Installation Guide

Adjust in the file dotNETRunner_Register.reg the path of the CodeBase entries to the path where dotNETRunner_x64.dll and dotNETRunner_x86.dll are stored. After that it is necessary to execute the file dotNETRunner_Register.reg, this writes the necessary entries in the Windows registry. All entries are stored in HKCU hive of the current user.
With the execution of the file dotNETRunner_Unregister.reg it is possible to delete all the written registry entries of dotNETRunner.

Prerequisite

- Installed dotNET Framework version >= 4.5
- Installed Windows PowerShell version >= 5.0

User Guide

dotNETRunner

This class offers the possibility to execute C#, VB.NET, Jscript or PowerShell code.

Output

It is possible for the …String methods to write the result into an internal buffer. On this way you can collect different results and get them up at once, with the property OutputString. To use the internal buffer set OutputMode to 1 (OutputBuffer). The default of OutputMode is 0 (OutputDirect), which means that you get a result with each call of a …String method.

- Property *OutputMode*: Type int, sets or gets the current output mode.
  Possible values are OutputDirect = 0 or OutputBuffer = 1, default is 0.
  Hint: If you set the OutputMode to OutputBuffer (1) then all results of the …String methods are written into an internal buffer.

- Property *OutputString*: Type string, gets the content of the output buffer.

- Method *ClearBuffer*: Type void, clears the internal output buffer where output is added.

C#, VB.NET or JScript

- Property AddAssembly: Type string, name of an assembly to add,
  e.g. System.Windows.Forms.dll.

- Method *ClearAssemblies*, Type void, clears added assemblies.

- Method *CompileCode*: Type string, compiles the source code.
  Hint: It is necessary to call this or CompFile method before you call a run method.
  - Language: Type string, CS for C#, VB for VB.NET or JS for JScript are allowed.
  - Code: Type string, the C#, VB.NET or JScript code.
  - IncludeDebugInformation: Type bool, generate debug information, optional, default is true.
  - WarningLevel: Type int, level at which the compiler starts displaying warnings, optional, default is 3.

- Method *CompileFile*: Type string, compiles the source code in the file.
  Hint: It is necessary to call this or CompCode method before you call a run method.
  - Language: Type string, CS for C#, VB for VB.NET or JS for JScript are allowed.
  - FileName: Type string, the file name which contains the C#, VB.NET or JScript code.
  - IncludeDebugInformation: Type bool, generate debug information, optional, default is true.

  - WarningLevel: Type int, level at which the compiler starts displaying warnings, optional, default is 3.

- Method *Run*: Type object, executes a method with parameters.
  - Instance: Type string, name of the instance to create.
  - ConstructorParameters: Type string, optional, default is an empty string.
  - Method: Type string, optional, name of the method of the instance to execute.
  - MethodParameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.

- Method *RunString*: Type string, executes a method with parameters.
  It is a wrapper for the method run but the result is converted to string.
  Hint: This is the method for the ABAP call.
  - Instance: Type string, name of the instance to create.
  - ConstructorParameters: Type string, optional, default is an empty string.
  - Method: Type string, optional, name of the method of the instance to execute.
  - MethodParameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.

PowerShell

- Method *IsPowerShellInstalled*: Type bool, checks if PowerShell version >= 5 is installed and delivers true, otherwise false.

- Property *PowerShellVersion*: Type string, delivers the PowerShell version.
  Hint: It is necessary to call method IsPowerShellInstalled first.

- Method *RunPSCode*: Type Collection of PSObject, executes PowerShell code.
  - Code: Type string, the PowerShell code.
  - Parameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.
  - STA: Type Boolean, run code in single-threaded apartment mode, optional, default is true.

- Method *RunPSFile*: Type Collection of PSObject, executes PowerShell file.
  It is a wrapper for the method RunPSCode.
  - FileName: Type string, the name of the file which contains the PowerShell code.
    Parameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.
  - STA: Type Boolean, run code in single-threaded apartment mode, optional, default is true.

- Method *RunPSCodeString*: Type string, executes PowerShell code.
  It is a wrapper for the method RunPSCode but the result is converted to string.
  - Code: Type string, the PowerShell code.
  - Parameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.
  - STA: Type Boolean, run code in single-threaded apartment mode, optional, default is true.

- Method *RunPSFileString*: Type string, executes PowerShell file.
  It is a wrapper for the method RunPSCodeString.
  - FileName: Type string, the name of the file which contains the PowerShell code.
  - Parameters: Type string, optional, default is an empty string.
  - Separator: Type string, optional, default is a comma.
  - STA: Type Boolean, run code in single-threaded apartment mode, optional, default is true.