

Forum:[Hack Forums](#)

Thread: [\[Open Source\] .bat to exe converter](#)

Robert Rouge - March 28, 2025, 12:46 pm

here is a program that uses C# to compile a batch script into an executable using visual basic.
had lots of trouble making, but finished now. almost gave up but now i give to you

```
Code using System;
using System.CodeDom.Compiler;
using System.IO;
using System.Text;
using Microsoft.VisualBasic;

namespace BatchToExeConverter
{
    class Program
    {
        private static string[] sourceTemplate = new string[]
        {
            "Imports System",
            "Module BtE",
            "Sub Main()",
            "If System.IO.File.Exists(System.Environment.GetEnvironmentVariable(\"TMP\") & \"\\cmd.bat\") Then",
            "    System.IO.File.Delete(System.Environment.GetEnvironmentVariable(\"TMP\") & \"\\cmd.bat\")",
            "End If",
            "System.IO.File.WriteAllText(System.Environment.GetEnvironmentVariable(\"TMP\") & \"\\cmd.bat\", {BATCH})",
            "System.Diagnostics.Process.Start(System.Environment.GetEnvironmentVariable(\"TMP\") & \"\\cmd.bat\")",
            "End Sub",
            "End Module"
        };

        static void Main(string[] args)
        {
            if (args.Length < 2)
            {
                Console.WriteLine("Usage: BatchToExeConverter <batch file path> <output executable path>");
                return;
            }

            string batchFilePath = args[0];
            string outputExePath = args[1];

            try
            {
```

```

        string batchContent = File.ReadAllText(batchFilePath);
        string sourceCode = GenerateSourceCode(batchContent);
        CompileToExecutable(sourceCode, outputExePath);
        Console.WriteLine($"Executable created successfully at {outputExePath}");
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error: {ex.Message}");
    }
}

private static string GenerateSourceCode(string batchContent)
{
    StringBuilder result = new StringBuilder();
    foreach (var line in sourceTemplate)
    {
        result.AppendLine(line.Replace("{BATCH}", EscapeBatchContent(batchContent)));
    }
    return result.ToString();
}

private static string EscapeBatchContent(string batchContent)
{
    StringBuilder result = new StringBuilder();
    string[] lines = batchContent.Split(new[] { Environment.NewLine }, StringSplitOptions.None);

    foreach (string line in lines)
    {
        string escapedLine = line.Replace("\"", "\\\"");
        result.Append($"\"{escapedLine}\" & Environment.NewLine & ");
    }

    if (result.Length > 0)
    {
        result.Length -= (" & Environment.NewLine & ".Length);
    }

    return result.ToString();
}

private static void CompileToExecutable(string sourceCode, string outputExePath)
{
    var provider = new VBCodeProvider();
    var compilerParams = new CompilerParameters
    {
        GenerateExecutable = true,
        OutputAssembly = outputExePath,
        CompilerOptions = "/target:winexe /optimize+",
        GenerateInMemory = false
    };

    CompilerResults results = provider.CompileAssemblyFromSource(compilerParams,
sourceCode);

```

```

        if (results.Errors.HasErrors)
        {
            StringBuilder errorMessages = new StringBuilder("Compilation errors:\n");
            foreach (CompilerError error in results.Errors)
            {
                errorMessages.AppendLine($"Error {error.ErrorNumber}: {error.ErrorText} at line
{error.Line}");
            }
            throw new Exception(errorMessages.ToString());
        }
    }
}
}
}

```

}Just compile and run in command line [bat2exepath] [bat path] [exe path]

thank you to helpful people here: <https://hackforums.net/showthread.php?tid=6294893>

paulteng686 - May 6, 2025, 1:55 am

I am looking for exe converter to js

Robert Rouge - May 7, 2025, 1:59 pm

(May 6, 2025 01:55 AM)paulteng686 Wrote: [\[CITE\]](#) I am looking for exe converter to js
that is not what this is

Phantom_NiNja - May 7, 2025, 7:00 pm

(Mar 28, 2025 12:46 PM)Robert Rouge Wrote: [\[CITE\]](#) here is a program that uses C# to compile a
batch script into an executable using visual basic.

had lots of trouble making, but finished now. almost gave up but now i give to you

```

Code using System;
using System.CodeDom.Compiler;
using System.IO;
using System.Text;
using Microsoft.VisualBasic;

```

```

namespace BatchToExeConverter
{
    class Program
    {
        private static string[] sourceTemplate = new string[]
        {
            "Imports System",
            "Module BtE",
            "Sub Main()",
            "If System.IO.File.Exists(System.Environment.GetEnvironmentVariable(\"TMP\") &
            \"\\cmd.bat\") Then",
            "System.IO.File.Delete(System.Environment.GetEnvironmentVariable(\"TMP\") &
            \"\\cmd.bat\")",
            "End If",
            "System.IO.File.WriteAllText(System.Environment.GetEnvironmentVariable(\"TMP\") &
            \"\\cmd.bat\", {BATCH})",
            "System.Diagnostics.Process.Start(System.Environment.GetEnvironmentVariable(\"TMP\")
            & \"\\cmd.bat\")",
            "End Sub",

```

```

        "End Module"
    };

    static void Main(string[] args)
    {
        if (args.Length < 2)
        {
            Console.WriteLine("Usage: BatchToExeConverter <batch file path> <output executable
path>");
            return;
        }

        string batchFilePath = args[0];
        string outputExePath = args[1];

        try
        {
            string batchContent = File.ReadAllText(batchFilePath);
            string sourceCode = GenerateSourceCode(batchContent);
            CompileToExecutable(sourceCode, outputExePath);
            Console.WriteLine($"Executable created successfully at {outputExePath}");
        }
        catch (Exception ex)
        {
            Console.WriteLine($"Error: {ex.Message}");
        }
    }

    private static string GenerateSourceCode(string batchContent)
    {
        StringBuilder result = new StringBuilder();
        foreach (var line in sourceTemplate)
        {
            result.AppendLine(line.Replace("{BATCH}", EscapeBatchContent(batchContent)));
        }
        return result.ToString();
    }

    private static string EscapeBatchContent(string batchContent)
    {
        StringBuilder result = new StringBuilder();
        string[] lines = batchContent.Split(new[] { Environment.NewLine }, StringSplitOptions.None);

        foreach (string line in lines)
        {
            string escapedLine = line.Replace("\"", "\\\"");
            result.Append($"\"{escapedLine}\" & Environment.NewLine & ");
        }

        if (result.Length > 0)
        {
            result.Length -= (" & Environment.NewLine & ".Length);
        }
    }

```

```

        return result.ToString();
    }

    private static void CompileToExecutable(string sourceCode, string outputExePath)
    {
        var provider = new VBCodeProvider();
        var compilerParams = new CompilerParameters
        {
            GenerateExecutable = true,
            OutputAssembly = outputExePath,
            CompilerOptions = "/target:winexe /optimize+",
            GenerateInMemory = false
        };

        CompilerResults results = provider.CompileAssemblyFromSource(compilerParams,
sourceCode);

        if (results.Errors.HasErrors)
        {
            StringBuilder errorMessages = new StringBuilder("Compilation errors:\n");
            foreach (CompilerError error in results.Errors)
            {
                errorMessages.AppendLine($"Error {error.ErrorNumber}: {error.ErrorText} at line
{error.Line}");
            }
            throw new Exception(errorMessages.ToString());
        }
    }
}

```

}Just compile and run in command line [bat2exepath] [bat path] [exe path]

thank you to helpful people here: <https://hackforums.net/showthread.php?tid=6294893>

Is this correct from what I read it looks like in a nut shell.

It looks like you made a program that grabs a batch file opens it up and reads it line by line

then creates a c# code that runs batch with system functions stores that files and then uses a compiler with the optimization option and compiles the c# file to generate the .exe file?

This is correct in a simplified way. Meaning don't have to go into details but in a general sense.

Anonymuk - September 24, 2026, 9:58 am

Made one of these before.... Interesting to see other coder's take on how to make this.