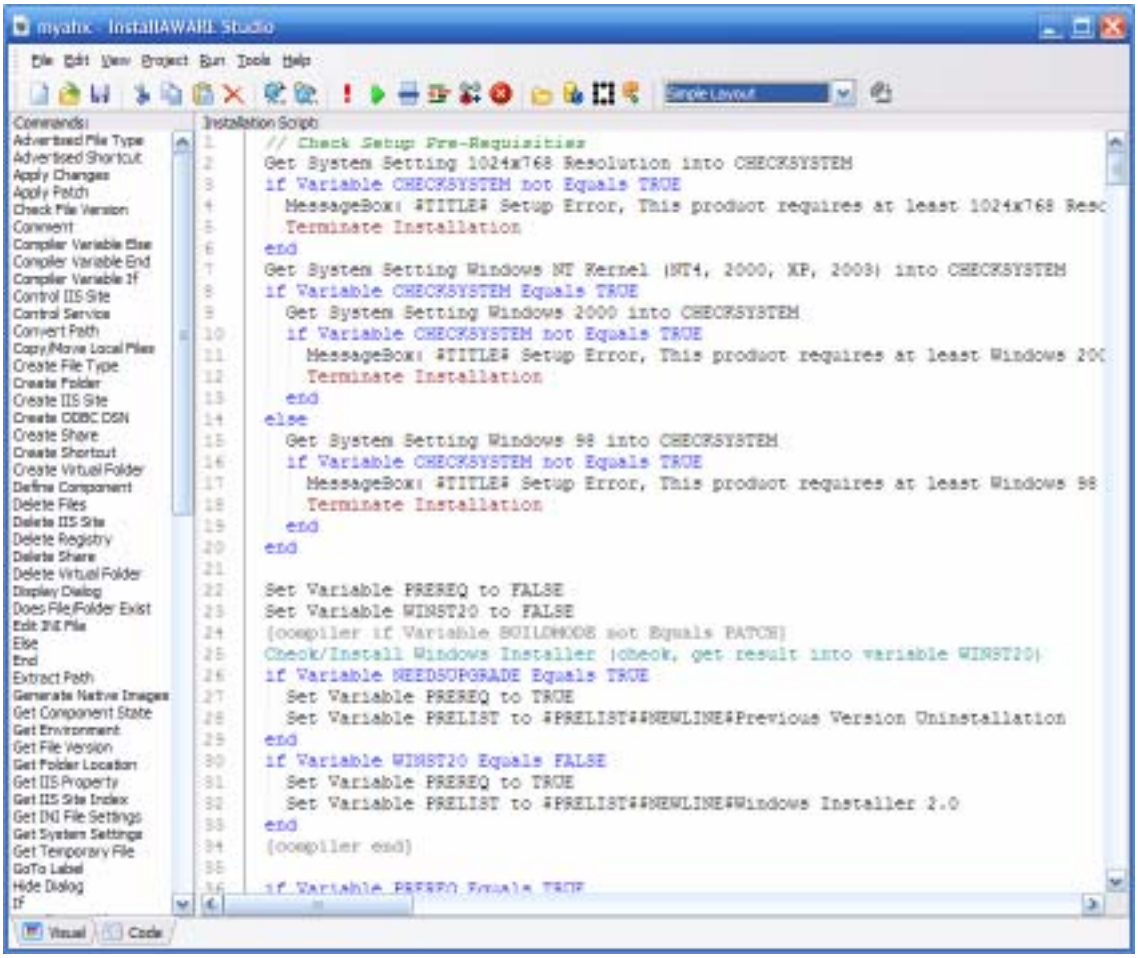
This section focuses on the features found in the various panels, designers, dialog boxes, and views of the IDE.

One IDE, Multiple Views

Whether you are coding a complex setup with highly customized scripting logic, or a simple one that does not require any tweaking, InstallAware 2005's IDE provides you with a consistent and powerful set of views designed to increase your productivity.



When you first open the IDE, it will launch in the visual view. The visual view provides a purely visual representation of your underlying setup script. As you make changes visually in the visual view, the IDE will automatically update the underlying scripting code for you – transparently and seamlessly, without you writing a single line of code. The many pages available on the visual view enable rapid setup development and eliminate the need to manually edit your setup scripts in most cases.



The code view offers unfettered access to the heart of your installation. One of the strengths of InstallAware is that nothing is hard-coded in the setups it creates. You have complete control over your setup program, and the code view is the place to exercise that control. What's better, you do not need to memorize a new scripting language, its syntax, or command set either! Simply drag-drop commands from the command list on the left to the installation script on the right. Your commands will be inserted into the script at their drop point.

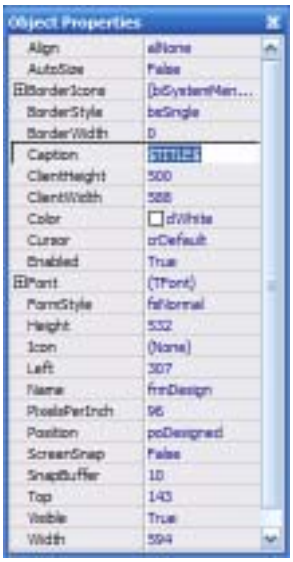
To edit the parameters of a command, just double-click it. A dialog box pops up with the switches that are available for that command, and if you are unsure – help is just one key press away. Press F1 at any time for context sensitive help anywhere in the InstallAware 2005 IDE. Of course, the code view offers all the conveniences one is used to find in traditional source code editors such as cut/copy/paste/delete, find, commenting in/out blocks of code, moving around blocks of code, and so on.

As you work in the code view, whenever you switch back to the visual view, the visual view will properly render the latest changes you have made to your script. The visual view does a great job of keeping up with your modifications even in highly custom scripts – bringing you true two-way integrated setup editing that saves you time and effort.

Dialog Designer

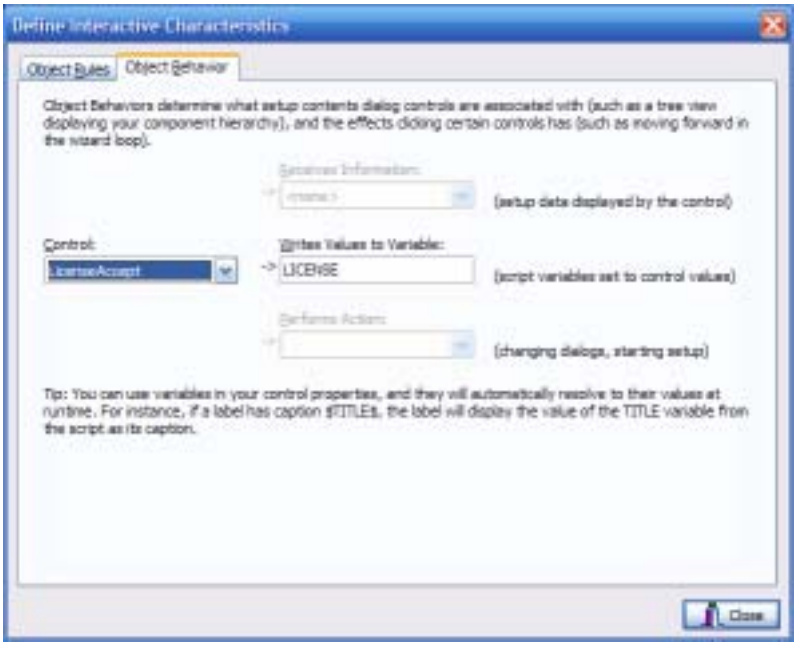
The InstallAware 2005 IDE ships with a state-of-the-art dialog designer. Using the dialog designer you can completely customize all aspects of your setup windows, and even design new ones from scratch. InstallAware 2005 ships with a pre-built collection of twelve setup themes which you can use out-of-the-box, or as a starting point for own customizations. You can of course create and share your own themes as well.

The component palette on the dialog designer will especially seem familiar to developers who have used visual programming languages such as Visual Basic and Delphi. Simply select a control on one of the palette pages and then click on the window being designed to insert an instance of that control. The dialog designer offers a very wide variety of dialog controls, including unique items that are not available with any other setup development environment: flash containers for displaying interactive flash tutorials/movies/animations during installation progress, html containers for rendering sophisticated html content, visual browsers styled after native Explorer tree/list/combo controls for quick and easy selection of files and folders, and so on.



The Object Properties window lets you view and customize the many published properties for the controls that you place on your dialogs. Set text captions, load/save bitmaps, choose animations, set visibility, and do more in this window.

Having controls on your windows interact with one another is very easy. Just double-click any control to bring up the Object Rules editor. This editor lets you visually configure how controls respond to state changes of other controls on the same window. For instance, if a license checkbox is unchecked, you can disable the Next button on the window, to prevent users who do not accept your license agreement from installing your product.

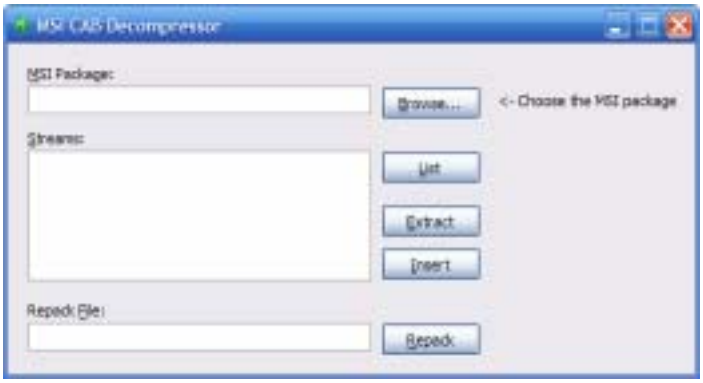


Getting/setting values of variables between the script and dialog controls is also very easy. Choose the Object Behavior tab after double-clicking any control, and simply type in the name of the script variable that should get/set the state of the control on the window. The Object Behavior tab also lets you configure controls to receive pertinent setup information, such as progress percentage, textual description of the current installation task, and so on.

Rich Collection of Tools

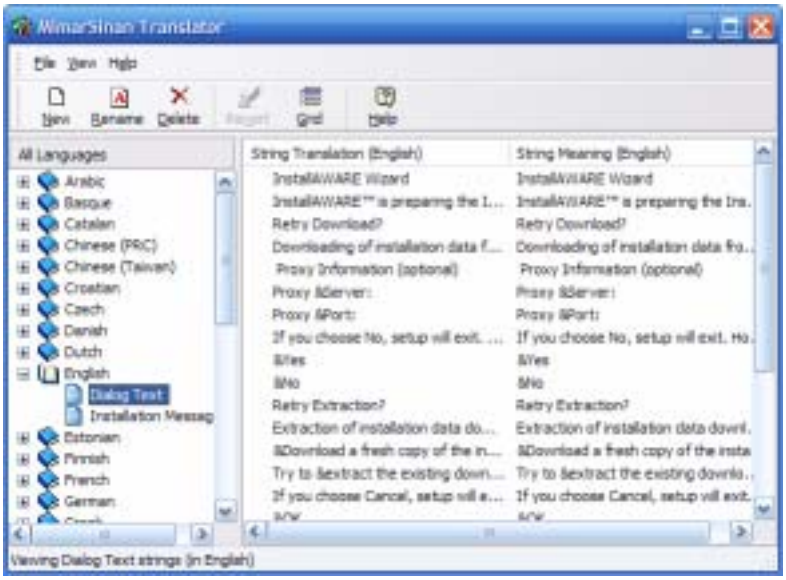
The InstallAware 2005 IDE is complemented by a rich array of tools which assist with everyday setup authoring tasks, helping automate tedious chores and performing other useful functions.

Using the Form Batch Update Tool, you can update multiple dialogs in a single pass. You may replace a particular kind of a control used on a dialog with another kind, or update the character set used for fonts on your dialogs.



The MSI CAB Decompressor tool lets you increase the compressibility of external Windows Installer databases that will be included with your setup package, such as merge modules (MSM files) or installation databases (MSI files). While InstallAware 2005's compression works miracles in reducing runtime and application sizes, it cannot recompress pre-compressed data. This tool processes Windows Installer databases and retrieves a list of compressed cabinet streams which can then be extracted from the database, repacked with zero compression, and then reinserted back into the original database. This procedure assures you will get the smallest possible file size for your installation programs.

The stand alone code signing tool lets you sign your executables using Authenticode technology, and a software publishing certificate/key pair. Authenticode code signing validates and protects the integrity of your installation executables and provides information on the publisher. This information assures end users that the program they are about to install on their PC has not been tampered with, and is also required by recent security additions in Windows XP Service Pack 2 to provide a pleasant user experience. Of course, the InstallAware 2005 IDE itself can also code sign the setups it creates directly at build time.

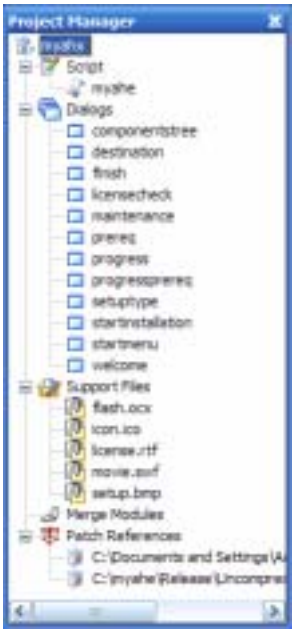


The translator is your repository of installation messages used by InstallAware 2005 and provides a convenient location to visually store your localized setup strings.

Desktop Layouts

The InstallAware 2005 IDE offers the capability to save/load desktop layouts under different names, and also remembers the different layouts associated with each of the visual and code views. For instance, you could dock the variable watches window to the main IDE window, call it the debug layout, and use that layout whenever working in the code view.

The Watches window lets you view the state of setup variables during installation, and also override the values of variables. Watches are a very helpful debugging aid.

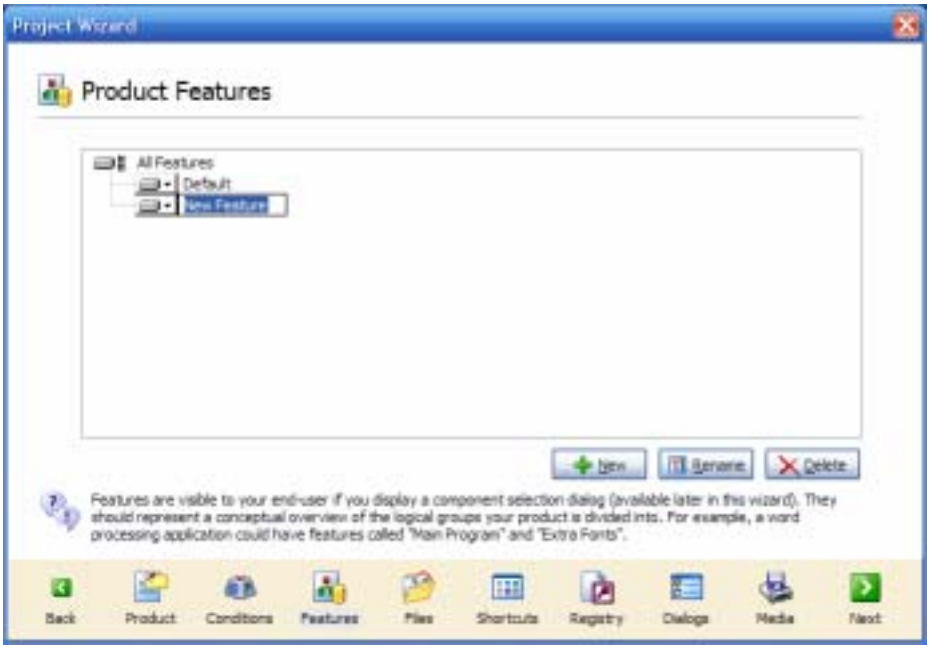


The Project Manager window hierarchically displays the files that are part of your setup project, including your setup script, wizard dialogs, support files, merge modules, and references to other installers for patch creation. You can rapidly add/remove/modify any of the files that are a part of your project here. Docking the Project Manager to the main IDE window will also be helpful when working in the code view.

InstallAware 2005 Wizards and Templates

Wizards are small applets that help you to quickly create the projects, dialogs, and files that you use in InstallAware 2005. Templates are pre-built setup projects ready to be customized further. The InstallAware 2005 IDE comes with a variety of wizards and templates to help you get started faster with creating your setups.

Project Wizard



The Project Wizard guides you step-by-step, asking you specific questions related to your setup when creating a new installation project. Setups created by the Project Wizard have all the scripting logic in place for one-click patch creation, web media blocks, pre-installing various application runtimes, advertised installs, auto upgrades, and more advanced features.

PackageAware

InstallAware 2005 includes a new wizard, called PackageAware. PackageAware is unique in that it may be used to convert legacy installations, built without InstallAware and/or Windows Installer technology, to a Windows Installer installation powered by InstallAware technologies. PackageAware works by scanning the system before and after an application is installed. It then compares those two scans and encapsulates the delta in a ready-to-use InstallAware installation project, complete with a script that you may further customize as necessary. PackageAware provides an effortless migration path to Windows Installer/InstallAware and also a clever way to monitor changes made to the system by installations, or other types of software applications.

Templates

Several templates provide baseline points for your setup projects:

- Blank Script: Creates an empty project with an empty script. Useful for highly custom scripting actions.
- Win32 Script: Creates a pre-built setup for a Win32 application. Comes complete with a splash screen, support for pre-installing the Windows Installer runtime, example read-me and license files, as well as HTML progress billboards. Also supports one-click patching, advertised installs, and auto upgrades.

- .NET Script: Creates a pre-built setup for a .NET application. In addition to the features in the Win32 template, offers support for pre-installing the .NET runtime.
- Java Script: Creates a pre-built setup for a Java application. Offers support for pre-installing the Java Runtime. Note that Java Scripts may only install Java applications onto Windows platforms.

Plug-Ins

Plug-Ins offer ways to seamlessly extend the InstallAware 2005 IDE. Each plug-in has full access to the state of the installation, and can modify the values of variables used in the setup script. Plug-ins appear as an integral part of the setup script and can render their script line dynamically based on the chosen switches. Several components of the InstallAware 2005 setup development framework are implemented as plug-ins and you may also develop your own plug-ins.

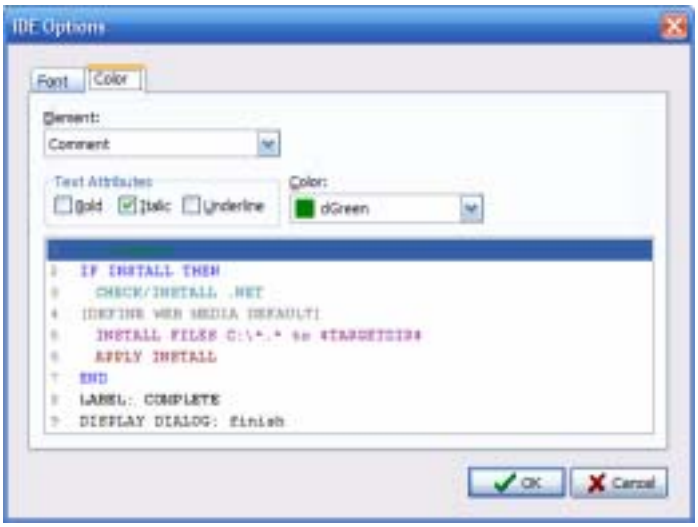
Any language capable of developing Win32 DLLs may be used for plug-in development. If the generated DLLs have runtime requirements, be sure to include the necessary runtime files directly with the plug-in DLLs. Two template projects – Visual C++ Plug-In and Delphi Plug-In – are available. These templates already declare all required DLL exports and are ready to be populated with custom behavior.

Refactoring

When more than one developer is working on a setup project, or setup projects are moved between machines, it often becomes necessary to change the source paths referenced by the project.

The Refactor Paths tool offers a very convenient mechanism to update embedded paths used in your setup script. The tool intelligently analyzes your setup script and extracts all hard-coded path names. It then obtains the common root paths and displays them for you to modify in a window. In the InstallAware 2005 IDE, the Refactor Paths tool can also refactor URLs.

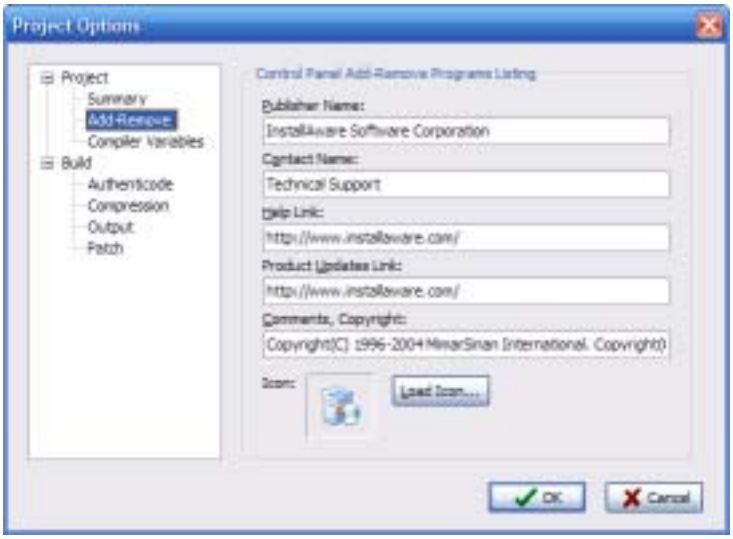
IDE Options



The IDE Options window lets you customize the font used for the script editor. You may also change the syntax highlighting used for the different types of scripting commands that are available in the InstallAware code view. For instance, by default, compiler directives are highlighted in gray, whereas scripting commands are displayed in black.

Project Options

The Project Options window lets you customize your project settings and build options.



Project settings include the name of your project, how your application appears in the Control Panel Add-Remove programs applet, the summary database information that is recorded directly into the MSI installation database, the compiler variables and their default values, the custom setup icon, and more.

Build options include the output folder and default file name for your setup, the package deployment mode (uncompressed, compressed, web), authenticode code signing settings, a list of the previous installers of your product for one-click patching, and the strength of compression that is applied to setup packages.

Scripting Aware

InstallAware©
2005



Genuine Scripting for Windows Installer

One of the major strengths of InstallAware 2005 is the unique architecture it employs for creating Windows Installer packages.

While all other setup tools for Windows Installer provide high level visual editors for creating Windows Installer packages, none of them offer a native scripting capability for Windows Installer. Any scripting that is offered either requires an external scripting engine, which needs to be pre-installed and thus adds an extra layer of complexity to setups; or executes directly through another scripting technology which does not require to be pre-installed before use.

Even though this approach is suitable, it bifurcates installation projects into two disparate sets – one that goes through Windows Installer, and another that goes through the third party scripting engine (which may or may not need to be pre-installed). Separate editors and environments must be mastered for these two independent aspects of setup projects, and this increases the burden on the developer. It also complicates the development process as the two separate parts can interact with each other in unexpected ways and have unexpected dependencies. InstallAware has addressed this problem since its very first release in a conceptually clean, technically sound manner.

Types of Scripting Commands

InstallAware 2005 presents a single, unified setup script which effectively deals with the bifurcation problem. The setup script is the heart of the installation and has complete control over what happens – and when.

The script contains several different classes of commands:

- Comments: As found in all programming environments, comments are hints for the developer.
- Flow Control: Commands that direct program flow into alternate branches based on the evaluation of conditions.
- Plug-In: Custom plug-in provided commands.
- Directive: Directives to the setup compiler, such as compiler variables, conditionally including/excluding certain types of code, and web media block declarations.
- Windows Installer: Commands that have direct correlates in the MSI database that is created at build time. Most commands that actually install the application (as opposed to, for instance, displaying dialogs and controlling program flow) fall in this category.
- Modify System: Commands that apply the pending changes to the system (either an installation or an uninstallation).
- Label: Labels for use in directing program flow control.
- Statement: Any scripting command that either obtains system information, or makes direct, immediate changes to the system without going through the Windows Installer engine.

If you are a developer familiar with other Windows Installer authoring tools, you may find it easy to think of every command type, except Windows Installer commands, as custom actions. In effect, they are custom actions that are executed by the InstallAware engine. And, InstallAware also achieves conditional flow for your Windows Installer commands. The two combined together in a single setup script effectively eliminates the bifurcation problem with zero technical/conceptual overhead.

Achieving Conditional Program Flow with Windows Installer

Windows Installer does not permit conditional program flow – that is, the developer may not create setups with execution logic like the following:

```
If Variable X is True then
  Install File Set A
Else
  Install File Set B
End
```

Windows Installer does permit attaching conditions to each component that installs, however this is a very inconvenient and counter-intuitive way of achieving conditional program flow, and in most cases is not a functional equivalent of conditional program flow either.

InstallAware, with its unique Genuine Scripting for Windows Installer technology, makes conditional program flow possible. Inside the installation script, you are free to use as many Windows Installer commands as you like, and you may enclose those commands inside as many nested conditional commands as you need, directing program flow according to your installation requirements. This flexibility lets you focus on the actual logic of your setup, and gets your application installed correctly on a diverse array of hardware and software platforms, without having to worry about the implementation details on the Windows Installer side.

When you build your setup, InstallAware does several things behind the scenes:

- It parses your script for Windows Installer commands, and populates the necessary MSI table structures for them, and
- It attaches a unique condition to each Windows Installer command.

At runtime, as the InstallAware setup driver executes your scripting code, when it comes across Windows Installer commands, it sets their unique condition to true. And when the InstallAware setup driver encounters a Modify System command, it applies all pending installation (or uninstallation) changes to the system. This mechanism constitutes the heart of Genuine Scripting for Windows Installer, and very effectively creates the effect of conditional program flow on the Windows Installer platform, embracing and extending MSI technology to do what could not be done before. Genuine Scripting for Windows Installer will save you countless hours of development and testing time when building setups that require complex installation logic.

A Single Script for Installs, Maintenance, and Uninstalls

Another time saving feature offered by the Genuine Scripting for Windows Installer technology is having a unified script for the entire install/maintain/uninstall cycle. If parts of an application are to be removed during a maintenance operation, and other parts added, absolutely no additional scripting code is required to install the new parts, or remove the old parts. And of course, no scripting code is required to remove all installed parts during an uninstallation.

A script with manual coding for the install/maintain/uninstall cycles could look like the following:

```
If Installing or Maintaining
  If Feature A is Selected
    Install File Set A
  If Feature B is Selected
    Install File Set B
End

If Maintaining
  If Feature A is Not Selected
    Remove File Set A
  If Feature B is Not Selected
    Remove File Set B
End

If Uninstalling
  Remove File Set A
  Remove File Set B
End
```

With InstallAware, you have a much simpler script that does everything in one place. If parts of your application are no longer needed, InstallAware automatically removes them (during either a maintenance operation, or an uninstallation) – no manual coding is necessary. If new parts of your application must be installed, InstallAware automatically adds them. All powered by the same block of code – including the first time install:

```
If Uninstalling
  Modify System: Apply Uninstall

If Installing or Maintaining
  If Feature A is Selected
    Install File Set A
  If Feature B is Selected
    Install File Set B
  Modify System: Apply Install
End
```

When you compare the two code listings above, you immediately see that Genuine Scripting for Windows Installer, through its unified scripting model, saves you many lines of manual coding for the maintenance and uninstallation cycles of your application, while still operating on a highly customizable scripting model.

Limitations of Genuine Scripting

Because each InstallAware setup is still based on Windows Installer technology, there are some unavoidable restrictions. Fortunately these are very few in number – only two – and will not impact most setups. Moreover, these restrictions would apply to any setup based on Windows Installer technology – regardless of whether InstallAware or another Windows Installer tool was used.

The first limitation is that loops are not supported. While the InstallAware script contains constructs that allow you to code loops, if you execute a Windows Installer statement multiple times within a loop, only the last iteration of that command will actually run on the target system. Because InstallAware cannot determine at compile time the number of times a command located inside a loop may execute, it can populate only a single set of Windows Installer table structures for that command. At runtime, even though the command may execute multiple times, only the last iteration will have any effect. This is because only the last execution's setting of internal Windows Installer conditions will survive – the previous internal conditions will all have overridden one another.

The second limitation is that no changes are actually made to the target system when a Windows Installer command executes. You may think of this as InstallAware having cached and queued the command for deferred execution. The deferred commands are all executed at once when one of the Modify System commands are called (Apply Install, Apply Advertised, Apply Uninstall). Apply Install installs the application, Apply Advertised installs the application in advertised mode, and Apply Uninstall removes a previously installed application. Because Windows Installer does not permit conditional program flow, the

only alternative to this approach would be immediately applying changes to the system any time a Windows Installer command was called. While the InstallAware development team investigated this option early during development, it caused extremely severe drawbacks in execution time and was abandoned.

The impact of either limitation on installation development is very small.

For the first limitation, very rarely would commands that change the target system reside inside a loop. In fact, in the entire range of setup scripts developed by InstallAware Software Corporation, including internal development and external consulting projects, none of the setups had this requirement.

As for the second limitation, conditions that require setup commands to be applied immediately are also rare – they only occur under circumstances in which the setup flow branches based on the outcome of a preliminary setup command. If you do have a need for such branching, you will be relieved to learn that InstallAware allows you to call the Apply Install command more than once in your script, effectively sidestepping this limitation. The only drawback is that each time Apply Install is called, the Windows Installer engine is invoked – which performs a complete installation routine every time. Therefore calling Apply Install multiple times will noticeably degrade your installation speed, especially if the calls come after large blocks of file copy operations.

"Non-Windows Installer" Scripting Commands

As extensive a setup engine Windows Installer is, it does not provide the total set of installation commands – and conveniences – that one expects to find in a complete, self-contained installation authoring environment. InstallAware 2005 provides an extended, user customizable set of scripting commands which save you both time and effort in developing application installations.

Get System Information

Built-in scripting commands, which fall under the generic Statement category, offer ways to obtain extensive system information, including hardware properties (physical memory, display resolution), software properties (the running operating system, third party applications), user properties (privilege level, logged on user/domain/computer name), and more.

Advanced Install Actions

Statements are also available which perform advanced installation actions. For instance, you may create and update IIS websites and virtual folders. You can share folders or remove existing shares. You may set permissions on file system, registry, or other types of system objects. You may connect to a wide variety of database platforms and execute SQL scripts, including MS SQL, My SQL, and Oracle database servers (no client software is required on the target machine to connect to remote servers).

Run Programs, Call Dynamic Link Libraries

A very convenient way to extend the behavior of your setup scripts is to call outside code – especially if you've already encapsulated such code into stand alone executable files or dynamic link libraries. InstallAware lets you run any external program, or call any arbitrary dynamic link library function with a variable list of parameters, all directly as part of your setup script. The InstallAware 2005 IDE also makes it even easier to run programs in your setup finish dialog, or before/after the main installation/uninstallation.

Plug-In Extensibility

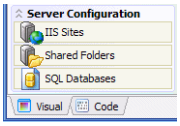
Whenever you find that the InstallAware scripting language falls short of meeting your needs (and such cases ought to be rare), you can always tap the power of your favorite programming language and directly code what you want to do in the development environment that you are most comfortable and productive

in. InstallAware is plug-in extensible, and what's even better, each plug-in command also looks and works just like a native part of your script: It is visible in the script editor, can be copied/cut/pasted like any other scripting command, and has full access to the state of the installation, including reading from/writing to script variables.

InstallAware ships with several pre-built plug-ins that perform various tasks, and also includes two plug-in templates for plug-ins implemented using the Visual C++ and Delphi programming languages. Of course, you may develop your own plug-ins in any environment capable of creating standard Win32 DLLs.

Two-Way Integrated Visual View

One of the most time-saving features when scripting in the InstallAware IDE, and one that has been greatly enhanced in the InstallAware 2005 release, is the visual view that represents the installation actions taken in your setup script graphically.



Just click the visual tab at the bottom of the InstallAware IDE window to switch to the visual view. The visual view intelligently parses your setup script and also works with highly customized scripts. It contains 26 separate pages that represent different aspects of your installation, from files being installed to your setup dialog designs. Whenever you make changes in the visual view, it will automatically update the underlying scripting code for you. It will do so without damaging your existing setup logic and correctly insert/remove code where applicable.

The visual view is a great time saver and helps you make the best of your installation script – by letting you avoid custom scripting unless absolutely necessary.

Automation Interface

While scripting setups in the InstallAware 2005 IDE is a fun and enjoyable process, there may be times in which you need to programmatically emit setup scripts without depending on the IDE. For instance, you may be building a large set of installers with very similar features but different branding/names for your network of resellers. Especially if you have a large number of resellers and installers to customize, this could very quickly escalate into a development nightmare.

With the automation interface in InstallAware 2005, you get full programmatic access to InstallAware setup scripts. Directly from your own external programs, you may:

- Emit a complete setup script, line by line
- Emit a setup project, with its own settings, dialogs, support files, and such
- Build a setup, or build a patch

This provides unprecedented flexibility in developing installers. Continuing our reseller scenario, you could build a website where each reseller could log on to their account, and upload their custom graphics and text for their brand/version of the product. Then they would click a "Build Installer" button which would immediately invoke the InstallAware automation interface, emit and build their setup, and deliver the download link. Any time you updated your software to a newer version, you wouldn't have to worry about manually updating the many installers for your resellers either! You'd just let them know that you have a newer version available, and they could all log on to their accounts and generate their personalized installers – all delivered by the InstallAware automation interface.

This is just one example for a case where InstallAware automation takes a major burden off of your shoulders. There are countless other uses for this technology. Some InstallAware licenses even permit you to redistribute the automation libraries, so you can build installers directly on end-user systems! For instance, consider a screen saver creator application – each screen saver it generates would also need an installer. Developing a home made installer for the screen savers produced would be a major undertaking. With a redistributable automation interface, you can simply call the InstallAware library functions and let InstallAware do what its best at – building setups – while you get to focus on your core expertise.

The automation interface is cleanly encapsulated inside Win32 DLLs which can be called from any Windows programming language. The automation interface also provides an ASP object which can be called from an Active Server Pages website.

The InstallAware Customer Site, which vends licenses to developers who have purchased InstallAware, uses the automation interface to create watermarked installers that contain the name of the individual or organization that purchased the license.

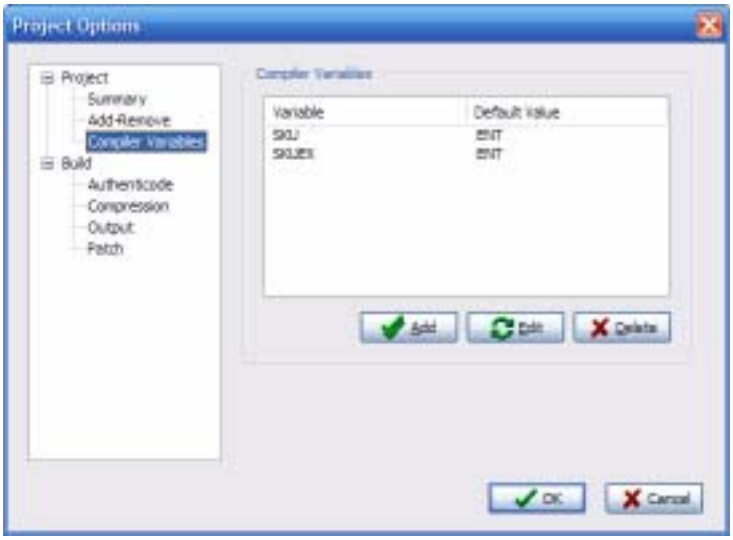
Compiler Variables

There may also be cases in which you need to customize your installers – but not as thoroughly as one might using the automation interface. For instance, you may have several editions for a product, each with progressively more features and higher prices. When building the installers for these various editions, instead of coding multiple scripts for each, you would rather code and maintain a single script. In this case, using the automation interface would be overkill – another way to quickly customize the script becomes necessary.

Compiler variables are resolved at build time, and let you achieve the following:

- Include or exclude various parts of code based on compiler variable values
- Substitute compiler variables used in script with their literal values

Continuing our example with the various editions, you could use Compiler Variable If statements to conditionally include/exclude parts of your setup script and application files, based on the edition. You could also use compiler variables directly in the script for build time substitution of their values into the setup script (such as, the edition name).



Compiler variables, and their values, are defined in the Project Options window. The values specified in the Project Options window may also be overridden while building from the command line, or via the automation interface. Coupled with the redistributable command line build interface that is available in some InstallAware licenses, compiler variables may also be used to create customized installation packages directly on end-user systems.

The InstallAware product itself comes in various editions, and each edition has a different set of features and source files. The setup script for InstallAware makes use of compiler variables and delivers all the different editions from a single source script.

Integrated Debugging

Despite the beauty and power of Genuine Scripting for Windows Installer, or perhaps because of it, the need to debug the scripts you have developed in InstallAware will arise sooner or later. Along with the canonical MessageBox command that can be used to output values of variables to the screen, InstallAware 2005 provides a visual, integrated debugger right within the main IDE.

The run menu contains all the debugging commands you need:

- Start debugging
- Step through code line by line
- End a debug session
- Add variable watches
- Set breakpoints in source code

You may also override the values of variables while in a debugging session to test alternate routes of program flow.

Web Aware

InstallAware©
2005



Web Deployment (Partial Web Deployment)

Another unique feature of InstallAware 2005, one that is unavailable in any other setup development tool, is the possibility of partial web deployment.

Traditionally, setups that are web deployed reside entirely online. The end-user downloads a very small stub that contains only the setup driver. The stub then connects to the Internet and downloads the entire remainder of the setup from the Internet. This approach definitely saves time in some cases – for instance, the stubs don't download runtime components, such as the .NET runtime, if they are already found on the end-user system. However, having to download the entire application from the web as well, in addition to the runtimes, is an inconvenience at best. Even if the end-user system already has the required runtimes, the setup will still require and force an Internet connection.

The other alternative is to pack every runtime, along with the application itself, into a self extracting installer which contains everything. However, the downside to this approach is that while the setup is completely self-contained, and thus installs without requiring an Internet connection, it is bloated and may already contain runtimes that the end-user already has. Especially in the case of larger runtimes, such as the .NET runtime, this bloat can far exceed the size of the main application itself, and frustrate users in that they have to download and wait for what they already have on their system. InstallAware has had the perfect answer to this dilemma since its very first version: web media blocks technology.

Web Media Blocks

In InstallAware 2005, setups are comprised of one or more web media blocks. A web media block is actually a script directive, very much like the compiler directives found in most programming languages. It tells the setup compiler how the setup is to be split into multiple online and offline pieces.

A web media block directive has two fields: the name of the web media block, and its download URL. The name of the web media block determines the name of the output file that is created for that web media block. This file name will typically be of the form <block name>.7zip. The download URL indicates where the installer can expect to find the web media block file at runtime, and can point to any HTTP, FTP, or network location. The default file name of the web media block can also be modified in the download URL. As long as the actual web media block file exists at the given location, the download will succeed.

There is also a special kind of web media block, called the offline block. If a web media block statement contains an empty name, that web media block is considered to be an offline block, and the download URL field is disabled. The installer does not create a separate file for an offline block, and instead packages everything inside the offline block into the main setup executable (hence, the “offline” block).

When InstallAware 2005 builds your setup, it parses your script and identifies which files belong inside which web media blocks. A web media block directive has effect from the point of its declaration in the script up to the next web media block directive. Each installation command that requires source media, such as Install Files, becomes associated with the last declared web media block directive and the files for those commands are packaged into the web media block file defined by that directive, to be downloaded dynamically at runtime. You may declare an unlimited number of web media blocks in your setup script, and also use identical declarations more than once, in this way precisely crafting the web deployment structure of your setup.

At runtime, when the setup script is executing on the end-user system, if an installation statement that falls inside a web media block executes, the installation engine attempts to download that web media block (if it has not been previously downloaded). If no statements that require access to source media (such as Install Files) execute inside a particular web media block, that web media block will not be downloaded. The

download process is resilient and has the capability to resume after an interruption. Moreover, inside corporate environments where proxy servers are used, the download process automatically detects correct proxy settings and works seamlessly.

Web media blocks technology, only available with InstallAware, provides flexible, partial web deployment capabilities for the first time in setup development history. You may place each separate application runtime inside its own web media block, therefore preventing bloating the size of your main setup executable. You may also use this facility to separate rarely used or space consuming features of your application from the main setup file. And any part of your setup which is inside an offline block goes directly inside the main setup file, helping you build a self contained installer that works without demanding an Internet connection.

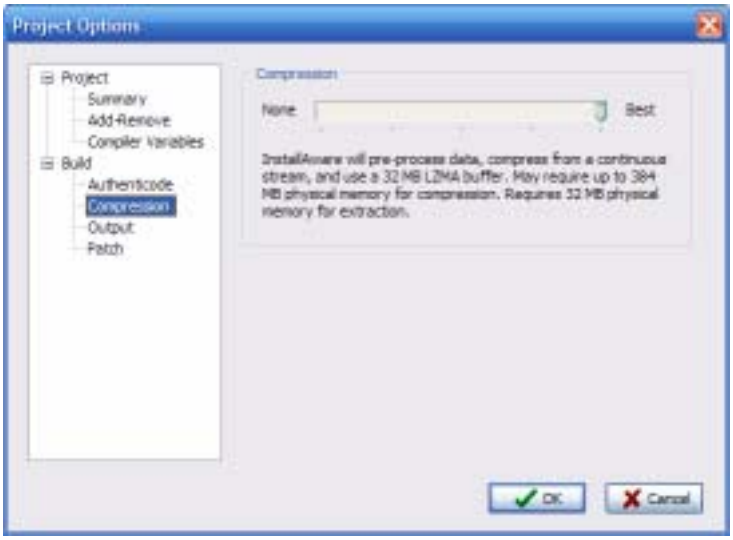
Splitting your application into web media blocks will not only save your users undue frustration, but it will also result in significant bandwidth savings for your business. Because large runtimes and optional features will be downloaded only when absolutely required, your servers will conserve bandwidth up to levels not possible before.

While every InstallAware setup script contains one or more web media blocks, the build process will generate web media blocks only if you are targeting the web (as configured in the Project Options dialog). If you are doing an uncompressed or compressed build, web media blocks will be excluded from the installation, and every part of your setup will be available as either a single monolithic compressed file, or a collection of uncompressed files and folders suitable for CD/DVD deployment. This adds an additional

layer of convenience when you are targeting multiple types of distribution media and avoids the tedium of having to manually modify your setup script for each different output type.

Superior Compression

In tandem with its unique web media blocks technology, InstallAware 2005 provides a revolutionary form of compression that works nothing short of miracles in reducing file sizes. InstallAware 2005 expands on the abilities of its previous versions and goes even further in reducing download times and sizes.



Compression settings are configured in the Project Options dialog, and five levels of compression are available, with a sliding scale from none to best. The best compression mode utilizes a very advanced data processing algorithm that pre-processes all files using Binary Call Jump Converters (version 2), thus increasing their compressibility; and then passes the optimized data streams through a LZMA encoder with a 32 MB compression buffer.

InstallAware's compression speaks for itself. The following table lists size gains for several popular application runtimes after being processed by InstallAware:

Runtime	Compressed Size	InstallAware Size	Compression Gains
Windows Installer	3,446 KB	1,770 KB	49%
Internet Explorer 6	79,849 KB	47,110 KB	41%
Microsoft .NET 1.1	23,697 KB	12,338 KB	48%
Microsoft JSharp	6,764 KB	2,596 KB	61%

Notice that the compression gains listed in the table above are over the already compressed sizes of the runtimes. In most cases, InstallAware reduced more than half in size what was already the compressed size of the runtime.

Similar compression benefits will carry over to your own program files as well, creating the smallest possible installers available today.

Repacking Databases

While the examples above imply that it is possible to re-compress pre-compressed data, this is not the case. No matter how good a data compression algorithm is, because of the law of entropy in data compression theory, it cannot recompress data that has been compressed before, even if the prior algorithm was an inferior one. Therefore, InstallAware actually comes with de-compressed copies of the runtimes in the list above (among others), and it is these runtimes which it compresses to half the size of the original compressed sizes. But what about your own runtimes and program files?

When you are trying to take advantage of the superior compression that is provided by InstallAware, you should make sure that none of the source files going into your setup are pre-compressed. For instance, do

not use EXE-packers, which reduce your executable program size, with InstallAware setups – because while the EXE sizes will indeed be reduced with the EXE-packer, they will be reduced to even less of a size if you let InstallAware compress them from scratch.

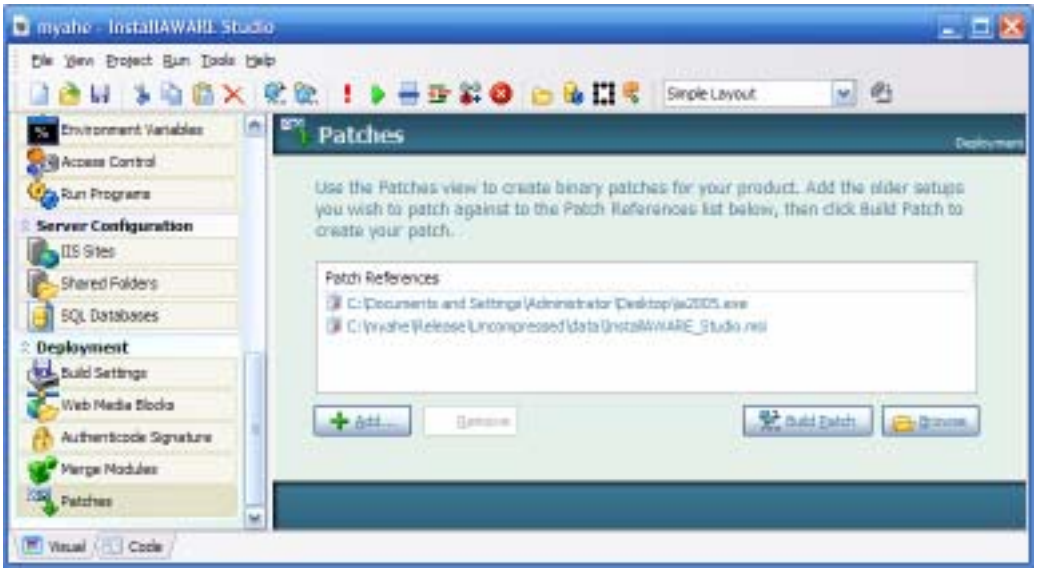
There might also be cases in which you include MSI and MSM Windows Installer databases in your setup, to install additional components or dependencies. Unfortunately, MSI and MSM databases generally contain compressed data streams within them, and of course, even though the native Windows Installer compression is inferior (CAB-MSZIP/CAB-LZX), InstallAware will be unable to further compress this data because it has already been compressed.

InstallAware 2005 provides a new tool, the MSI CAB Decompressor, which you can use to repack these installation databases. This tool opens MSI and MSM databases, and updates the compressed data they contain with the exact same data in uncompressed form. This allows InstallAware's superior LZMA/BCJ2 algorithm to work miracles in re-compressing all the data that is found inside the Windows Installer databases.

Whenever you add an MSI or MSM database to your installation, be sure to process it first with the MSI CAB Decompressor tool.

One-Click Patching

No matter how good the compression you use, and how many web media blocks you split your setup into, the time inevitably comes when you need to issue an update to your end-users, and you just cannot afford to resend a complete installer. This is especially the case when a relatively minor number of files have been updated for a relatively small number of changes and bug fixes in your application, and you just want to update those files – instead of re-distributing the entire setup from scratch.

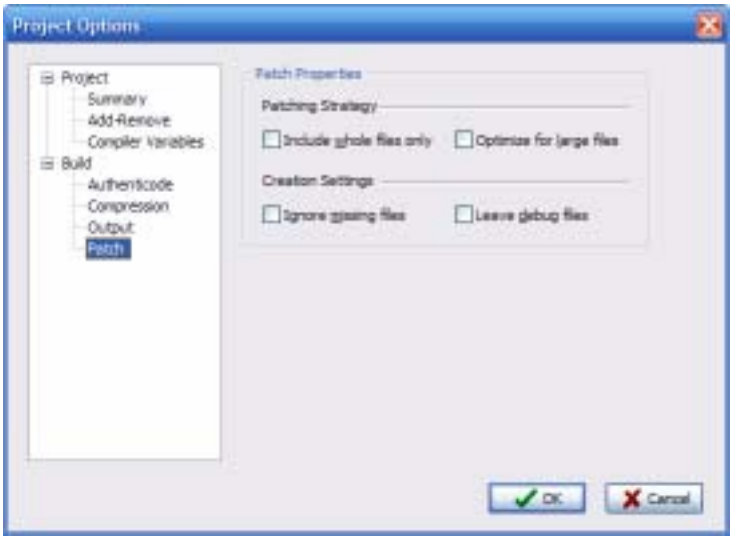


InstallAware 2005 introduces the novel one-click patching feature to address this need. To use one-click patching, just include its latest set of files in your setup as before. Then, add the installers for the previous versions of your product to your setup project as Patch References (only add the installs that you wish to support patching against). When you click the Build Patch button, InstallAware does the following, all automatically, with no manual intervention or supervision on your part:

1. Builds the latest version of your application setup
2. Extracts the referenced previous versions of your application setups (if compressed), to a temporary working folder
3. Compares the current version of your setup with the previous versions
4. Invokes the Windows Installer engine to generate a native Windows Installer patch file
5. Repacks the native Windows Installer patch file with superior compression

InstallAware takes care of all the dirty work in patch creation for you. Unlike other leading commercial products, no upgrade syncing, no patch compatibility testing, and no pre-flight deployment testing is required. InstallAware patches take truly one-click to create, and they just work, without complications.

The created patches contain the absolute minimum data and files that are required to upgrade your old versions to the newest version. This is made possible by taking binary/differential scans of the files in your different versions, and examining them. Only the changes between the different versions are included in the patch. Even when patching very large applications, the output patch files will be very small – dramatically simplifying patch distribution and reducing your costs.



Further patch creation options, such as using full files instead of storing binary/differential patch data, are also available in the Project Options dialog.

Windows Installer Aware

InstallAware©
2005



Windows Installer Functionality

InstallAware has always been based on Windows Installer, the software installation technology that is developed by Microsoft and is gaining ever-increasing acceptance. Windows Installer has always been strong in the corporate market because of its innovations in deployment and management, and it is also gaining increasing acceptance in the desktop market. The future is clearly heading in the direction of Windows Installer – and InstallAware takes you there with a minimum of effort today.

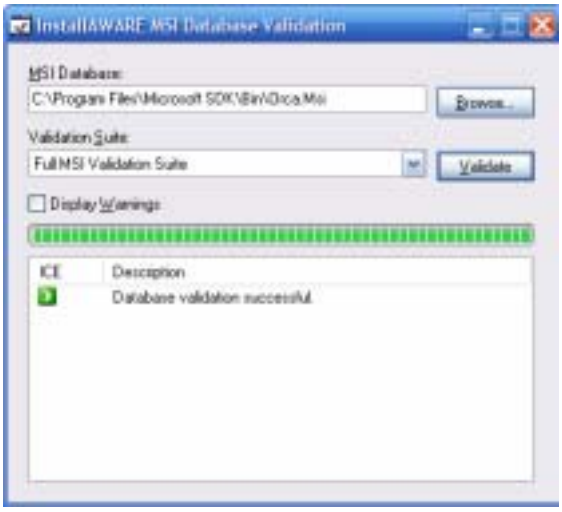
InstallAware 2005 adds even more support for Windows Installer features, and does so in the traditional InstallAware way – with no effort or expertise necessary on your part. While Windows Installer is very complex to learn and use, InstallAware has always provided crisp abstractions in its IDE and setup authoring process, isolating you from all intricacies of the underlying relational database and implementation details. InstallAware 2005 extends these abstractions to the entirety of the Windows Installer domain – so today, with InstallAware 2005, you leverage all the power of Windows Installer, with none of the cost.

Logo Certifiable Setups

One of the application requirements for earning the coveted Designed for Windows logo, as determined by Microsoft, is having a setup routine developed in Windows Installer. This requirement has two main parts:

1. The setup database (MSI file) must pass ICE (internal consistency evaluators) tests
2. The setup database must create advertised entry points for the application, in order to enable install-on-demand

InstallAware MSI databases have always passed ICE tests. As provided by the Genuine Scripting for Windows Installer technology, whenever an InstallAware setup script is compiled into an MSI database, it is automatically ICE compliant.



You may manually test the ICE compliancy of your setup databases at any time using the MSI Database Validation Tool. The tool contains specialized validation suites for Windows 2000 and Windows XP logo programs. It also runs on all Windows platforms (unlike most validation tools which fail to validate running under the Windows Server 2003 operating system).

New in InstallAware 2005 is added support for the second part of the logo requirement: advertising. InstallAware 2005 adds several new scripting commands which support the creation of advertised entry points in applications. The setup engine can now also perform advertised installs of a product.

Advertising

Advertising is a concept in application installations that is unique to Windows Installer. Currently only Windows Installer based setups, including those created by InstallAware, support advertising. In a nutshell, the notion of advertising represents two capabilities:

- An installation mode, in which an application is not actually installed, but only appears to be installed; to be fully installed upon first use (install-on-demand)
- A way to verify the integrity of an application installation every time it is launched (self-healing installations)

Install-on-Demand

When we install an application, our indicators that the application is installed are shortcuts in the Start Menu, new file types which are now visible in Explorer with their custom icons, and the like. These are termed "application entry points," and any time we actually invoke the application through one of these entry points, the actual application executable is run. And of course, the application executable is already present on the system.

When an application is installed in advertised mode with Windows Installer, that is, the application is "advertised," only the entry points of the application are created. Thus, to the ordinary eye, the application appears to be fully installed – the Start Menu items are all present, new Explorer file types are visible with their custom icons, and so on. However, during an advertised install, no files are actually copied. Advertised installs therefore dramatically accelerate installation time, and provide the appearance that the advertised product is actually installed, when in fact only the application entry points are installed – hence, the application is being 'advertised'. When an advertised entry point is actually invoked, such as a Start Menu shortcut being launched or a file associated with the application being opened, the Windows Installer service automatically intervenes and immediately installs the required files, and then seamlessly launches the application.

This is called install-on-demand and is made possible with Windows Installer. InstallAware 2005 fully supports installation-on-demand, and even better, does not require any changes to your script sources for advertising. When you use the Apply Advertised command in your setup script, InstallAware advertises the

product (compare this behavior to the Apply Install command, which fully installs the product). New setup scripts created using InstallAware 2005 with either the project templates or the project wizards allow specifying command line parameters for running the installation in advertised mode. You may also copy-paste code found in the newer projects to your older ones, so that they may also support advertised installs from the command line.

Self-Healing Installations

Another benefit of the advertising technology offered by Windows Installer is the ability to create self-healing installations. Windows Installer not only supports install-on-demand from application entry points, it also uses the application entry points as an opportunity to monitor the health of the application installation. In particular, if the target of a Start Menu shortcut or a file type is missing, Windows Installer will again automatically invoke and repair the installation, replacing the missing files.

This self-healing behavior helps assure the integrity of applications and provides an additional level of convenience for end-users who may have lost application files due to their own error, a system fault, or any other cause. Whenever you use the new scripting commands in InstallAware 2005 that create advertised entry points, your installs will automatically become self-healing. The InstallAware 2005 IDE also features enhanced visual views that automatically convert your shortcut and file type commands to the advertised kinds, without you having to manually recode them.

Of course, repairing an application through the maintenance mode (or changing the installed application feature set) is still available with InstallAware 2005, and works exactly like before. The new self-healing and install-on-demand features are invoked directly by Windows Installer at application entry points only (as is the case with all other Windows Installer setups).

Elevated Privileges

Operating systems based on the Windows NT kernel (Windows NT 4, Windows 2000, Windows XP, Windows 2003 and Windows Longhorn) allow setting permissions on file system and registry objects, thus preventing unauthorized users from making changes to important files and settings. This also has a downside – users with less than administrator privileges often find themselves unable to install applications without direct assistance from a system administrator.

When the "Always Install with Elevated Privileges" settings is enabled in the system group policy editor, the Windows Installer service permits users with limited access rights to perform system level changes to the system – that is, install files and registry entries into locations where the logged on user would not ordinarily have access. This is possible because Windows Installer is a system level service and runs with elevated privileges.

All installations created with InstallAware automatically take advantage of elevated privileges and successfully install inside many locked-down corporate and other types of secured environments.

Corporate Deployment

One of the main reasons Windows Installer rapidly became a corporate standard is its ease of deployment. If you are managing thousands of workstations on your corporate network, it is simply out of the question to manually install applications on all those workstations. Instead, you have to automate the deployment process. Windows Installer has been adopted rapidly in large organizations because it lends itself very cleanly to such types of mass deployment and maintenance.

Installations created with InstallAware have always supported silent installations, along with a configurable set of command line parameters for specifying advanced installation settings. What's more, in addition to

the standard configurability provided by Windows Installer, InstallAware installations can define their own custom command line parameters and give systems administrators even more automated deployment options. Setups execute completely silently in the background and also provide logging options for post-mortem diagnostics in case of complications with the deployment procedure.

Even better, the Web Media Blocks technology uniquely featured in InstallAware offers some very exciting opportunities when used in large organizations for deployment. If you specify your Web Media Block sources on corporate file servers (as opposed to Internet URLs), you get several advantages:

- Security: Setups will not run outside of the organization network, as the web media blocks will be accessible only on that network
- Versioning: Setups will always have access to the latest version of setup sources, since the centralized web media blocks will always contain the most up to date files
- Patching: Especially for patches that require access to source media (a problem that makes patch deployment a nightmare in large environments), web media blocks provide automatic source resolution, even across different versions of the same product

In the true InstallAware tradition, InstallAware 2005 takes corporate deployment one step further with the additional features and functionality it offers, on top of those already provided by Windows Installer – saving you time and money with its advanced deployment mechanisms.

Shelling to Other MSI Installations

Windows Installer offers the merge module mechanism for extending your setups. Merge modules are Windows Installer database files with the MSM extension, self contained blocks of installation logic and code, used for installing common parts of applications (such as application runtimes). Merge modules are "merged" into the main installation database, the MSI file, at build time, and with absolutely no effort on part of the developer, extend the MSI database with their additional logic and code, providing for the installation of the application runtimes, and so on.

InstallAware 2005 supports merge modules, but again adds an extra bit of functionality. With InstallAware 2005, you may (un)install or reconfigure complete third party products, given their MSI files, or product GUIDs, in addition to merging MSM databases with your setup.

The (Un)Install MSI Package command provides unprecedented flexibility in "shelling to" third party MSI packages. You may perform the full range of setup actions that are supported by Windows Installer on shelled packages – install or advertise, select all features or just a partial set, and even modify/remove an existing installation.

The command lets you customize the exact command line that will be used in the shelled-to installation, offers the ability to log setup execution, and reports whether the operation was successful, optionally capturing the error message if it failed. Setup packages are identified by their GUID or full path to the setup database. What's best, InstallAware even captures the progress of the shelled-to installation, displaying it as an integral part of your own application installation. Of course, you are able to completely hide the end-user interface of the MSI package you are shelling-to, including hiding any potentially confusing setup choices and dialogs. You get to install the packages with the exact settings you require, with no effort/confusion on part of the end-user running your master setup.

Shelling to setups provides a very viable alternative to application repackaging. Unlike application repackaging which is error-prone, and loses the intent of the original package author, shelling to setups preserves both the original setup logic of the installation, and adds the benefit of being a completely seamless and integral part of your own, larger installation.

Possible applications of this technology are endless. A very viable use for the shelling technology is resolving application conflicts automatically. For instance, if an application is detected on the end-user system which conflicts with your product, it can automatically be removed, all as a setup pre-requisite for your own installation. The Is MSI Setup Installed command reports whether a particular MSI package has been installed on the system, and complements the shelling facility in resolving application conflicts – all with zero end-user intervention/confusion.

Summary

Built from the ground up, InstallAware 2005 achieves what no other installation development environment can, providing you with state-of-the-art tools that save you time and effort while building modern application installations. InstallAware 2005's unique fusion of scripting, web deployment, and Windows Installer technologies makes InstallAware 2005 the ultimate setup development solution.

About InstallAware Software Corporation

InstallAware Software Corporation was founded by a former InstallShield employee in 2003. The company focuses on software installation technologies for the Windows Installer platform and brings a fresh approach to the setup development process. The privately held company has received numerous seed investments from venture capital firms and is a Borland Technology Partner. InstallAware's parent company, MimarSinan International, has been publishing software online since 1996.