Electric Cloud ElectricAccelerator

version 7.0

Technical Notes MS Visual Studio IDE Add-in

version 3.2.3

May 2013

This document contains information about the ElectricAccelerator integration with the Microsoft Visual Studio IDE. Topics include:

Overview

ElectricAccelerator® integrates with the Microsoft Visual Studio IDE. The ElectricAccelerator Visual Studio IDE Add-in allows you to build Visual Studio solutions and projects from within the Visual Studio IDE using Electric Make® (eMake). The add-in provides an Electric Cloud build menu and toolbar. The existing build menu remains intact for non-eMake builds.

IMPORTANT: The ElectricAccelerator Visual Studio IDE Add-in is different from the ElectricAccelerator Solution Support Add-in, which is a command line add-in used by eMake to convert Visual Studio projects into NMAKE makefiles.

Support Information

The ElectricAccelerator Visual Studio IDE Add-in supports the following versions of Visual Studio:

- Visual Studio 2012
- Visual Studio 2010
- Visual Studio 2008
- Visual Studio 2005

Note: The Visual Studio 2010 and 2012 add-in does not currently support Xbox builds, Windows Mobile configurations, or Custom build rules.

New Features and Improvements

for 3.2.3

- Added a new menu item that enables you to build with generated makefiles locally. (VSP-601)

for 3.2.2

- Corrected an issue that resulted in Exception of type 'System.Exception' thrown (VSP-584).
- Corrected an issue that caused the add-in to return Error HRESULT_FAIL (VSP-582).
- The add-in can now parse @ECLOUD_BUILD_ID@ when it is used in an annotation file path (VSP-578).
- Fixed the registration of the add-in for Visual Studio 2012 (VSP-570).

Known Issues

- The ElectricAccelerator Visual Studio IDE Add-in does not run on a system that has never had Visual Studio 2005 installed on it. On such systems, the IDE add-in throws an exception similar to the following: 3:Error: Adding Build menu item: Could not load file or assembly 'stdole, Version=7.0.3300.0, Culture=neutral, PublicKeyToken=b03f5f7f11d50a3a' or one of its dependencies. The system cannot find the file specified.

  Workaround: Install the Office 2003 Update: Redistributable Primary Interop Assemblies (http://www.microsoft.com/download/en/details.aspx?DisplayLang=en&id=20923). (VSP-510)

Installation

You must install Electric Make on the build machine for the add-in to work properly.

Installation for all Windows platforms

If you used a previous version of the Visual Studio IDE Add-in installer to perform a cluster upgrade, launching the installer may display a dialog stating “invalid command name...”

Workaround: Uninstall the existing Visual Studio IDE Add-in and then rerun the installer.

Installing the Visual Studio IDE Add-in

To install the Visual Studio IDE Add-in locally (you can also install the Solution Support Add-in locally), run the installer provided.

1. Run the `VSAddIn-<version>-Install.exe` file.
2. Welcome screen—click **Next**.
3. Choose Destination Location screen—accept the default install location or click **Browse** to change the location. Click **Next**.
4. Setup Type screen—select the setup type:
   - ElectricAccelerator VS IDE Add-in Local Install
   - ElectricAccelerator Solution Support Add-in Local Install
   - Custom (This option allows you to select multiple setup types from this list.)

   Click **Next**.
5. Start Copying Files screen—review your settings before continuing the installation. Click **Next** to continue or **Back** to make changes.
6. When the wizard displays “Install finished,” your installation is complete. Click **Finish** to close the installer.

The installation log file is in the install directory’s root, `C:\ECloud` by default.

Note: Due to an issue with previous versions’ uninstallers, an upgrade may cause the following error message: “Cannot find script file: C:\ECloud\i686_win32\bin\unregaddin.vbs.” You can safely ignore this message.

Finish all Visual Studio installations before installing the Electric Cloud IDE Add-in. Adding a new language to an existing Visual Studio installation with the Electric Cloud IDE Add-in causes Visual Studio to display an empty Electric Cloud menu. The workaround is to reinstall the add-in.

Installation Options

Use this structure for options: `<Install filename> [options]`

The following options are available to customize your installation:

Option	Description
`/help`	Displays help information.
`/mode [ARG]`	Sets the installation mode. Available values: `standard` or `silent`.

Option	Description
`/prefix [ARG]`	Sets the installation directory.
`/response-file [ARG]`	The file from which to read installer responses.
`/save-response-file [ARG]`	The file to which installer responses are written when the installer exits.
`/temp [ARG]`	Sets the temporary directory used by the program.
`/type [ARG]`	Performs the selected type of installation. Available values: `addin` or `uiaddin`.
`/version`	Displays installer version information.

Using response files with silent installs

A response file is a file that defines installation parameters. These parameters are the same values a user would normally set through the installer interface or command line.

To create a response file and then use it in multiple silent installs:

1. Run an installation with the `/save-response-file <filename>` option and your desired settings.

   This creates the response file in the directory where you ran the installer.

2. Use the resulting response file for silent identical installs by using the `/response-file <filename>` and `/mode silent` options.

Important Notes

Maximum PDBs

`ECADDIN_MAX_PDB_FILES` is now set to 16 by default. If you have fewer than 16 agents, you can decrease this value to be equal to or less than the number of agents.

Electric Make

The add-in checks for the presence of eMake when Visual Studio starts. If it cannot be found, its Build/Rebuild/Clean functions are disabled.

When eMake is run from Visual Studio, it must be run through an intermediate application named `ecspawn.exe`. This program ensures that eMake responds correctly to CTRL-C and that child processes are grouped together. This application is displayed in the Task Manager and should not be terminated; it stops when the build finishes or when the build is canceled.

Do not run non-eMake builds while running eMake builds and vice-versa.

ElectricInsight

The add-in checks for the presence ElectricInsight when Visual Studio starts. If it cannot be found, the Run ElectricInsight function is disabled.

When you run ElectricInsight from Visual Studio, Visual Studio looks for the currently running instance of `einsight`. In this case, the annotation file is not loaded (or reloaded). Manually open the annotation file from ElectricInsight, or close ElectricInsight and select Run ElectricInsight again.

Solving common issues

If you encounter issues, make sure you have done the following:

- Initialize Visual Studio

  Use the psexec method to initialize Visual Studio as shown:

  ```
  psexec -u ECloudInternalUser1 "C:\Program Files\Microsoft Visual Studio 8\
  Common7\IDE\devenv.exe"
  ```

  As an alternative, disable profiles for Visual Studio by running this regedit script:

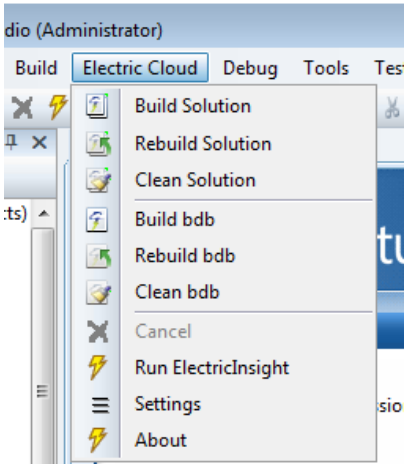
  ```
  REGEDIT4
  [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\VisualStudio\8.0\Profile]
  "AppidSupportsProfiles"="0"
  ```

- Disable the Windows error reporting service on the agent/EFS hosts.This avoids popup windows for crashed applications.
- Set the maximum number of parallel project builds to 1.
- Initialize the Customer Experience Improvement Program.

Using the Visual Studio IDE Add-in Interface

Main Menu and Toolbar

When you run Visual Studio, you are presented with the Electric Cloud main menu and toolbar (displayed in the following screenshot):



The menu has the following functions:

- Build Solution - Builds the current solution with eMake using the Cluster Manager specified and/or local agents.
- Build Solution Locally - Builds the current solution locally with eMake but ***without using*** remote agents or local agents (this function is equivalent to turning off the Cluster Manager and local agents). This function is hidden by default. See "Build Solution Locally" on page 6 for additional information about this function.
- Rebuild Solution - Rebuilds the current solution.

- Clean Solution - Cleans the current solution.
- Build <*project*> - Builds the current project or selection.
- Rebuild <*project*> - Rebuilds the current project or selection.
- Clean <*project*> - Cleans the project or selection.
- Cancel - Cancels a running eMake build.
- Run ElectricInsight - Runs ElectricInsight with the current annotation file (if it exists).
- Settings - Opens the solution settings dialog.
- About - Displays add-in information.

The build options use the solution settings and Tools > Options settings to create the eMake command. The add-in creates a makefile [`<solution name>.ecmak`] with the `devenv` call and calls eMake using the environment and command-line options specified.

An example solution rebuild:

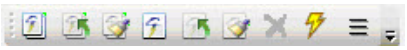
```
mysolution.ecmak:
all:
	devenv C:/mysolution.sln /rebuild Debug /useenv
```

The project and configuration are taken from the current context. The command is dependent on the menu item.

When a build is running, you can cancel it by selecting Cancel. Cancel is available only during a running build, rebuild, or clean.

At any time, you may run ElectricInsight to view the annotation file. ElectricInsight loads the specified annotation file or defaults to `emake.xml`. If ElectricInsight is already running, it gets the focus. In this case, you must manually reopen the file. The Settings selection displays the solution settings dialog.

The toolbar provides the same functionality as the Electric Cloud main menu and is customizable.

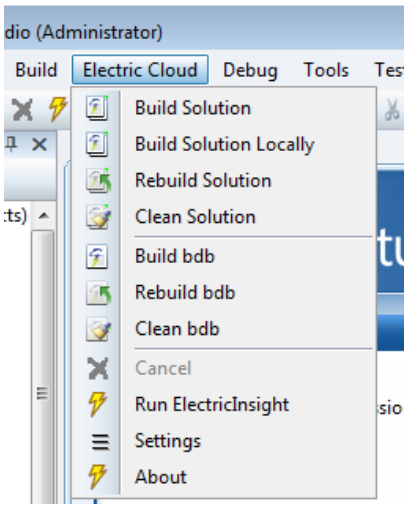


Build Solution Locally

You can choose to build a solution locally with eMake but ***without using*** remote agents or local agents. You may want to use this function if a distributed incremental build is slow, or if a local Visual Studio incremental build causes unnecessary rebuilding of objects.

To make this function visible in the menu, set the environment variable `ECUIADDIN_LOCAL_BUILD=true`.

The following screenshot illustrates the menu with the Build Solution Locally function.

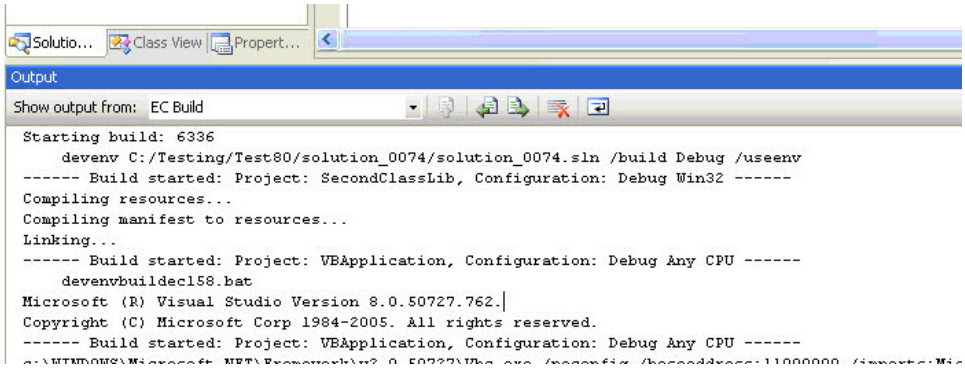


Advisories for Build Solution Locally

- The eMake local build does ***not*** support autodepend. This means changes in header files may not cause dependent source files to be recompiled.
- The eMake local build does ***not*** produce an annotation.
- Because history is not generated, unexpected conflicts may occur on subsequent eMake cluster builds.

Output Pane

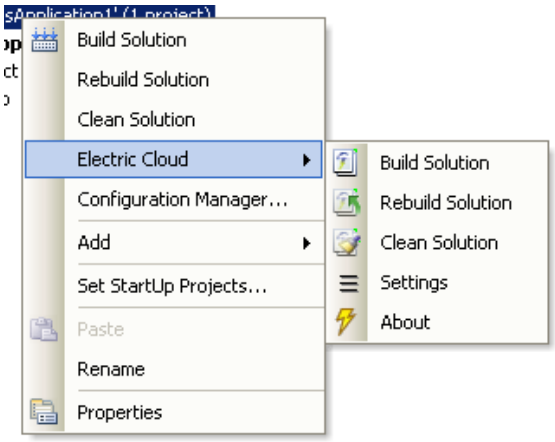
Output from an eMake build is displayed in the EC Build output pane (displayed in the following screenshot).



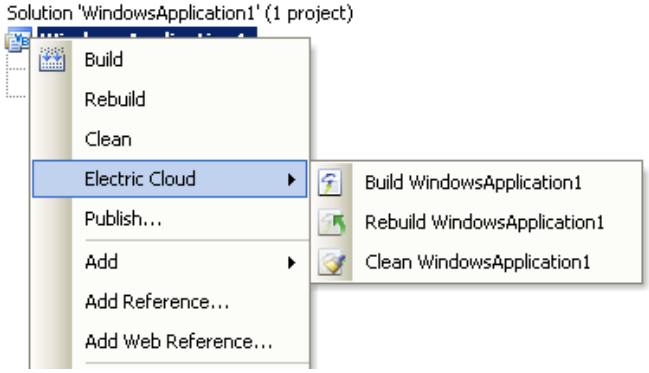
Context-Sensitive Menus

The add-in provides additional context-sensitive menus.

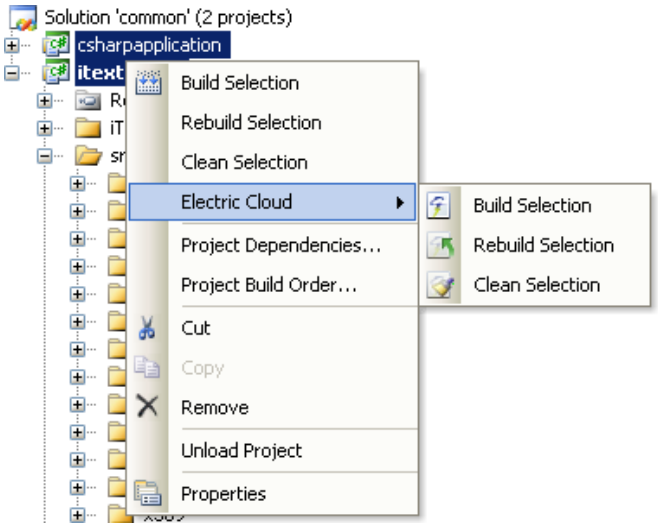
The following screenshot illustrates the Solution menu.



The following screenshot illustrates the Project menu.



The following screenshot illustrates the Selection menu.



Add-in Settings

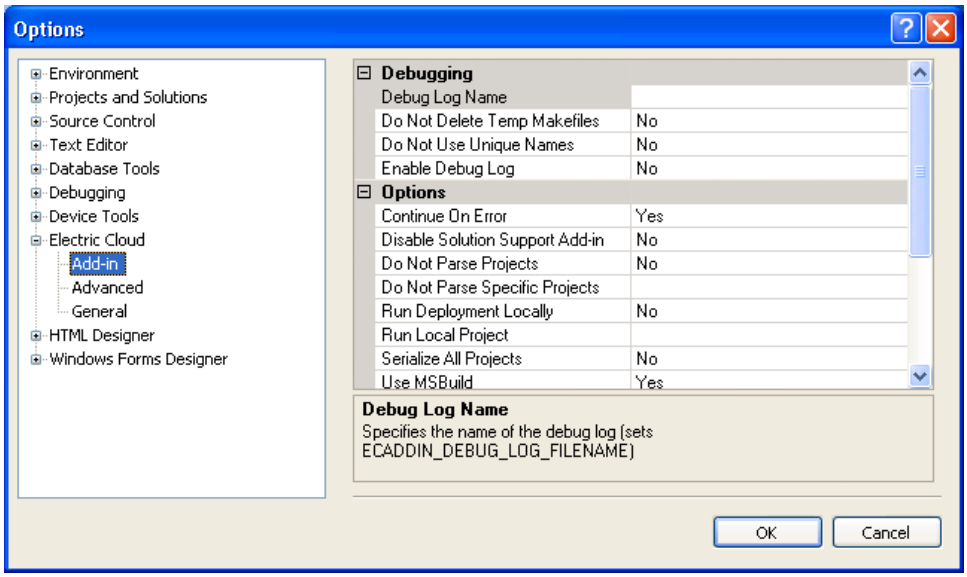
Note: See the *ElectricAccelerator Solution Support Add-in Technical Notes* for environment variable descriptions.

Global Options

Global options are stored in the standard Visual Studio Tools > Options dialog under the Electric Cloud entry. Three panes are available: Add-in, Advanced, and General. This information is stored in the registry under: HKCU\SOFTWARE\Electric Cloud\ECUIAddIn

Add-in pane

The following screenshot illustrates the Add-in pane.



Debugging

- Debug Log Name - Specify the name of the debug log (sets `ECADDIN_DEBUG_LOG_FILENAME`). The default is `C:\ecdebug<unique>.log`.

  Note: This file represents a file on the agent (not the build computer) that parses the solution.

- Do Not Delete Temp Makefiles - Do not delete temporary makefiles when the build completes (sets `ECADDIN_DONT_RM_TMP_MAKEFILES=true`).
- Do Not Use Unique Names - Do not use unique names for temporary files (sets `ECADDIN_DONT_USE_UNIQUE=true`).
- Enable Debug Log - Enable debug logging (sets `ECADDIN_DEBUG=true`).

Options

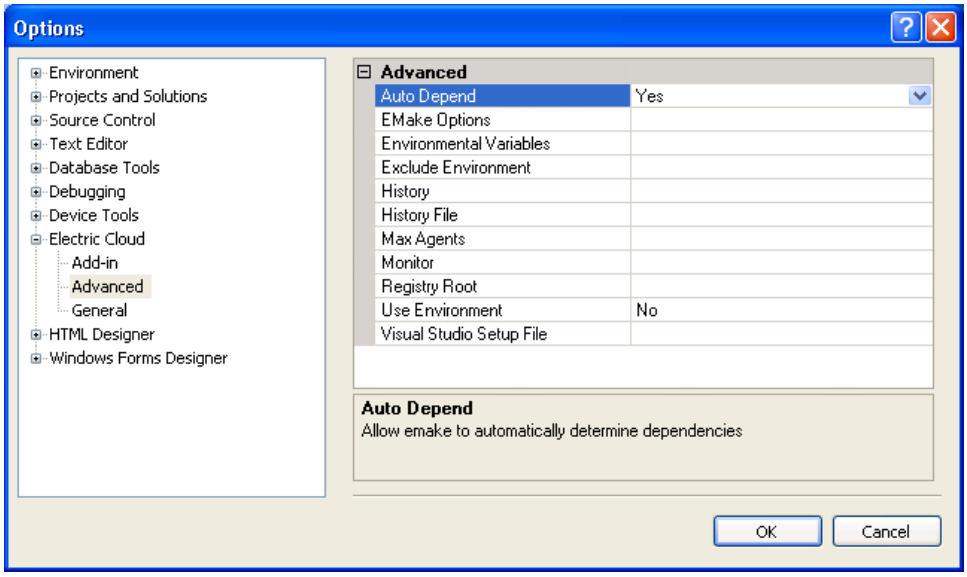
- Continue On Error - Continue when an error occurs (sets `ECADDIN_CONTINUE_ON_ERROR=true`, which adds `/I` to the eMake call).
- Disable Solution Support Add-in - Disable the Solution Support Add-in (sets `ECADDIN_DONT_USE=true`).
- Do Not Parse Projects - Prevent the Solution Support Add-in from breaking up C++ projects (sets `ECADDIN_DONT_PARSE_PROJECTS=true`).
- Do Not Parse Specific Projects - Prevents the Solution Support Add-in from breaking up specified C++ projects (sets `ECADDIN_DONT_PARSE_PROJECT`).
- Run Deployment Locally - Run deployment projects locally using `#pragma runlocal` (sets `ECADDIN_RUN_DEPLOYMENT_PROJECTS_LOCALLY=true`).
- Run Local Project - Run specified projects locally using `#pragma runlocal` (sets `ECADDIN_RUN_LOCAL_PROJECT`).
- Serialize All Projects - Serialize all projects using `#serialize` (sets `ECADDIN_SERIALIZE=true`).
- Use MSBuild - Use MSBuild internally for unparsed projects (sets `ECADDIN_USE_MSBUILD=true`).

Performance

- Add Implicit Dependencies - Add dependencies to improve first-time build speed (sets `ECADDIN_ADD_IMPLICIT_PDB_DEPENDENCIES=true`)
- Always Rescan Solution - Always recreate temporary makefiles even if the solution has not changed.
- Enable Incremental Link - Enable incremental linking (sets `ECADDIN_ENABLE_INCREMENTAL_LINK=true`).
- Force /Z7 - Force compiler option /Z7 (sets `ECADDIN_FORCE_Z7`).
- Maximum PDB Files - Maximum number of PDB files used when splitting (sets `ECADDIN_MAX_PDB_FILES`).
- Remove Dependencies - Remove dependencies and references to prevent Visual Studio from building dependent projects (sets `ECADDIN_REMOVE_DEPENDENCIES=true`).

Advanced pane

The following screenshot illustrates the Advanced pane.

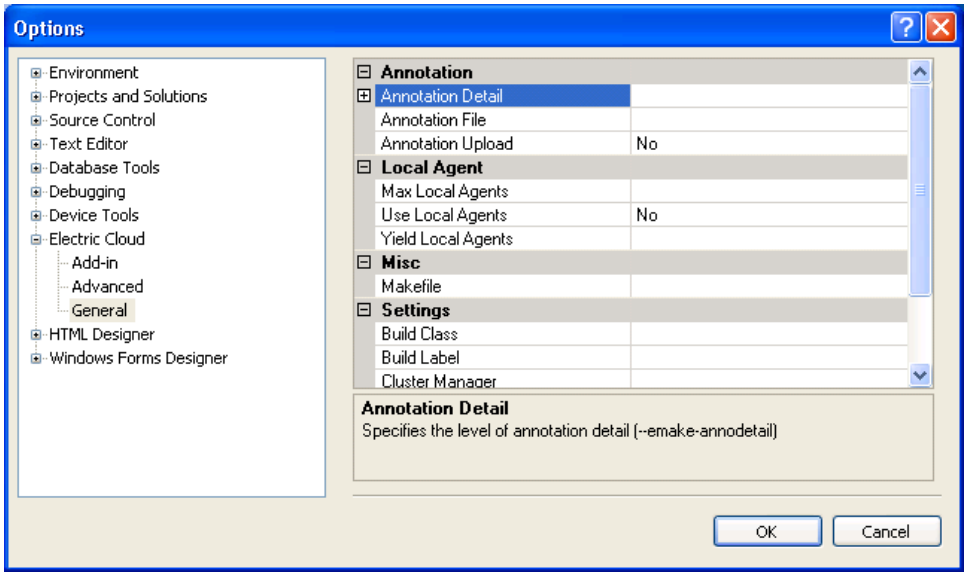


Advanced

- Auto Depend - Allows eMake to automatically determine dependencies. The default is **Yes**.
- EMake Options - A list of eMake options in the form `--emake-<option>=<value>` separated by a carriage return.
- Environmental Variables - A list of environment variables in the form `<variable>=<value>` separated by a carriage return. Do not use “set”.
- Exclude Environment - A list of environment variables to exclude from eMake (`--emake-exclude-env`), separated by ‘:’ [a colon].
- History - Specifies the emake history option (`--emake-history`). Values can be create, merge, or read.
- History File - Specifies the history file to use (`--emake-historyfile`). The default is `eMake.data`.
- Max Agents - Specifies the maximum number of agents to use during the build (`--emake-maxagents`).
- Monitor - Allows the build to be monitored by ElectricInsight (`--emake-monitor`).
- Registry Root - Specifies the registry root (`--emake-reg-roots`). You can specify multiple roots separated by ‘:’ [a colon].
- Use Environment - Determines whether to add `/useenv` to the `devenv` call.
- Visual Studio Setup file - Visual Studio setup file for command line builds (default is `vsvars32.bat`).

General pane

The following screenshot illustrates the General pane.



Annotation

- Annotation Detail - Specifies the level of annotation detail (`--emake-annodetail`) from the following:
  - Basic
  - Environment
  - File
  - History
  - Lookup
  - Registry
  - Waiting
- Annotation File - Specifies the annotation file (`--emake-annofile`). Required if annotation detail is set.
- Annotation Upload - Specifies whether to upload the annotation file (`--emake-annoupload`).

Local Agent

- Max Local Agents - Specifies the maximum number of local agents (`--emake-maxlocalagents`).
- Use Local Agents - Switches on local agents (`--emake-localagents`).
- Yield Local Agents - If using more than `N` local agents, then eMake releases the number agents over `N` every `T` seconds so they can be used by another eMake that is looking for local agents (`--emake-yield-localagents=N,T`).

Misc

- Makefile - Specifies a makefile to use rather than the default that the add-in generates.

Settings

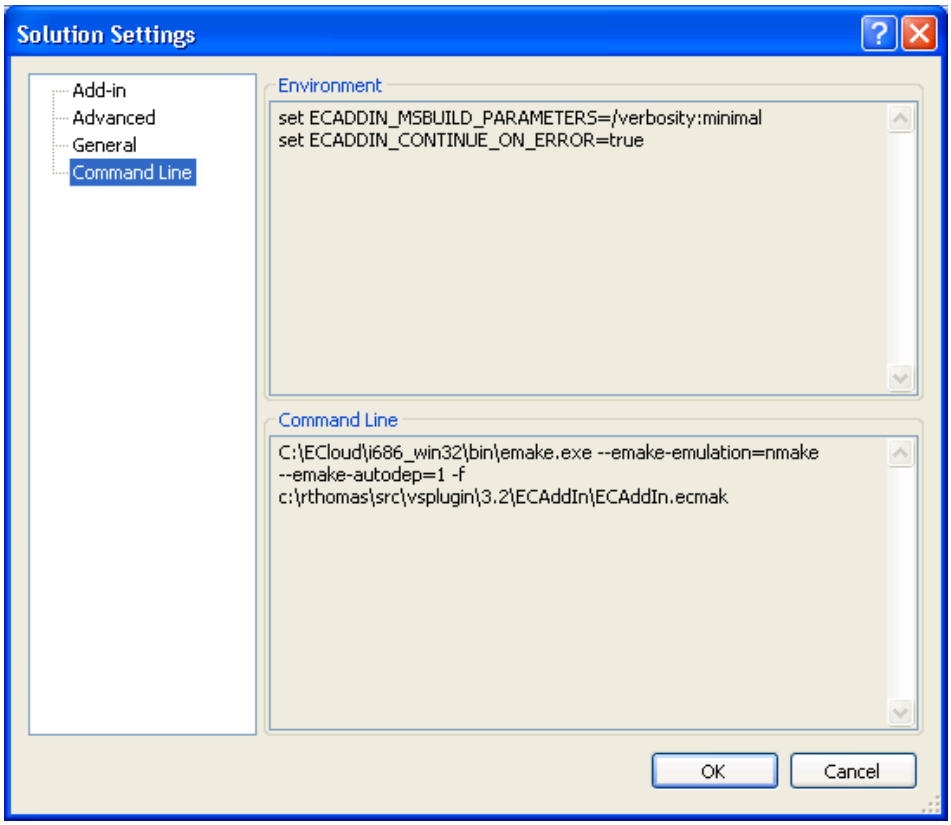
- Build Class - Specifies the build class (`--emake-class`).
- Build Label - Specifies the build label (`--emake-build-label`).
- Cluster Manager - The eMake Cluster Manager (`--emake-cm`). If this field is empty, an eMake build is performed with local agents when Use Local Agents is selected. When Use Local Agents is not selected, a local eMake build (without remote or local agents) is performed.
- Resource - Specifies the build resource (`--emake-resource`).
- Root - Specifies the eMake root (`--emake-root`). You can specify multiple paths separated by ':' [a colon].
- Use 64-bit eMake - Use the 64-bit version of eMake.

Solution Settings

Four settings panes are available: Add-in, General, Advanced, and Command Line.

For the Add-in, General, and Advanced panes, available fields and their descriptions are analogous to those in global options.

The following screenshot illustrates the `Command Line` pane.



This pane is read-only. It shows a preview of the environment settings and the command line arguments that will be used with eMake.

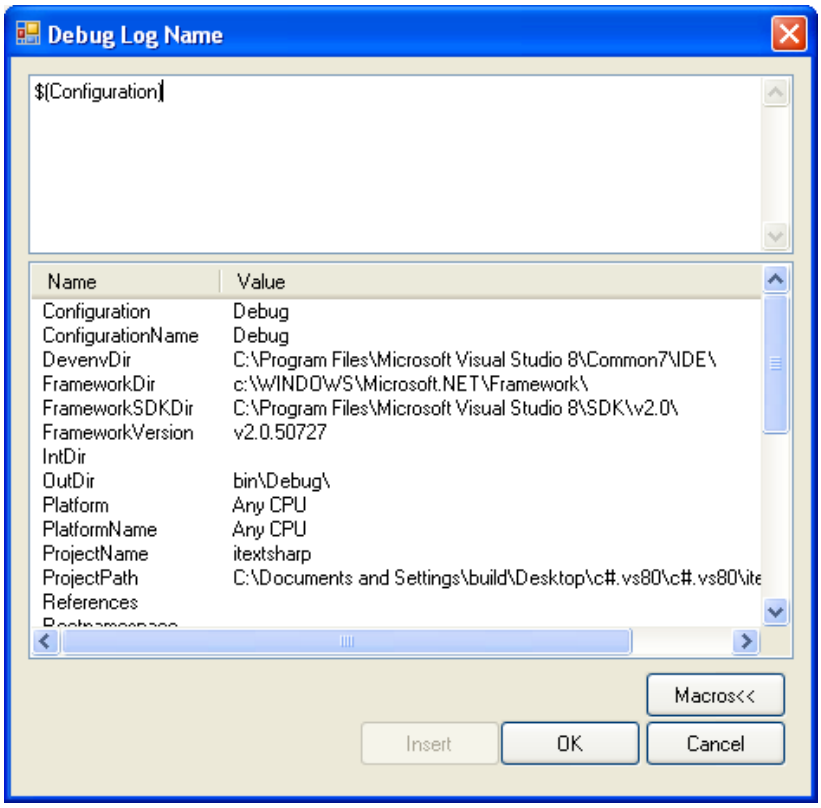
Inheriting Options

The add-in inherits its global options (Tools > Options) from the environment settings unless you explicitly set the options. Local options (Electric Cloud > Settings) inherit global options in the same manner.

To inherit global options, leave the corresponding solution settings field blank or select <inherit from global defaults>.

Using Macros

If file name settings include any variables that contain invalid DOS file name characters, such as a '\' [a backslash] or ':' [a colon], this will result in an error at run-time.



Uninstalling the Visual Studio IDE Add-in

To uninstall the Visual Studio IDE Add-in, go to the Control Panel >Add/Remove programs and select ElectricAccelerator VS IDE Add-in.

Setting Environment Variables for Visual Studio

You can control the way the add-in works by setting these environment variables on the Electric Make machine.

Note: Environment variables can be true or false. Valid boolean values are "0", "no", "false", "off" and "1", "yes", "true", "on". Case is not significant.

Environment Variable	Description	Usage
ECADDIN_DEBUG	Setting this variable to any value causes debug log files to remain in C:\ecdebug<ID>.log on the agent host. These files are used for troubleshooting by Electric Cloud engineers. Normally, you do not need to set this value.	debugging
ECADDIN_DEBUG_LOG_FILENAME	Specifies the debug log name. Requires ECADDIN_DEBUG. Use '$1' in the file specification to insert a unique ID. For example, C:\ecdebug_$1.log. Use a file location outside of emake root. The log file is stored on the agent.	debugging
ECADDIN_DONT_RM_TMP_MAKEFILES	Retains makefiles created during the build but normally deleted when the build finishes. This environment variable can have any value; it just needs to be set.	debugging
ECADDIN_DONT_USE_UNIQUE	Does not use unique names for temporary makefiles. Use with ECADDIN_DONT_RM_TMP_MAKEFILES.	debugging
ECADDIN_ECBREAKPOINT	Determines whether to invoke ecbreakpoint on failed jobs.	debugging
ECADDIN_ECBREAKPOINT_PROJECTS	Determines whether to invoke ecbreakpoint for specified projects. Use a semi-colon to delimit projects.	debugging
ECADDIN_ADD_IMPLICIT_PDB_DEPENDENCIES	Adds dependencies to improve first-time build speed.	performance
ECADDIN_MAX_PDB_FILES	Specifies the maximum number of PDB files produced (set to 16 by default).	performance
ECADDIN_REMOVE_DEPENDENCIES	Removes project dependencies (on by default).	performance
ECADDIN_USE_DEVENV_FOR_PROJECT	Uses devenv (instead of MSBuild) to build specific projects. Supply a list of projects (separated by a semicolon) to be built with devenv.	
ECADDIN_CONTINUE_ON_ERROR	Allows the build to continue after an error has occurred. Off by default for the Solution Support Add-in. On by default for the Visual Studio IDE Add-in.	switch
ECADDIN_CREATE_MISSING_DEPENDENCIES	Creates missing dependencies to avoid missing dependency warnings.	switch
ECADDIN_DISALLOW_BSC	Does not generate browse information files.	switch

Environment Variable	Description	Usage
ECADDIN_DISALLOW_PCH	Does not generate/use precompiled header files (implied by ECADDIN_MAX_PDB_FILES)	switch
ECADDIN_DISALLOW_PDB	Does not generate PDB files.	switch
ECADDIN_DISALLOW_SBR	Does not generate browse information files from sources.	switch
ECADDIN_DONT_ADD_PCH_LOCATION	Prevents the add-in from adding the location of the PCH file in all cases. This variable is relevant only if ECADDIN_MAX_PDB_FILES or ECADDIN_DISALLOW_PCH is switched on.	switch
ECADDIN_DONT_PARSE_PROJECT	This variable takes a list of project names separated by semi-colons and without white spaces. This variable is useful for deploying the add-in. If for any reason the add-in cannot build some of your projects, this variable allows you to work around the problem. When using this variable, you may experience an the warning MSB4098. You can ignore this warning because any project references are now converted into additional dependencies. MSBuild, however, does not provide a mechanism to turn off this warning.	switch
ECADDIN_DONT_PARSE_PROJECTS	This variable takes any non-blank value and its behavior is similar to ECADDIN_SERIALIZE. It calls devenv on each project (the add-in does not convert each project into individual compile/link steps).	switch
ECADDIN_DONT_USE	Disables the add-in. This environment variable can have any value, it just needs to be set. Also, you can disable the add-in on each host by using the Visual Studio Add-in Manager (on the Tools menu). **Note:** This is a “light-weight” uninstall program that disables one individual machine at a time.	switch
ECADDIN_DISABLE_MINIMAL_REBUILDS	Disables minimal rebuilds (off by default).	switch
ECADDIN_ENABLE_INCREMENTAL_LINK	Inserts a call to ectouch.exe.	switch
ECADDIN_EXPAND_LINKER_OBJECTS	Expand linker objects to one line per object. Prevents errors when link line length is exceeded.	switch

Environment Variable	Description	Usage
ECADDIN_FORCE_Z7	Enables /Z7 compiler options for all C++ files.	switch
ECADDIN_INCLUDE_CMAKELISTS	Excludes any source file with the name CMakeLists.txt (false is default). Set ECADDIN_INCLUDE_CMAKELISTS=true to execute the file.	switch
ECADDIN_MSBUILD_PARAMETERS	Add extra parameters to msbuild command line.	switch
ECADDIN_NORMALIZE_PATHS	Normalizes all paths in the makefile. The default value is false.	switch
ECADDIN_RUN_DEPLOYMENT_PROJECTS_LOCALLY	Runs deployment projects locally using #pragma runlocal.	switch
ECADDIN_RUN_LOCAL_LIB	A list of projects where the librarian tool should be run locally (using #pragma runlocal).	switch
ECADDIN_RUN_LOCAL_LINK	A list of projects where the linker should be run locally (using #pragma runlocal).	switch
ECADDIN_RUN_LOCAL_PROJECT	Use this variable if your build uses a local resource (for example, a resource only on the Electric Make host (for example, a database). You do not need to set this variable if your project build includes web deployment; this is handled by the add-in. The value of this variable is a list of project names separated by semi-colons. Each project name must be the unique Visual Studio identifier for the project (for example, solution1/project1.vcproj). Do not add quotation marks or white spaces.	switch
ECADDIN_SERIALIZE	Causes each project to be built serially. It inserts '#pragma allserial' into each makefile. This variable is equivalent to setting ECADDIN_DONT_PARSE_PROJECTS.	switch
ECADDIN_UP_TO_DATE_CHECK	Pre-parses the projects to determine whether there is anything to build. Prevents unnecessary rebuilding of static build steps.	switch
ECADDIN_USE_DEVENV	Use devenv for all unparsed projects (the default is msbuild).	switch
ECADDIN_USE_MSBUILD	Allows you to use MSBuild internally for projects that the add-in cannot parse (on by default).	switch
ECADDIN_USE_RELATIVE_PATHS	Use relative paths in the makefile to reduce line lengths.	switch

Troubleshooting and Getting Help

Contacting Technical Support:

Before you contact our technical support staff, please have the following information available.

- Your name, title, company name, phone number, fax number, and email address
- Operating system and version number
- Product name and release version
- Problem description

Hours: 8AM - 5PM PST (Monday-Friday, except Holidays)

Phone: 408-419-4300, Option #2

Email: support@electric-cloud.com