WebAware Installations

(Partial Web Deployment)

An InstallAware Whitepaper

August 2007

Contents

Introduction

InstallAware allows splitting the data contained inside a setup package into multiple online and offline parts. Web deployment traditionally involves putting the entire setup online, including the vital files that are always installed. This mandates 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. Isolating runtimes and rarely used application features from the main setup download saves time and bandwidth for end users installing the applications, and also for companies hosting downloads. Application runtimes which may already be present on end user systems, and typically bloat setup downloads, are ideal candidates to benefit from WebAware Installations. With WebAware Installations, 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.
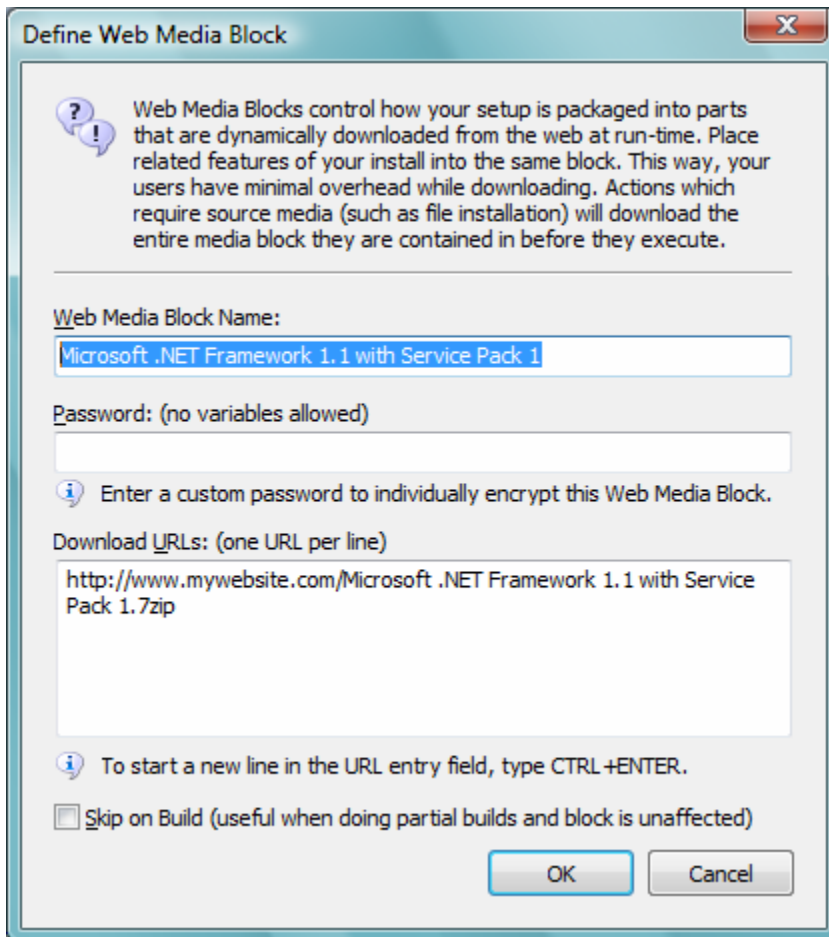
WebAware Installations (Partial Web Deployment)

A unique feature of InstallAware is the possibility of partial web deployment. With all other tools, 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 installed on the end-user system. However, having to download the main 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 that 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 provides the perfect solution to this dilemma since its very first version: web media blocks technology.

Web Media Blocks

In InstallAware 7, setups are comprised of one or more web media blocks. A web media block is actually an MSIcode directive, very much like the compiler directives found in most programming languages. It tells the MSIcode compiler how the setup is to be split into multiple online and offline pieces.

A web media block directive has three fields: the name of the web media block, an optional encryption password, and one or more download URLs. 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 password enables each web media block's data to be individually encrypted using strong 256-bit AES encryption. Download URLs indicate where the installer can expect to find the web media

block file at runtime, and can point to multiple HTTP, FTP, or network locations, adding redundancy and scalability. The default file name of the web media block can also be modified in the download URL list. As long as the actual web media block file exists at one of the given locations, 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 any additional fields are 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 7 builds your setup, it parses your MSIcode 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 MSIcode 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 MSIcode, and also use identical declarations more than once, in this way precisely describing the partial web deployment structure of your setup.
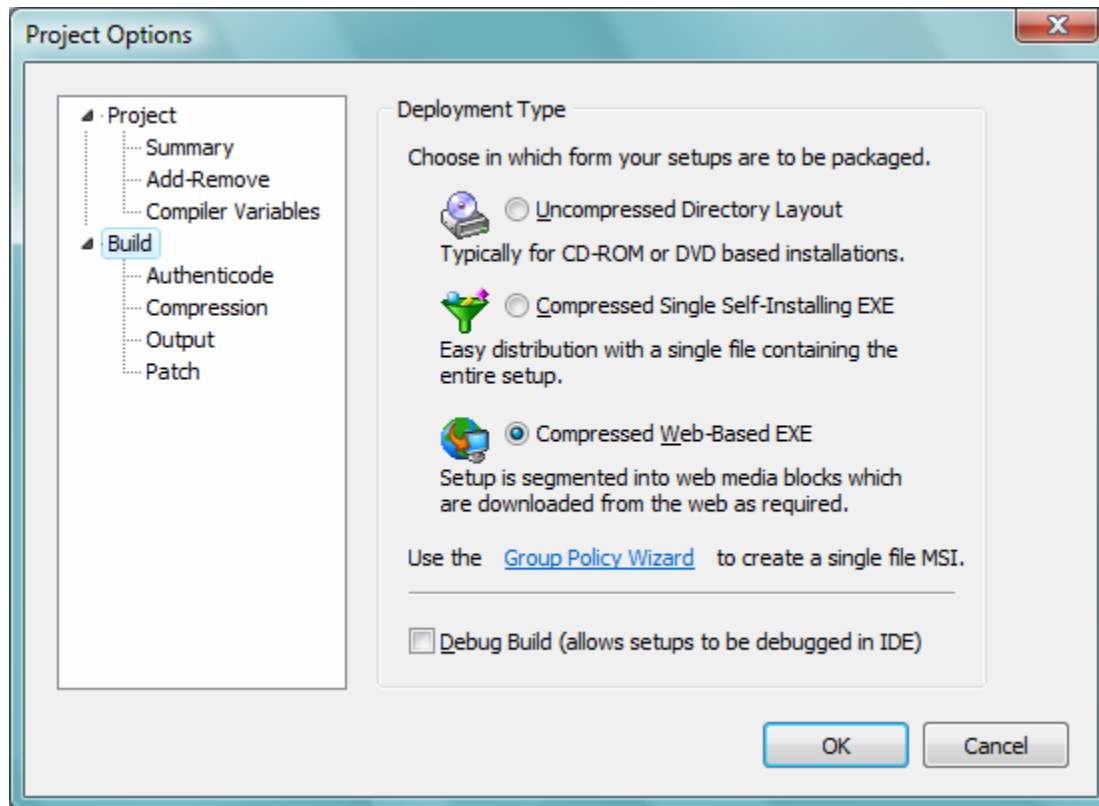
At runtime, whenever an MSIcode statement that falls inside a web media block executes, the installation engine attempts to download that web media block (if it has not been previously downloaded). If none of the statements that require access to source media (such as Install Files) inside a particular web media block execute, that web media block will not be downloaded. The download process is resilient and has the capability to resume after an interruption. Moreover, if operating within a corporate environment where proxy servers are used, the download process automatically detects correct proxy settings and works seamlessly.

WebAware Installation 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.

InstallAware 7 improves web media block security by allowing each web media block to be individually encrypted with a separate key. InstallAware 7 also improves web media block resiliency, by keeping a full list of download URLs for each web media block, in case one (or more) URLs fail to respond.
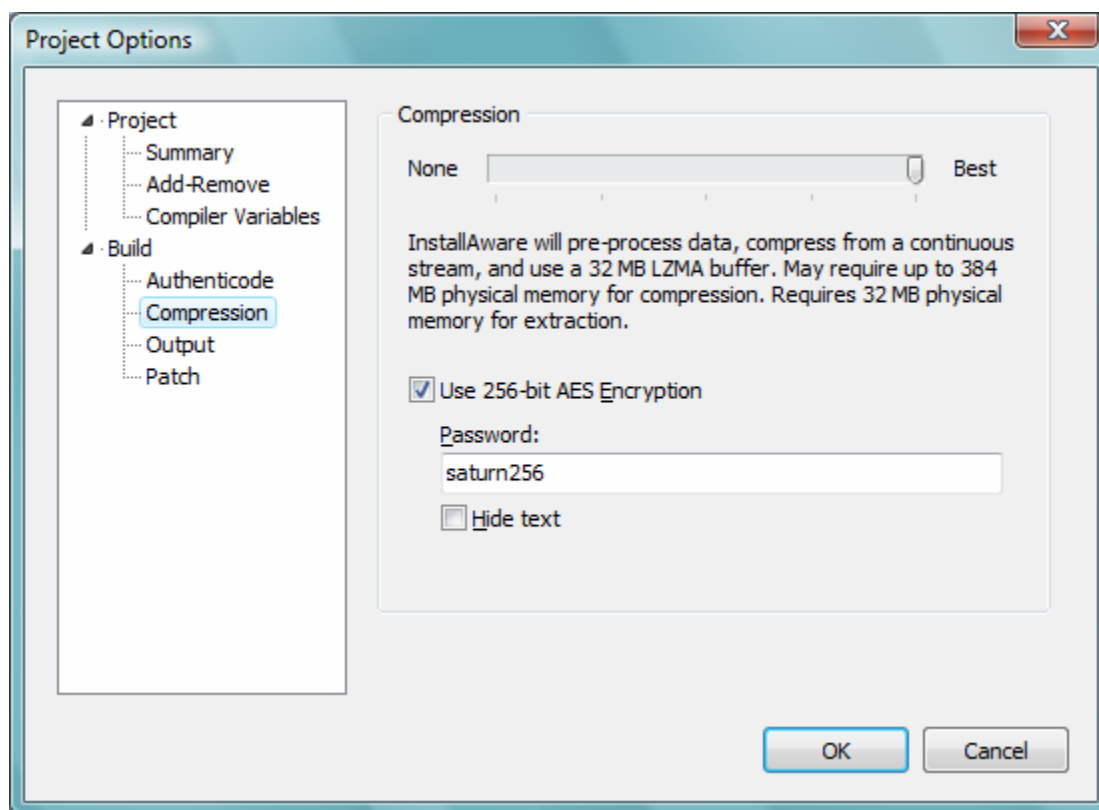

While every InstallAware MSIcode script will contain one or more web media blocks, the build process will generate web media blocks only if you are targeting the web (as configured in the Project Options dialog). If you are doing an uncompressed or compressed build, web media blocks will be excluded 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 project for each different output type.

Superior Compression

In tandem with its unique web media blocks technology, InstallAware 7 provides a revolutionary form of compression that works nothing short of miracles in reducing file sizes. InstallAware 7 expands on the abilities of its previous versions and goes even further in reducing download times and sizes.

New in InstallAware 7 is the ability to split compression tasks across multiple CPUs, CPU cores, or logical (hyper-threading) CPU units. Compression is computationally intensive, and parallel execution improves compression speeds by as much as 50% – saving you time, as well as delivering better compression ratios.

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.

You may also protect compressed setups with a global password. Data is encrypted using strong 256-bit AES encryption for bullet-proof security.

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

Runtime	Compressed Size	InstallAware Size	Compression Savings
Windows Installer 2.0	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 .NET 1.1 SP1	34,150 KB	11,403 KB	67%
Microsoft JSharp 1.1	6,764 KB	2,596 KB	61%
Microsoft .NET 2.0	22,959 KB	17,272 KB	25%
Microsoft SQL Express 2005	54,791 KB	33,036 KB	40%

Notice that the savings 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 carry over to your own program files as well, helping you create 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
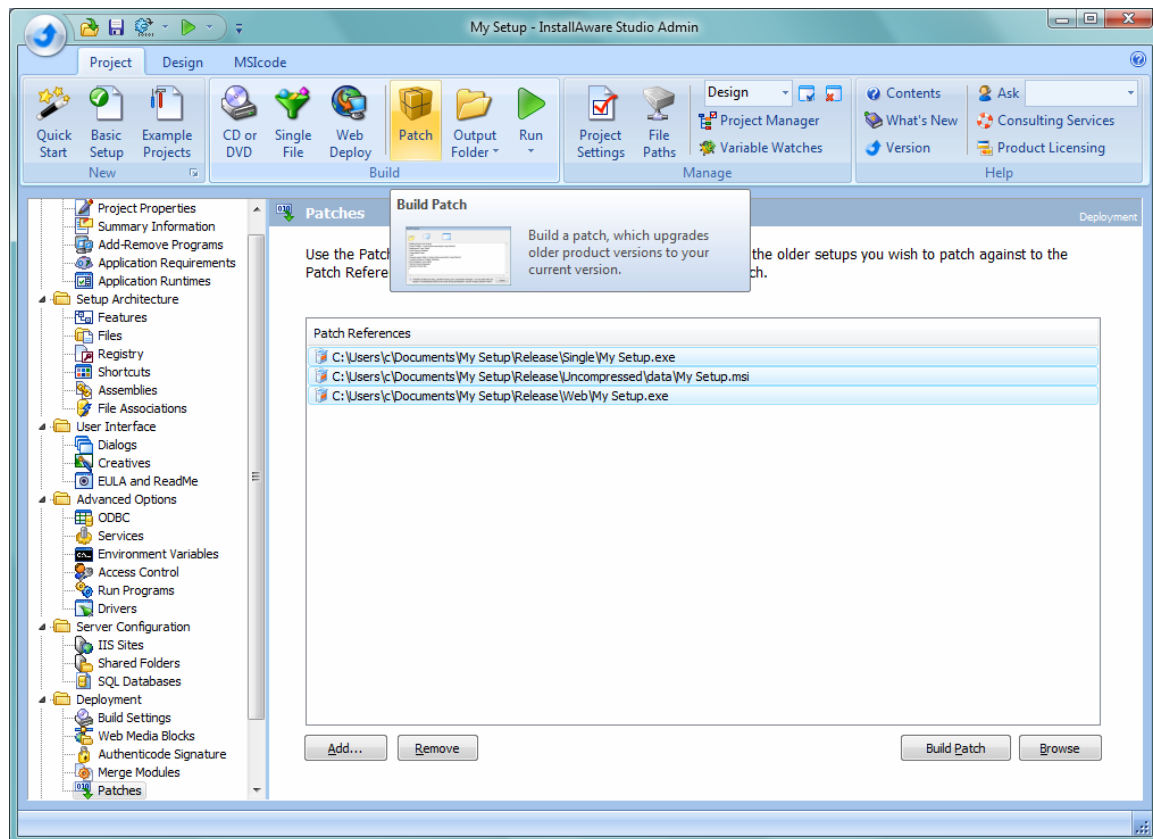
not use EXE-packers, which reduce your executable program size, with InstallAware setups – because while the EXE sizes will indeed be reduced with the EXE-packer, they will be reduced to even less of a size if you let InstallAware compress them from scratch.

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 7 provides a fully automated tool, the Setup Decompressor, which you can use to repack these installation databases. This tool works with MSI and MSM databases, as well as most Microsoft EXE setups. It 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 or a Microsoft EXE setup to your installation, be sure to process it first with the Setup 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.
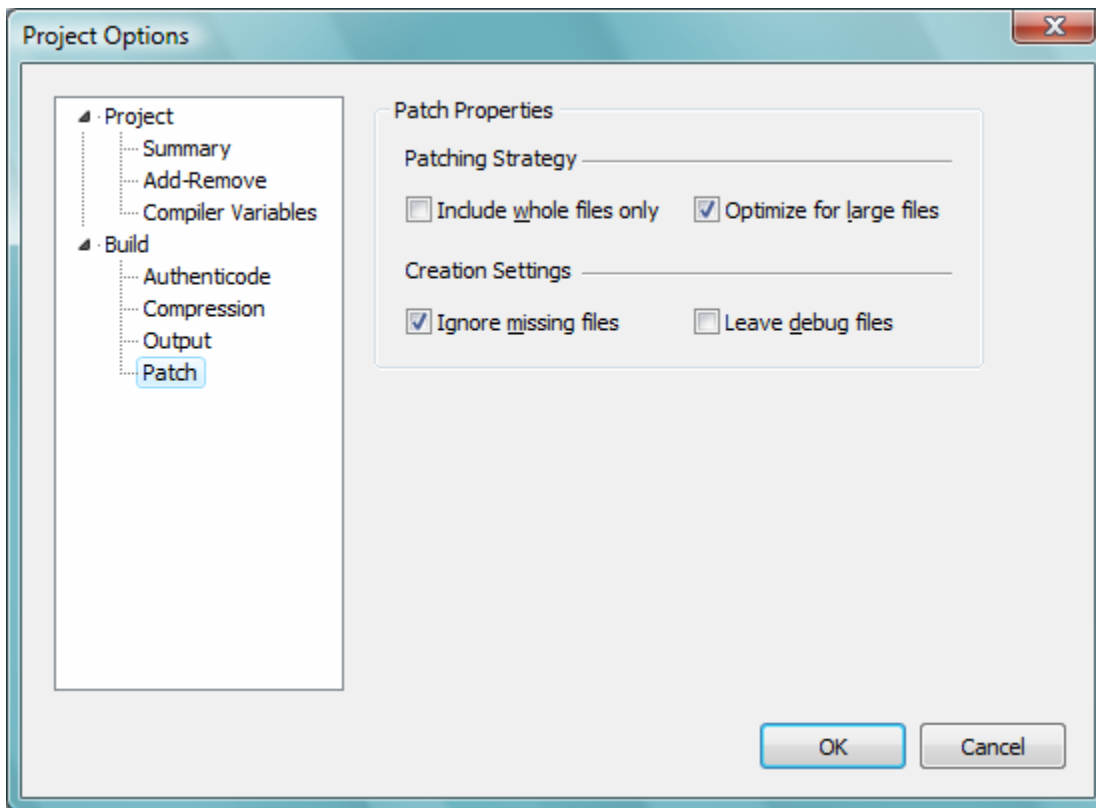
InstallAware 7 provides the unique one-click patching feature to address this need. To use one-click patching, first prepare your setup normally. Then, add the installers for the previous versions of your product to your setup project as Patch References (only add the installs that you wish to support 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 (.MSP), and
5. Repacks the native Windows Installer patch file using InstallAware's 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 binary/differential scans of the files in your different versions. 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.

Summary

InstallAware 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's unique fusion of scripting, web deployment, and Windows Installer technologies makes InstallAware the ultimate setup development solution.

About InstallAware Software Corporation

Focusing solely on the Windows Installer (MSI) platform for Microsoft Windows operating systems, InstallAware is the premier provider of software installation tools for Internet deployment. Founded in 2003, InstallAware is supported by thousands of users worldwide, and is a Borland Integrated Partner. For more information, visit www.installaware.com.