intel

How to Build EFI 1.1 Sample Code

Version 1.0

November 26, 2003

intel.

How to Build EFI 1.1 Sample Code

Files

File	Description
VCVAR32.BAT	A command script generated by VC setup, which would set all the variables needed by VC compiler for you. It is located in the **`Bin`** subdirectory under the VC installation directory. Every version of VC provides its own **`VCVAR32.BAT`**. Thus, if there were multiple versions of VC residing on your disk, you have to choose the correct version of it before you could use the intended VC compiler.
Build.cmd	There is one **`build.cmd`** under each build tip directory. Its intention is to set up the compiling environment for a specific tip. It could call **`VCVARS32.BAT`** for you, as long as the appropriate arguments were given.
Build-common.cmd	Set **`EFI_SOURCE`** automatically. It would be called by **`Build-common-ia32.cmd`** and **`Build-common-ia64.cmd`**.
Build-common-ia32.cmd	It does all the things common to all IA-32 build tips. It assumes Visual Studio was installed under **`%ProgramFiles%`** directory. If that is not the case, you have to edit this file manually.
Build-common-ia64.cmd	Currently it only calls **`Build-common.cmd`** and does nothing else.

Environment Variables

Variable	Description	Example
ProgramFiles	A variable defined by the system. It is the default location where you applications would be installed.	C:\Program Files
EFI_SOURCE	The root directory of EFI1.1 source tree. It would be set automatically by **`Build-common.cmd`**, and you don't have to set it manually.	C:\Projecs\EFI1.1
EFI_VCVER_DEFAULT	The default argument you want to provide to **`build.cmd`** script to indicate which version of VC you prefer. Possible values including **`AUTO`**, **`VC6`**, **`VC7`**, **`VC71`**.	VC7
MSVCDIR	Set by **`VCVARS32.BAT`**, and should be pointed to the folder where VC was installed.	C:\Program Files\Microsoft Visual Studio .Net 2003\VC7
EFI_VCVER	Set by **`Build.cmd`**, which is the numeric representation of you VC version.	7

Explanation of build.cmd Script

1. `Build.cmd` takes 0 to 1 arguments. We would use *`Arg1`* to identify the only argument.
2. *`Arg1`* is the VC version you prefer. The script would try to call `VCVARS32.BAT` if you passed `VC6`, `VC7`, or `VC71`, or try to detect your VC version if you passed `AUTO`. If *`Arg1`* is omitted, the script would use `%EFI_VCVER_DEFAULT%` in place of it. And then if `EFI_VCVER_DEFAULT` is not set, an error would be reported.
3. Examples:
   a. `Build.cmd auto`
      The script would assume `VCVAR32.BAT` had been run, and it would try to find out which version of VC you are currently using, and report an error if none of the VC version supported were found.
   b. `Build.cmd vc7`
      The script would invoke the `VCVAR32.BAT` corresponding to VC7, and report an error if `VCVAR32.BAT` not found.
4. Suggestions:
   a. You could set `EFI_VCVER_DEFAULT` to your favorite values. And every time when you want to build EFI1.1, what you have to do is just to type `Build.cmd`.
   b. One approach is to set `EFI_VCVER_DEFAULT` to `AUTO`, and create two shortcuts on your desktop, one to execute `VCVAR32.BAT` of version 6, and another of version 7. What you have to do is to type `Build.cmd`, and the script would switch to VC6 or VC7 accordingly.

Step by Step

1. If you were to pass `AUTO` to `Build.cmd`, invoke `VCVAR32.BAT`. Or, if you want to specify which version of VC to use, check `Build-common-ia32.cmd` and make sure that whatever paths hard coded in it did point to your various VC installation folders.
2. Invoke `Build.cmd` of a build tip you want to build, with whatever choice you made.
3. Change to the build tip folder and run `nmake`.

Troubleshooting

When an error occurred, the script would stop. And you should see several lines surrounded by three asterisk marks (e.g. `*** VCVERSEL ***`). You can pick up the closest line just above the error message, and then look it up in the following table for possible reasons.

Tag	Possible reasons
VCVERSEL	Illegal argument passed to *`Arg1`*.
BUILDCOMMON	Just report it to me. It is most likely that there is a bug in that script.
VCVARS32	The path to the `VCVARS32.BAT` file is not correct. You should check if the path hard coded in the script is correct.
VCVERCHK	Unsupported VC version.
CLCHK	`Cl.exe` of `find.exe` cannot be found. Check `%MSVCDIR%` did point to the path where VC resided.