



UpdateBuilder

Dec 2016

Copyright © 2016 by tmssoftware.com bvba
Web: <http://www.tmssoftware.com>
Email : info@tmssoftware.com

Index

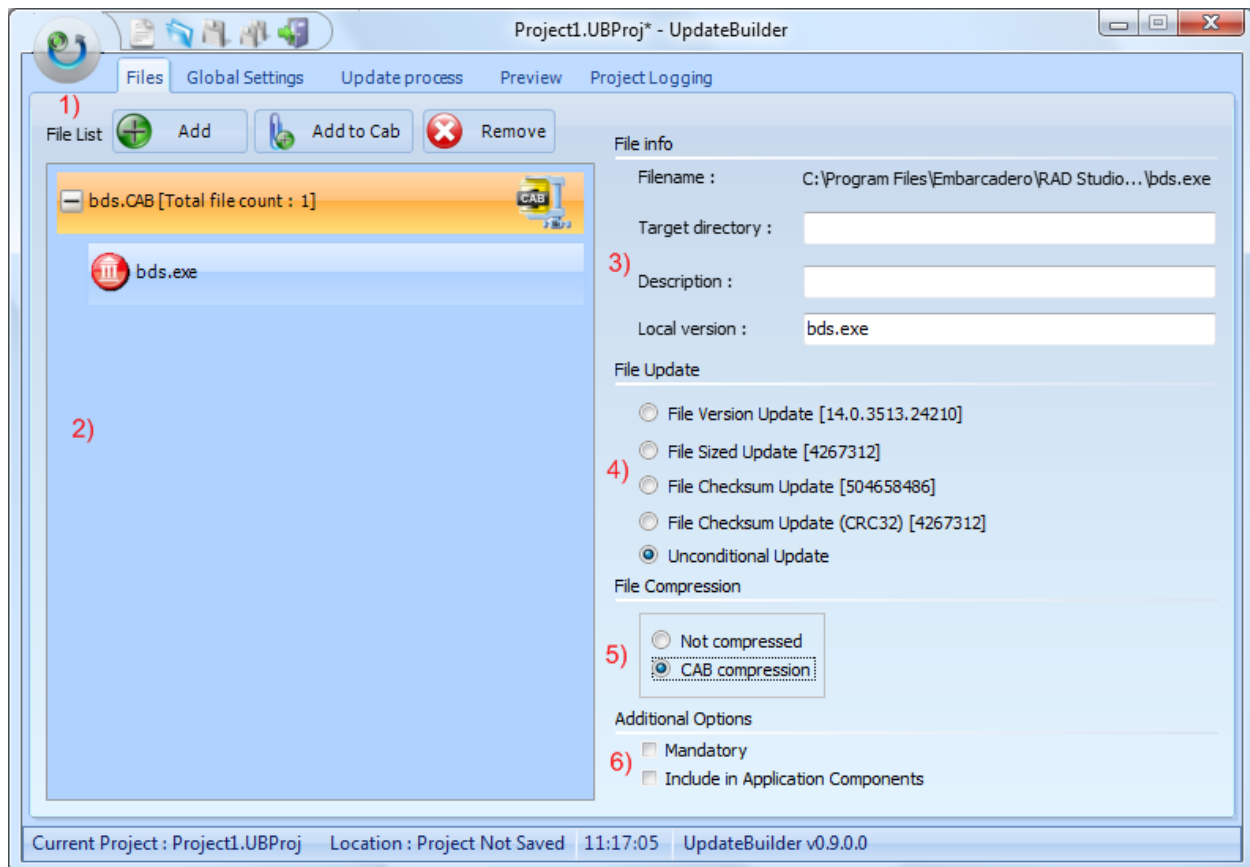
Using UpdateBuilder	3
Files	3
Global Settings - Version	5
Global Settings - Commands	6
Global Settings - Application	7
Global Settings - Upload	8
Global Settings - Custom	9
Global Settings - Update Description	10
Generated INF file	11
Update Process	12
UpdateBuilder - Console Application	14
UpdateBuilder - FinalBuilder support	14

Using UpdateBuilder

UpdateBuilder is an application that is designed to make building update control files easier. UpdateBuilder can be downloaded from <http://www.tmssoftware.com/site/wupdate.asp> It is easy to use when building an updateable application. This document is a Quick start guide that explains the most important features of this application.

Files

When starting the application the first tab that is visible is the files tab.

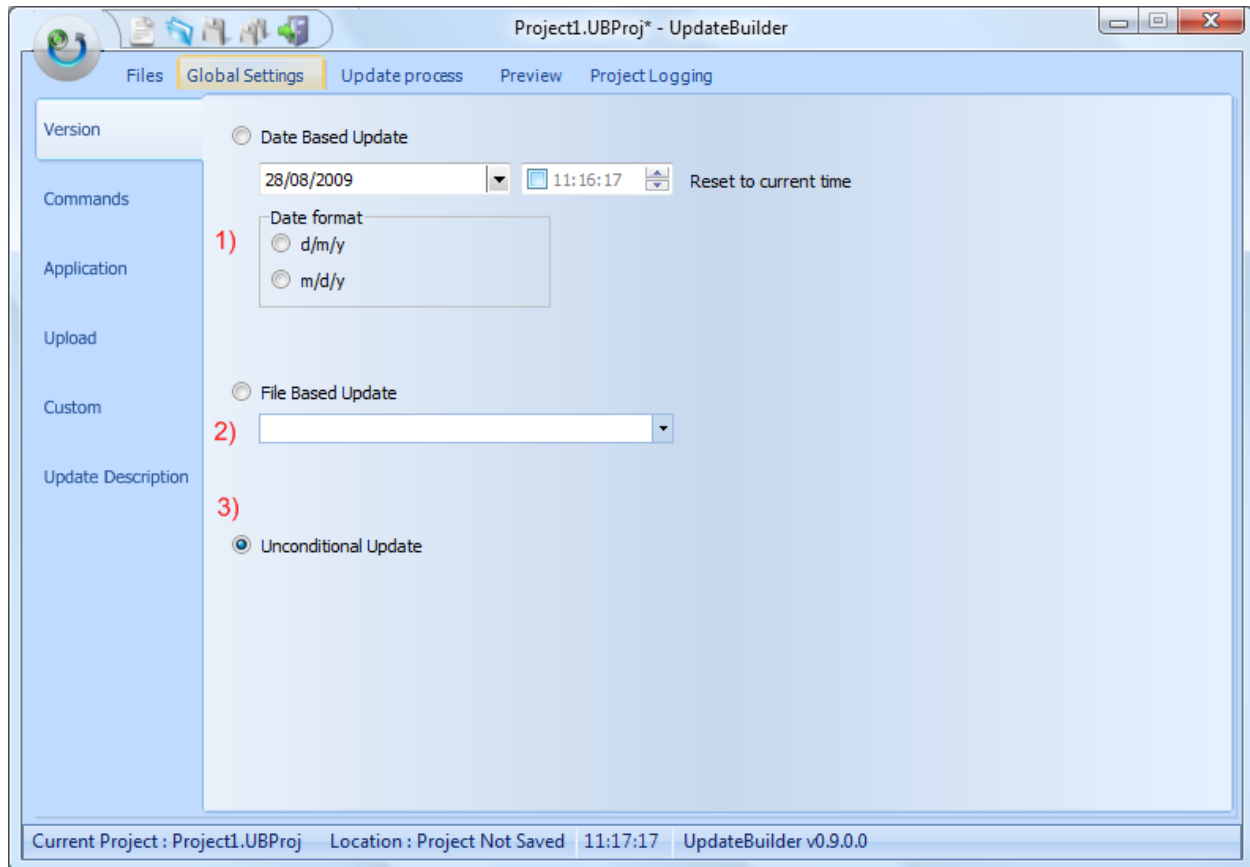


- 1) Add one or more file(s) to the file list that needs to be updated with the add button. The files can easily be removed with the Remove button and with the Add to Cab button multiple files can be compressed in a single CAB file.
- 2) The file list contains a set of files that needs to be updated. When clicking the file, the version, size, and checksum are automatically loaded from the file referenced in the project. The file list can contain single files with no compression, compressed CAB files that can have multiple sub items (indented file list items), and files with _NEW

extension that will be automatically added to the application components which is needed if the local application needs to be restarted.

- 3) After selecting a file, the filename, target directory, description and local version are extracted from the FileObject. The local version is automatically filled in as the local file is most likely to have the same filename as the new file. By default the target directory is empty, the file will be extracted to the default directory where the local application is running.
- 4) When the update process is started the new file can have a separate verification whether to update the local file or not. This can be based on the File Version, File Size, File CheckSum, or an unconditional update. In the last case the file is always downloaded.
- 5) When uploading large files, or multiple files, a CAB compression can be applied to the files. Select the main file where the update process should be based on and select the CAB file compression. The file is automatically recognized as a CAB file that will be generated in the Build process. More files can be added to the CAB file to simplify the update process by uploading only one file.
- 6) Additional options can be used to restart the application, show messages, what's new files, EULA's, etc... during the update process.

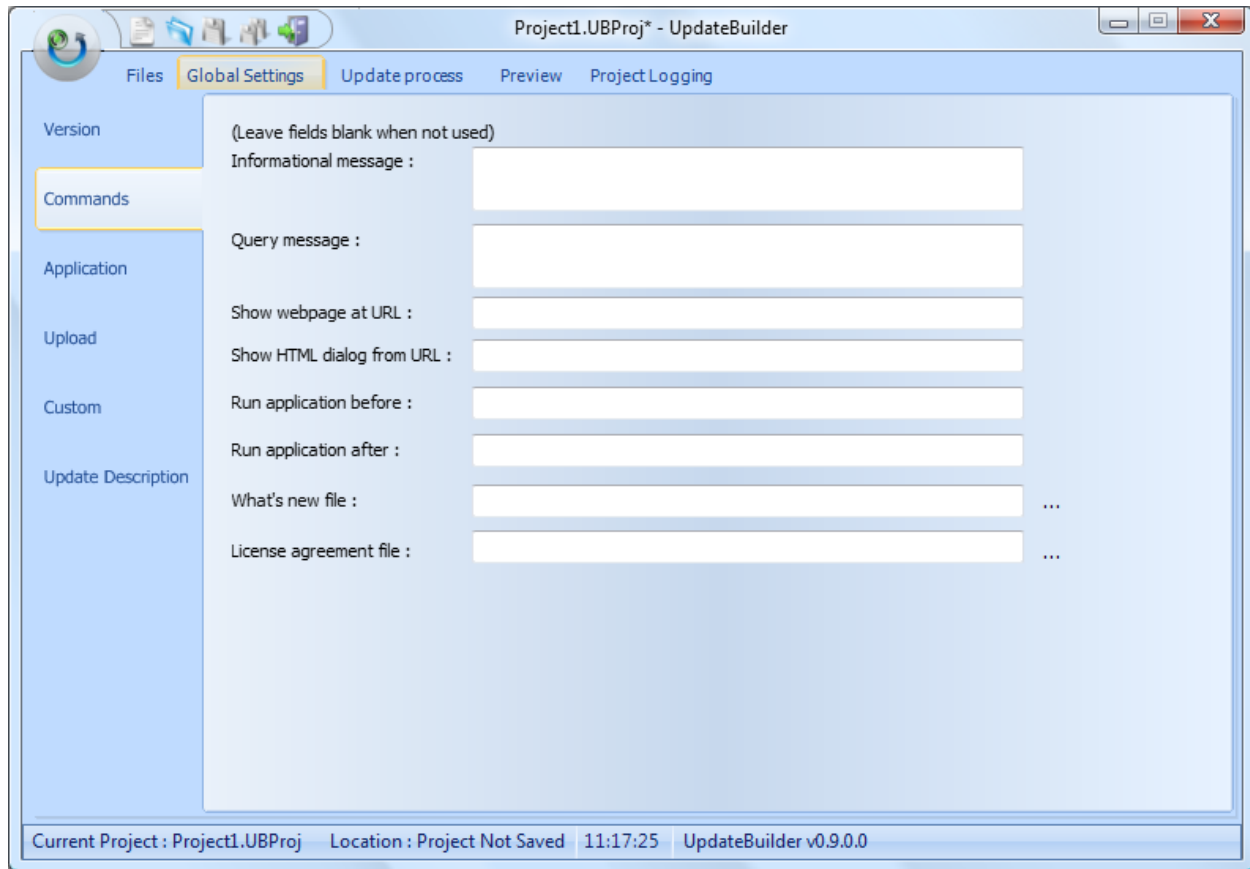
Global Settings - Version



By default the update process is unconditionally started, but in most cases it is desirable to have an extra verification whether the application should or should not be updated.

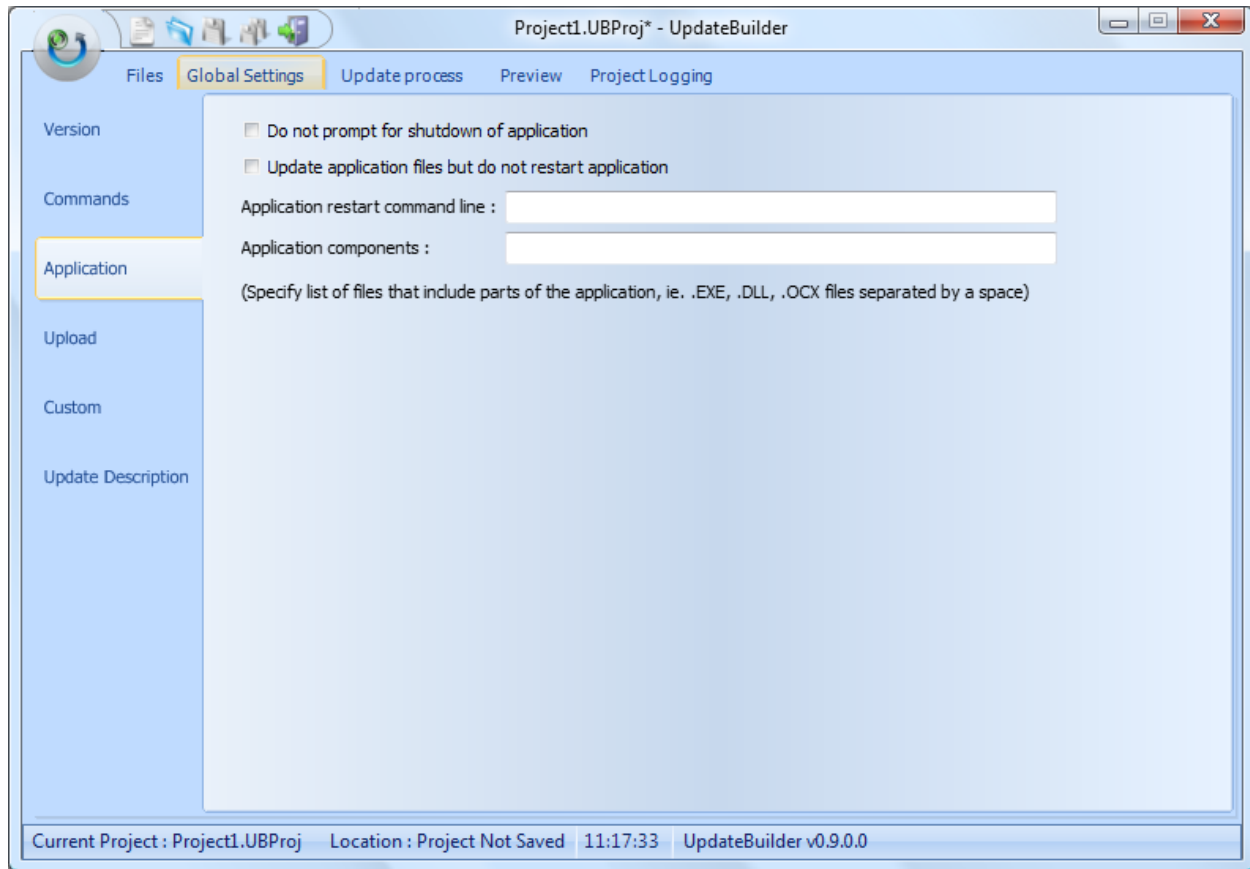
- 1) The verification to start an update process is controlled by the date (and optionally time) of the last update. When this method is chosen, you can set the date and time of the newest update available on the server.
- 2) Choosing a file based update means that the update process is controlled by a compare between a selected file on the local machine and an updated file on the server. Any file that is in the list of available updated files can be chosen as file to compare against but most typically this is the application executable.
- 3) Unconditional update

Global Settings - Commands



Some extra parameters / information / files that can be uploaded or written in the INF file. When the update process starts, the extra commands are shown.

Global Settings - Application

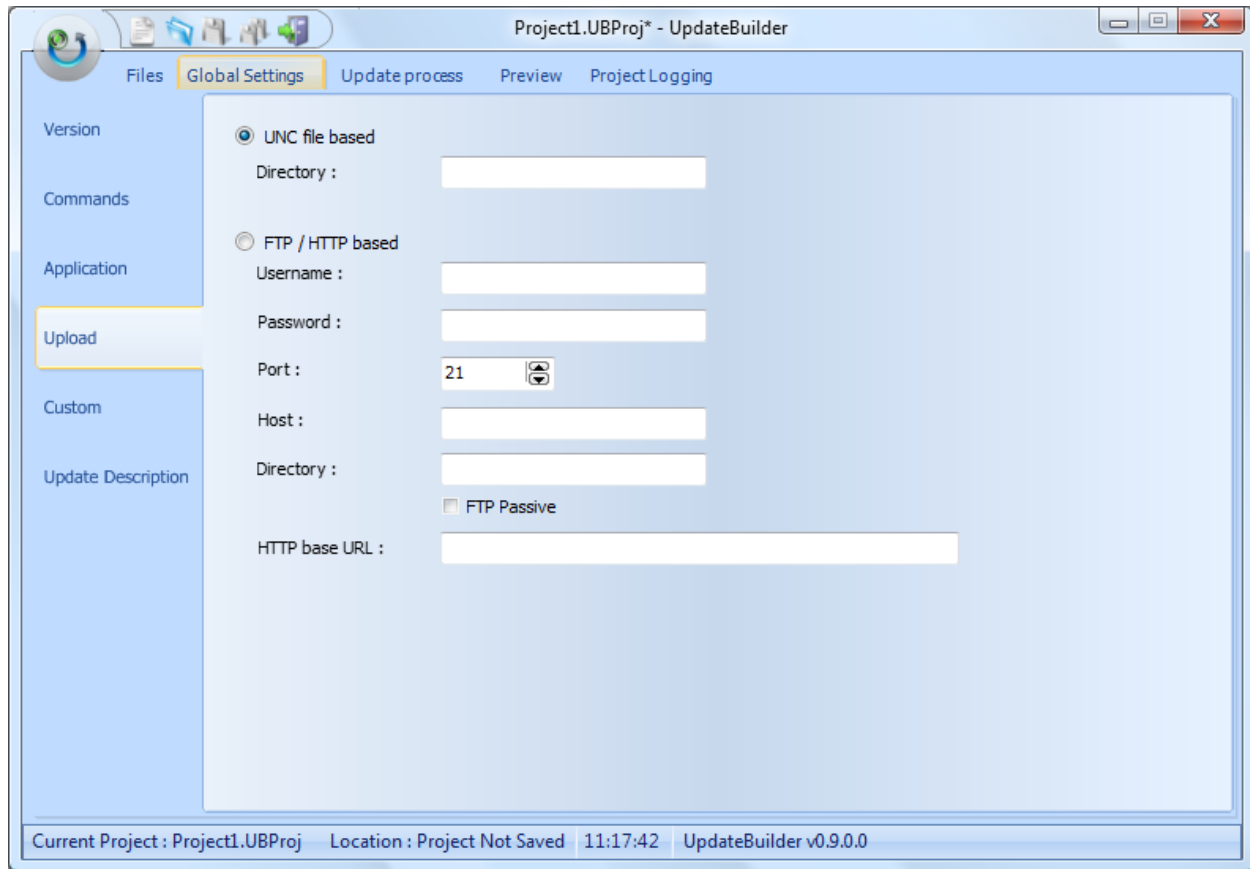


When the files that need to be updated are in use by the application, for example the executable and some dynamic link libraries, it will be required that the application is stopped before the files can be updated. All such files that require that the application is stopped are considered as “Application components”. To mark such file in the list of files of the update, check the Include in Application components option. The filename is automatically added to the Application components line in global settings.

For the application to be restarted, the Application restart command line has to be set to the local application executable. Optional command line parameters can be added here as well.

Some extra parameters can be set whether a dialog is shown to inform before shutting down the application or specify that the application does not need a restart.

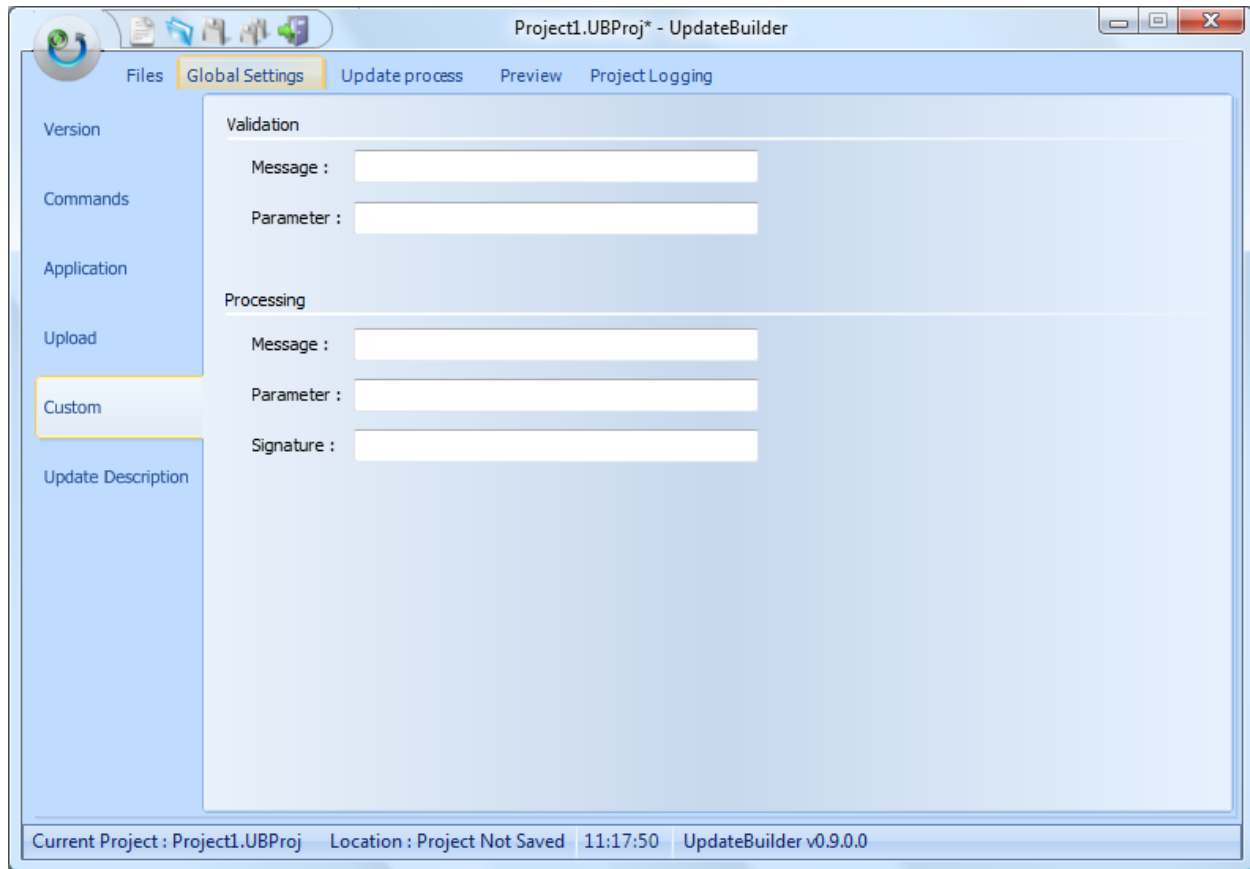
Global Settings - Upload



When building the project the files must be uploaded or copied to the specific folder where the local application will search for the INF file that describes the update process.

You can either choose for a UNC file based upload or an FTP / HTTP based upload.

Global Settings - Custom



Some extra custom validation parameters that can be set if the application implements the `OnCustomValidate` and / or `OnCustomProcess` events.

OnCustomValidate(Sender: TObject; Msg, Param: String; var Allow: Boolean);

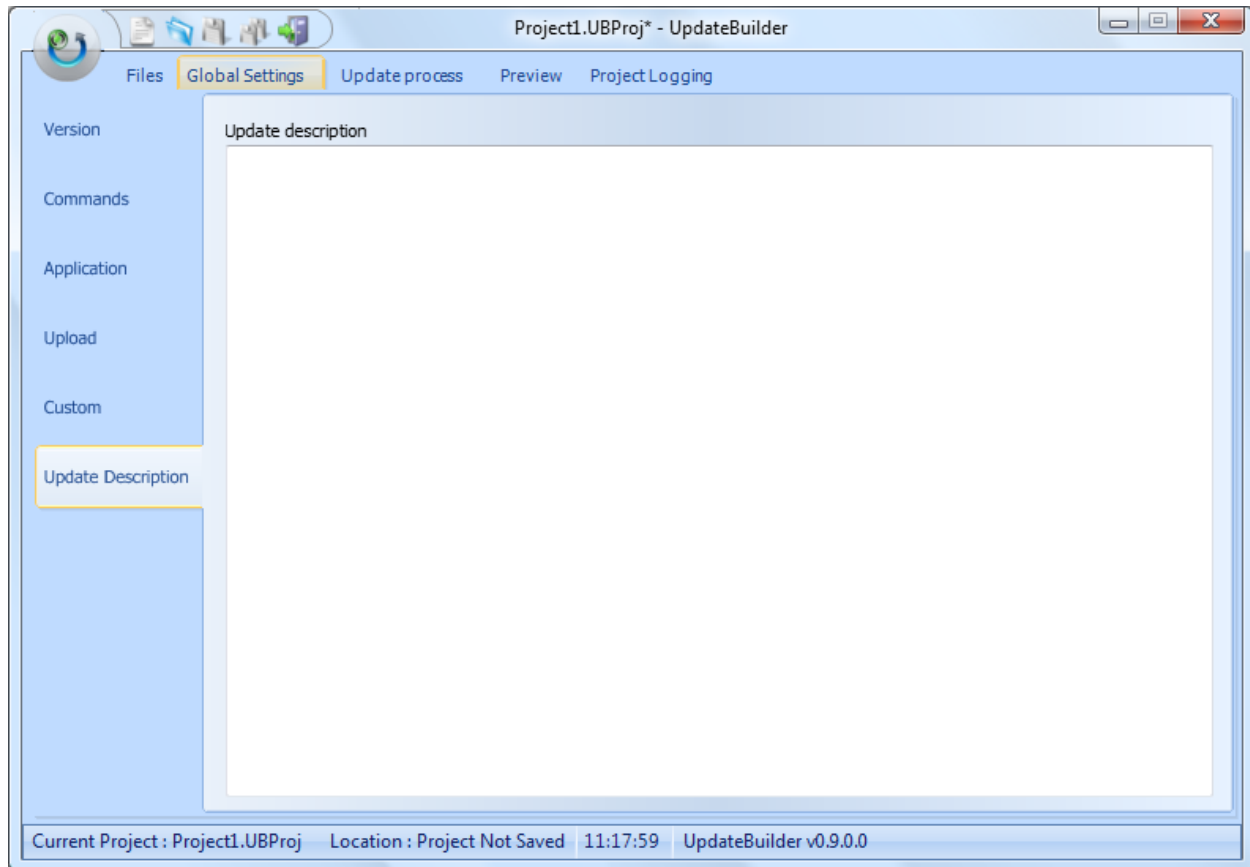
Event triggered for custom validation of parameters from the update control file. This can be used for example for custom licensing control and disallowing updates to happen for products where the free update period has expired. This event is only triggered when custom validation information is available in the control file.

OnCustomProcess(Sender: TObject; Msg, Param: String);

Event triggered when custom update processing info is present in the control file.

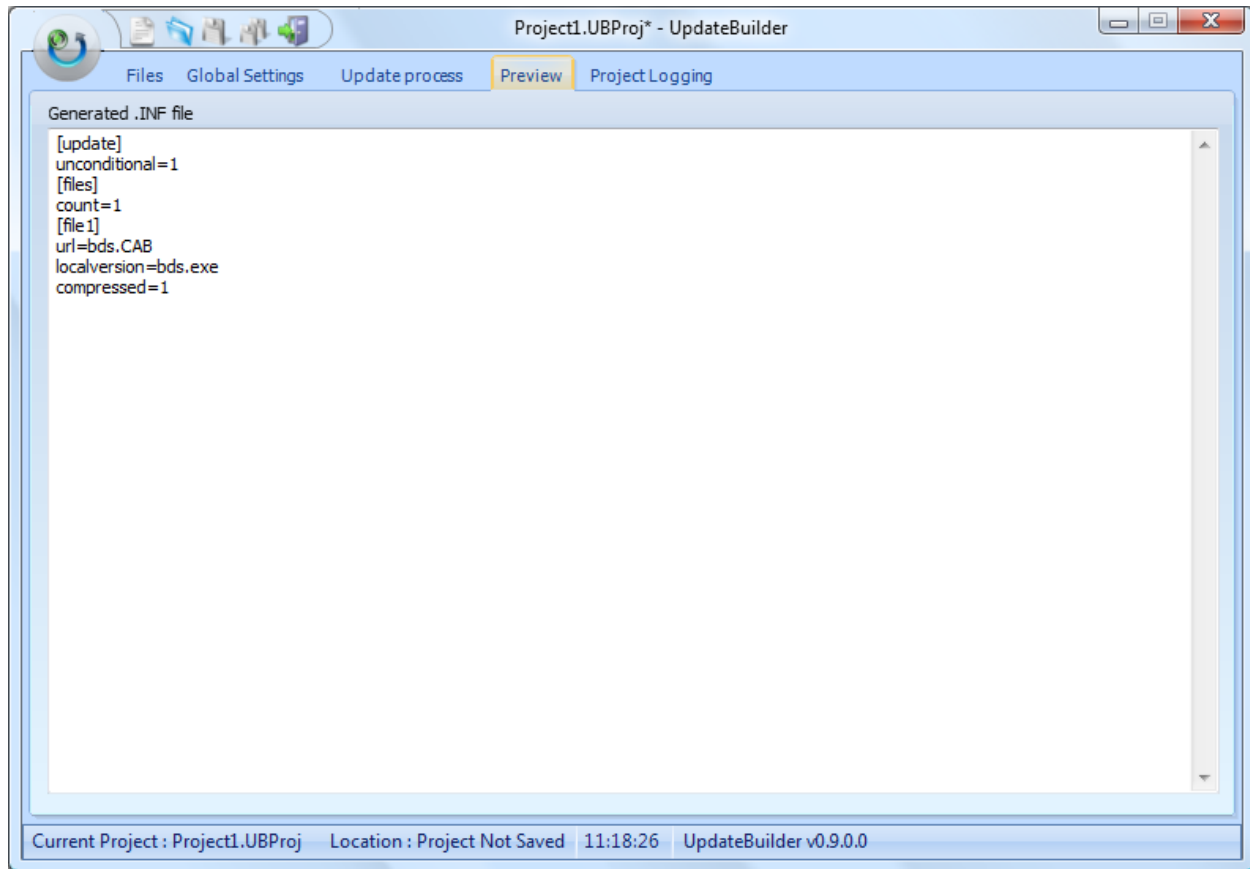
This signature value is compared against the `TWebUpdate.Signature` property when `TWebUpdate.SignatureCheck` is true. This can be used to check the integrity of the control file. When an incorrect signature is encountered, the `OnStatus` update is triggered with error parameter `ErrUpdateSignatureError`.

Global Settings - Update Description



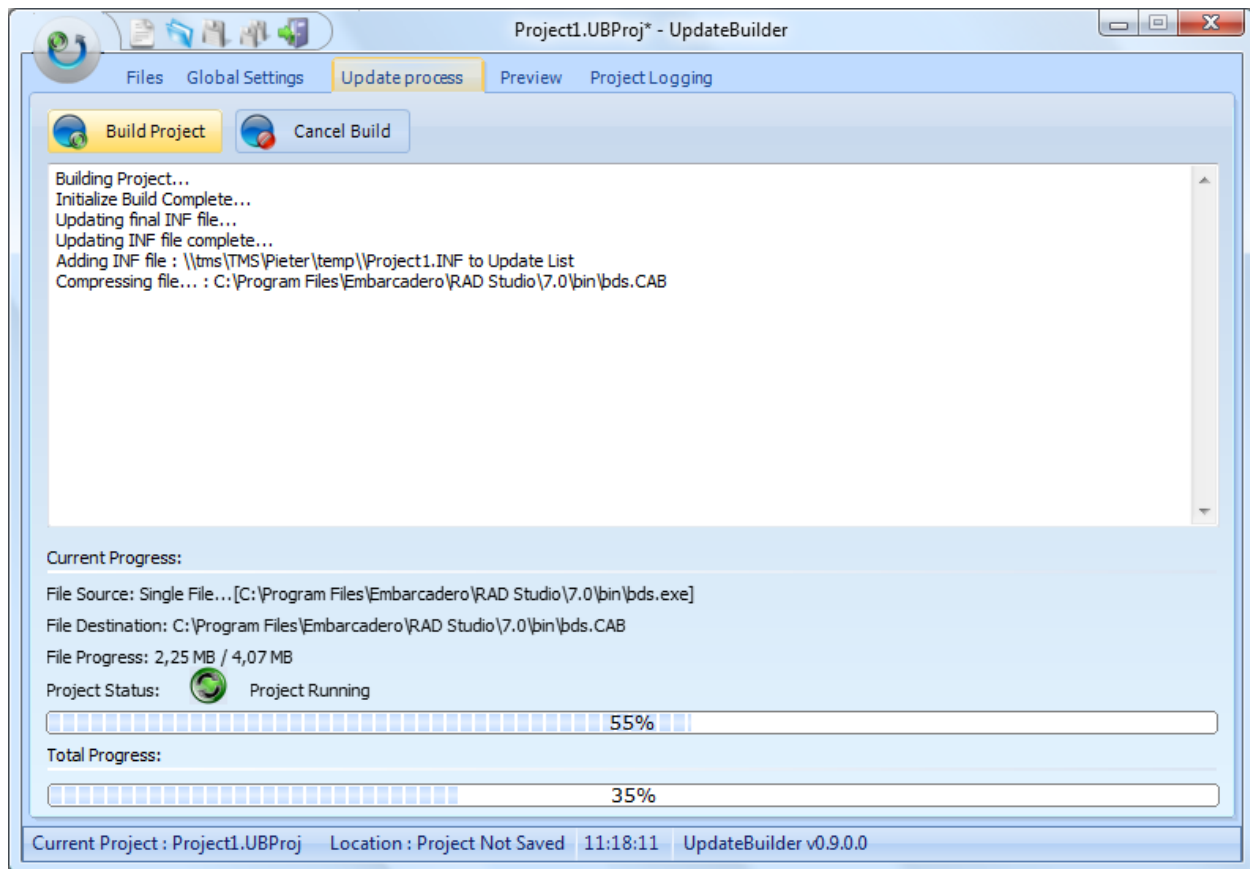
The Update description can be a version history of the new application.

Generated INF file



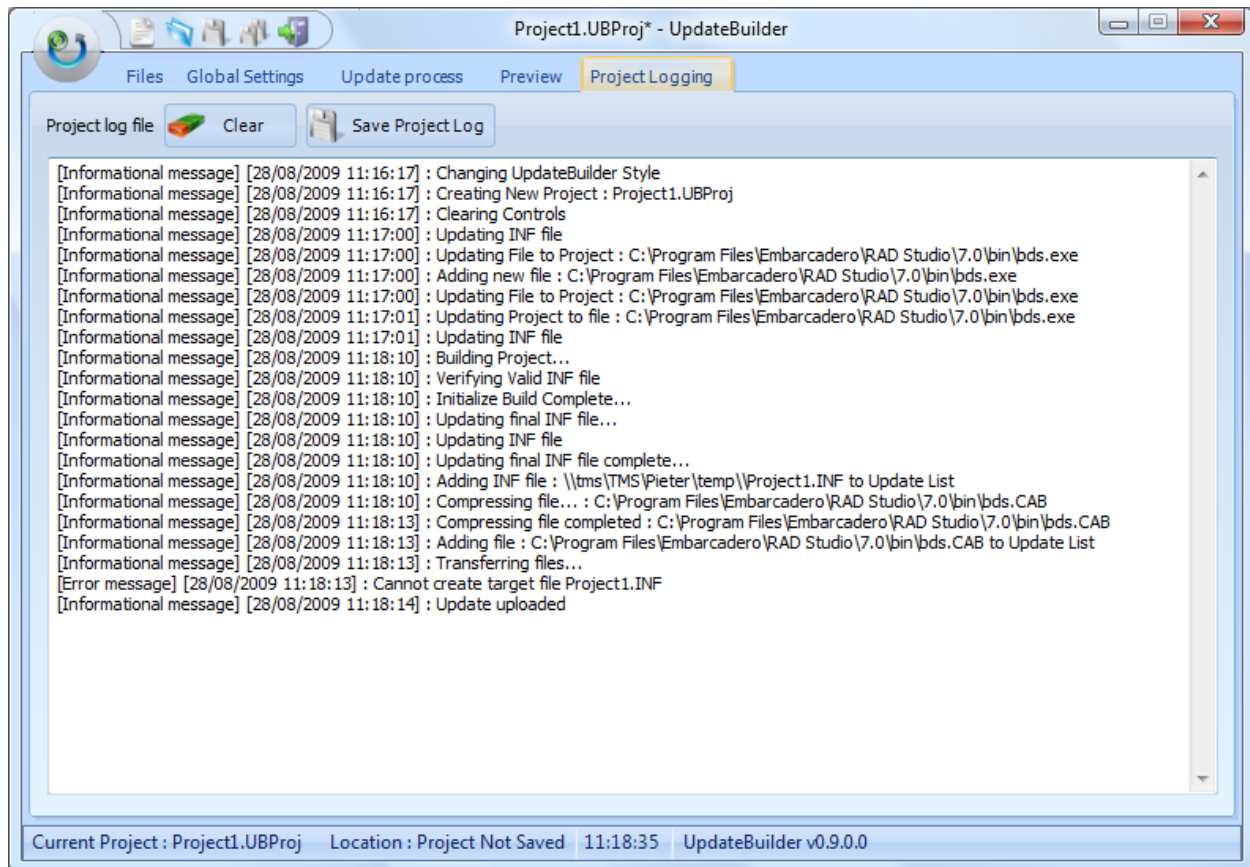
Before building the project, go to the Preview tab, where the INF file is generated for verification purposes.

Update Process



After finishing creating the update file, setting all properties and adding all necessary files, the update process can be started. Click the Build Project button to start the process. UpdateBuilder will automatically verify if all needed parameters are correct to start the build process. If errors occur during build process, a message is shown and the ability to show the log file will be given.

The Project logging tab keeps record of the most important actions during the project creation and build process



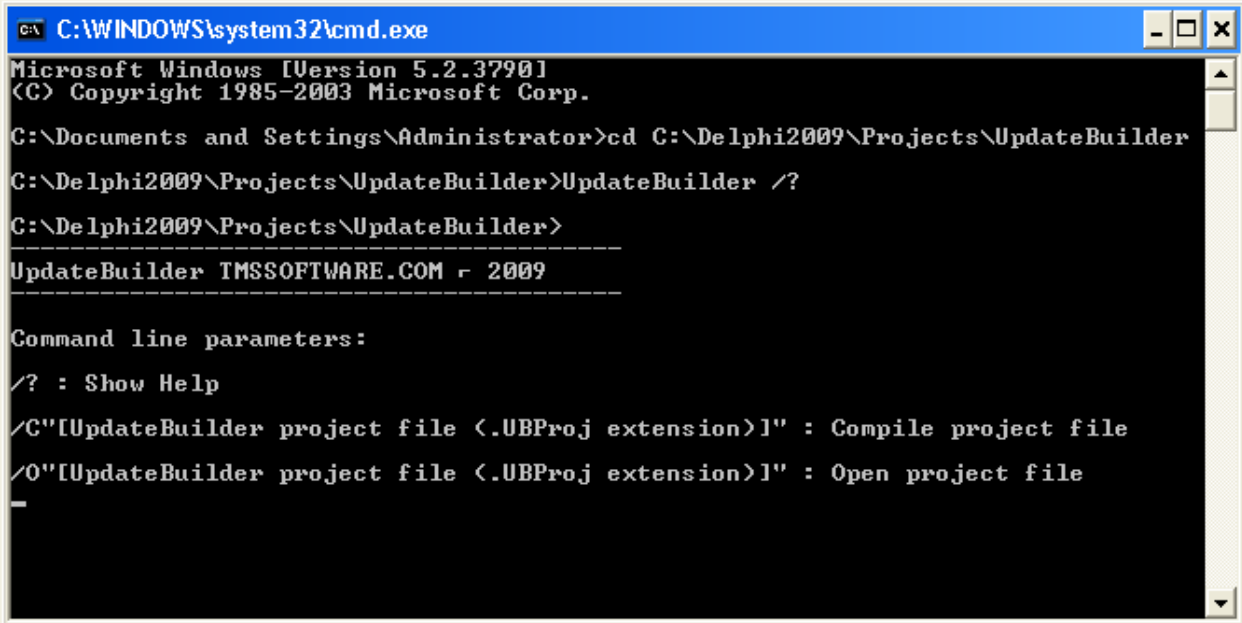
When no errors are found, and the build process is started, the CAB-files will be automatically generated and the files will be transferred to the correct UNC directory / FTP or HTTP location.

A cancel build button is active during the Build process and allows the user to cancel the process at any time.

UpdateBuilder - Console Application

UpdateBuilder is designed to quickly load and compile UpdateBuilder Projects and therefore has built-in support to compile Projects from command line.

Run “UpdateBuilder.exe /?” to show the parameters needed to compile and build the project.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\Administrator>cd C:\Delphi2009\Projects\UpdateBuilder
C:\Delphi2009\Projects\UpdateBuilder>UpdateBuilder /?
C:\Delphi2009\Projects\UpdateBuilder>
-----
UpdateBuilder TMS SOFTWARE.COM r 2009
-----

Command line parameters:
/? : Show Help
/C"[UpdateBuilder project file <.UBProj extension>]" : Compile project file
/O"[UpdateBuilder project file <.UBProj extension>]" : Open project file
-
```

UpdateBuilder - FinalBuilder support

Installation

UpdateBuilder is designed to quickly update your application with the latest version, therefore we have created a FinalBuilder plugin that allows you to integrate UpdateBuilder project compilation within FinalBuilder.

When installing UpdateBuilder an “UpdateBuilder.fbap” file copied in the installation folder. To use this plugin, copy the “UpdateBuilder.fbap” file to the ActionDefs folder where FinalBuilder is installed.

By default the path looks like:
































```
*:\Program Files\FinalBuilder *\ActionDefs
```

Note: If FinalBuilder is running, FinalBuilder must be restarted before the new plugin is loaded.


Use

To add a new “Compile UpdateBuilder Action” search for the keywords “Compile, Update, UpdateBuilder or Builder” in the FinalBuilder action list search field, or find the UpdateBuilder action in the “Compilers” category.

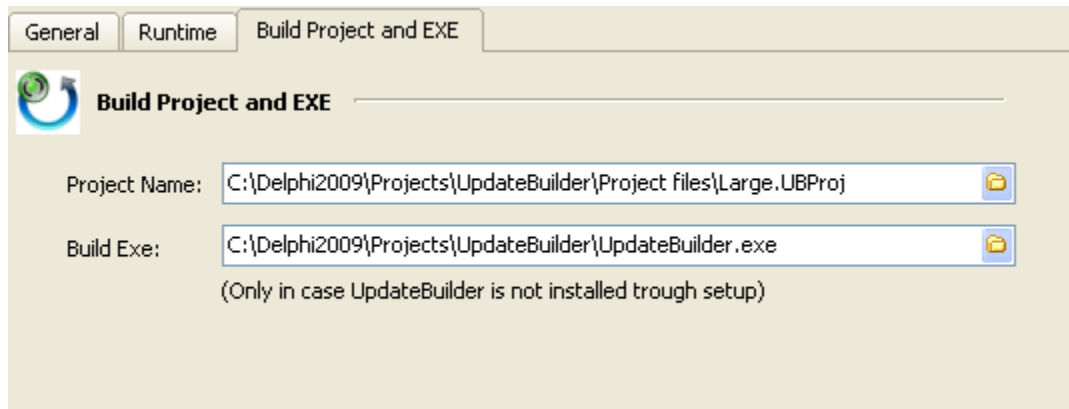
Categories

-  Frequently Used
-  .Net Tools
-  Archivers
-  Build Tools
-  CD/DVD Burning
-  COM+
-  **Compilers**
-  Custom Action Examples
-  CVS
-  Database
-  Disk Management
-  Executable Packers
-  File Sets
-  Files & Directories
-  Microsoft J# Compiler
-  Microsoft J# Project Compiler
-  Borland C# Builder Compiler
-  Compile VB6 Project
-  Compile Delphi Win32 Project
-  Compile Borland Resource Script
-  BPG File Iterator
-  Compile C++Builder Project
-  C++Builder 2006/7/8 Project
-  madExcept
-  Build VC6/Embedded Project(s)
-  Build VS.Net Solution
-  Win32 Version Info Updater
-  VS.NET Solution Iterator
-  Chrome
-  Compile Delphi for .NET Project
-  Compile UpdateBuilder Project

Select the “Compile UpdateBuilder Project” action and press enter or drag-drop the action in the main listview.

 Compile UpdateBuilder Project [C:\Delphi2009\Projects\UpdateBuilder\Project files\Large.UBProj]

Double-click or right-click -> Edit properties to show the UpdateBuilder options dialog



Select the Project in the first Field that needs to be compiled
The second field is automatically filled in if an UpdateBuilder installation is found, therefore it is advised to install UpdateBuilder with the original signed setup.exe.

When the FinalBuilder project is running, UpdateBuilder will automatically load and compile the project.

