

Application Update Framework for .Net



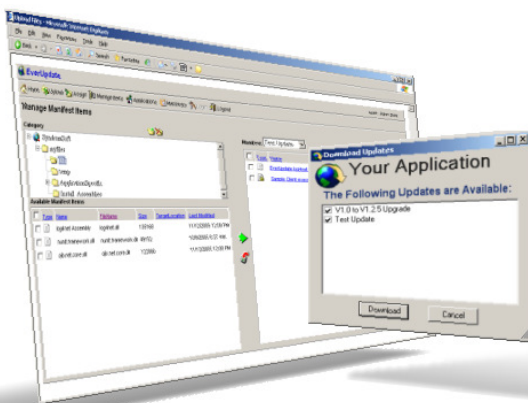
Practical Apps is proud to introduce **EverUpdate™**, a complete solution for update-enabling your end-user applications.

Why invest countless man-hours developing custom update facilities for your .Net end-user applications? The **EverUpdate** SDK provides a simple component that you embed in your applications to instantly enable robust patch and update management capabilities. Via simple API calls, you can enable your

application to check for available patches or updates. You can even use the included 'update daemon' as-is to place an update checker in your end-user's Systray. Patches and Updates are hosted via our hosting service or, if you purchase an Enterprise license, via your own web service application. *Update Manifests* are managed through a simple, web-based interface.

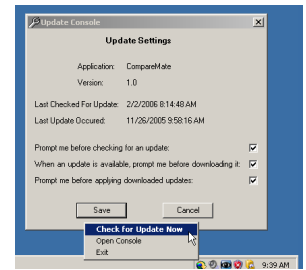
Features

- Ability to define an update "Manifest" which contains Manifest items.
- Each Manifest Item is one of several "Deployment Types" : File, Assembly, Registry Entry
- Support for Add, Replace & Execute of File deployment type
- Support for Add & Replace of Registry Entry deployment type
- Support for including custom Installers that comply with the .Net Installer architecture
- Support for update of locked files (requires reboot)
- Support for additional Deployment Types in the future.
- Supports defining patches based on application name, version and patch level
- Optional, configurable prompts for: checking for update, prompting before downloading, prompting before applying, success notification
- Optional failure notification
- Pluggable interface for custom prompts/notifiers with default implementations
- Ability to require reboot following patch
- Logging of update activities
- Includes a simple client library (assembly) which enables easy integration into existing apps
- Simple setup program installs all SDK components and samples
- Includes sample applications (with source):
 - A stand-alone, demonstration client application
 - An update daemon that runs in the windows Systray and checks periodically for updates



Getting Started with EverUpdate

Getting started with EverUpdate can be as simple as installing the client SDK, and using a config file tailored to your application with the included Systray daemon sample application.



Alternatively, you can refer to the included developer's guide and sample client source code for how to embed update functionality directly in your application. You may also choose to create and register custom GUI windows, such as a form containing a progress bar, with the updater.

For more advanced control over update behavior you can also call specific API's exposed by the EverUpdate client proxy, such as `IsUpdateAvailable()` or even register callbacks with the proxy so that your application is notified of progress as update check, download or apply processes occur.

Options for Hosting Your Updates

During the Beta, if you register for EverUpdate, an account will automatically be created for you in our hosted Administration website. You will be able to create Patch Manifests limited by size and number of components for testing purposes. When we release the licensed version of EverUpdate following the Beta, we currently plan on offering two pricing models:

Under the first model, you purchase the SDK, embed the client library in your applications, and we host your admin and update services for a monthly fee.

In the second model, we will offer Enterprise licenses which will include the hosting applications and permit you to host your own updates.

SDK Prerequisites

- Microsoft® Windows XP or 2000 operating systems
- Microsoft .Net Framework V1.1
- Microsoft Visual Studio 2003 or 2005
- Microsoft's Web Services Enhancements (WSE) 2.0 (free download from Microsoft)

Hosting Prerequisites

- Microsoft® IIS v5 or later
- Microsoft® .Net Framework V1.1
- Microsoft® Web Services Enhancements (WSE) 2.0
- Microsoft® SQL Server 2000 or later or MSDE database