

# HTML to RTF .Net

*(Multi-platform .Net library)*

## macOS development manual

### Table of Contents

|                                  |   |
|----------------------------------|---|
| 1. Preparing environment .....   | 2 |
| 2. Dependencies and samples..... | 6 |

[SautinSoft](http://SautinSoft)

[support@sautinsoft.com](mailto:support@sautinsoft.com)

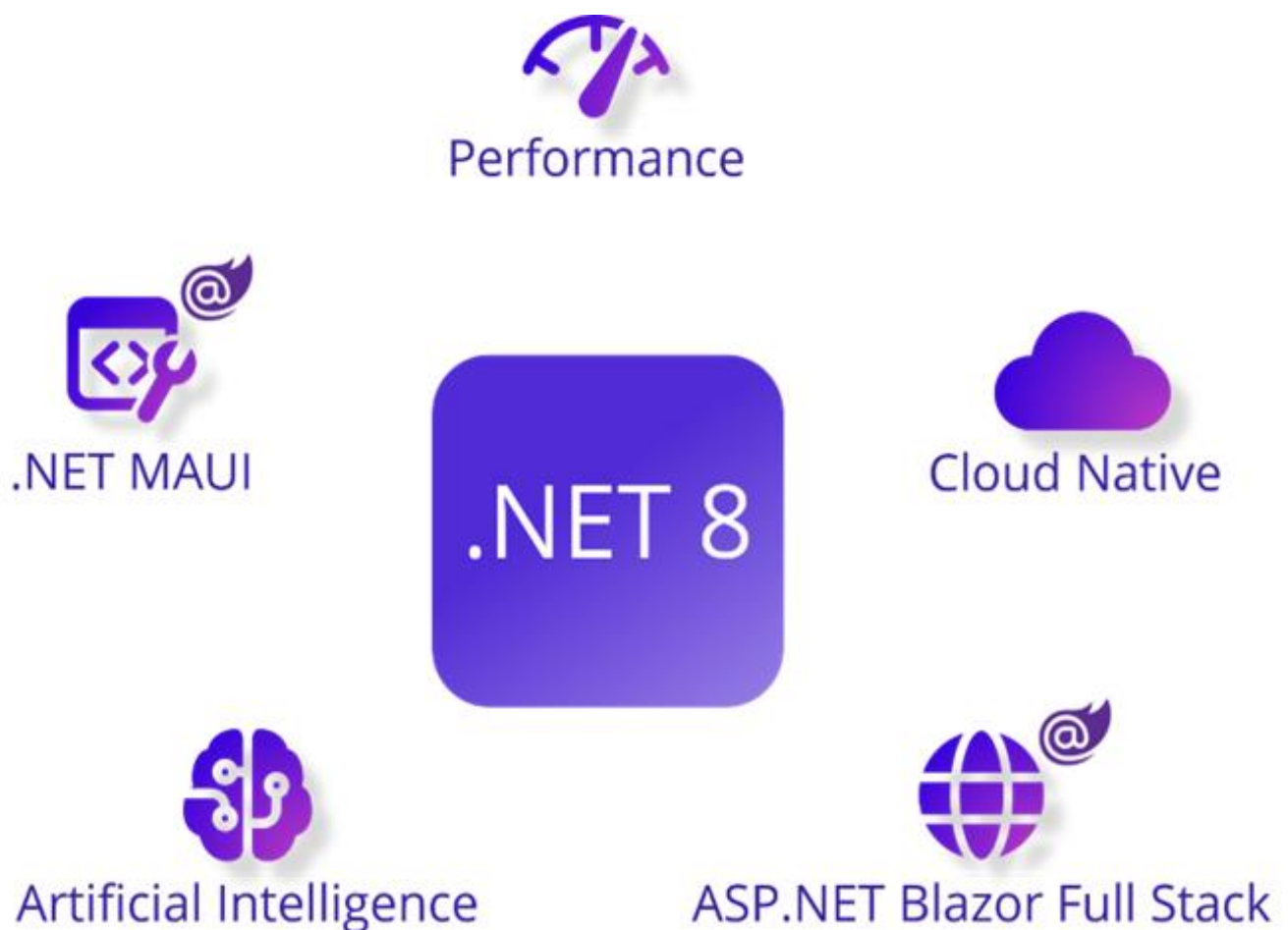
# 1. Preparing environment

In order to build multi-platform applications using .NET on macOS, the first steps are for installing in our MAC machine the required tools.

**We need to install .NET Core SDK from Microsoft and to allow us to develop easier, we will install an advance editor with a lot of features, Visual Studio Code from Microsoft.**

Both installations are very easy and the detailed description can be found by these two links:

- [Install .NET Core SDK for macOS.](#)



- [Install VS Code for macOS.](#)

The image shows the Visual Studio Code website on the left and a screenshot of the VS Code interface on the right. The website has a dark blue header with the Visual Studio Code logo and navigation links: Docs, Updates, Blog, API, Extensions, and FAQ. Below the header, a banner reads 'Code editing. Redefined.' with the tagline 'Free. Built on open source. Runs everywhere.' A green button labeled 'Download for Windows' is visible, with a dropdown menu showing download links for macOS, Windows x64, and Linux x64. The screenshot on the right shows the VS Code interface with the 'EXTENSIONS' view open, displaying a list of popular extensions like C#, Python, and ESLint.

Visual Studio Code

Docs Updates Blog API Extensions FAQ

Version 1.37 is now available! Read about the new features

# Code editing. Redefined.

Free. Built on open source. Runs everywhere.

Download for Windows  
Stable Build

|             |                | Stable | Insiders |
|-------------|----------------|--------|----------|
| macOS       | Package        | ↓      | ↓        |
| Windows x64 | User Installer | ↓      | ↓        |
| Linux x64   | .deb           | ↓      | ↓        |
|             | .rpm           | ↓      | ↓        |

[Other downloads](#)

File Edit View Goto Help

EXTENSIONS

@popular

- C# 1.2.2 356K ★★★★★  
C# for Visual Studio Code (p...  
Microsoft [Install](#)
- Python 0... 211K ★★★★★  
Linting, Debugging (multi-t...  
Don Jayamanne [Install](#)
- Debugger for Chrome 148  
Debug your JavaScript code...  
Microsoft JS Diagno... [Install](#)
- C/C++ 0.7... 143K ★★★★★  
Complete C/C++ language ...  
Microsoft [Install](#)
- Go 0.6.39 99K ★★★★★  
Rich Go language support f...  
lukehoban [Install](#)
- ESLint 0.10... 88K ★★★★★  
Integrates ESLint into VS Co...  
Dirk Baumer [Install](#)

master 11 131 0 0 0

Once installed VS Code, you need to install a C# extension to facilitate us to code and debugging:

Install [C# extension](#).

At the time of this writing, .NET Core has a problem with supporting the GdiPlus library on macOS. There is a solution that will help you to run any .NET libraries using GdiPlus.

## Problem:

You got the whole thing up and running in debug, but when you went for your dotnet run, you got the following crash:

```
The type initializer for 'System.Drawing.GDIPlus' threw an exception. -> System.DllNotFoundException: Unable to load DLL 'gdiplus': The specified module or one of its dependencies could not be found.
```

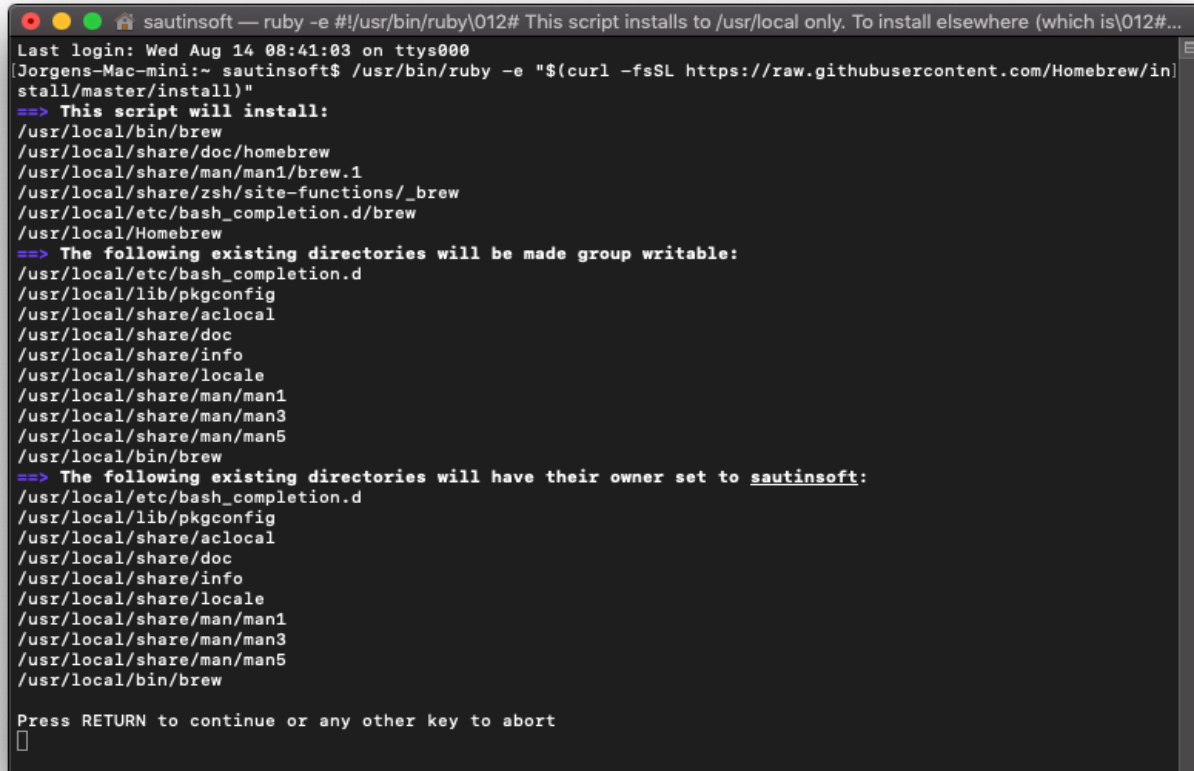
## Solution:

First of all, you need to install "[Homebrew](#)" - The missing package manager for macOS.

*Paste that in a macOS Terminal prompt:*

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

The script explains what it will do and then pauses before it does it.



```
sautinsoft — ruby -e #!/usr/bin/ruby\012# This script installs to /usr/local only. To install elsewhere (which is\012#...
Last login: Wed Aug 14 08:41:03 on ttys000
Jorgens-Mac-mini:~ sautinsoft$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/in]
stall/master/install)"
==> This script will install:
/usr/local/bin/brew
/usr/local/share/doc/homebrew
/usr/local/share/man/man1/brew.1
/usr/local/share/zsh/site-functions/_brew
/usr/local/etc/bash_completion.d/brew
/usr/local/Homebrew
==> The following existing directories will be made group writable:
/usr/local/etc/bash_completion.d
/usr/local/lib/pkgconfig
/usr/local/share/aclocal
/usr/local/share/doc
/usr/local/share/info
/usr/local/share/locale
/usr/local/share/man/man1
/usr/local/share/man/man3
/usr/local/share/man/man5
/usr/local/bin/brew
==> The following existing directories will have their owner set to sautinsoft:
/usr/local/etc/bash_completion.d
/usr/local/lib/pkgconfig
/usr/local/share/aclocal
/usr/local/share/doc
/usr/local/share/info
/usr/local/share/locale
/usr/local/share/man/man1
/usr/local/share/man/man3
/usr/local/share/man/man5
/usr/local/bin/brew

Press RETURN to continue or any other key to abort
^
```

[SautinSoft](#)

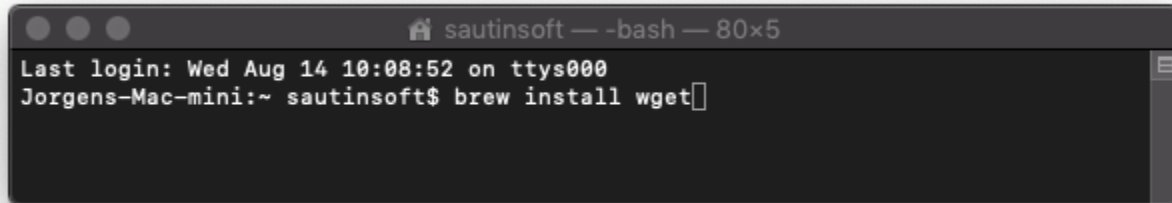
[support@sautinsoft.com](mailto:support@sautinsoft.com)

Homebrew installs [the stuff you need](#) that Apple (or your Linux system) didn't.

*Paste that in a macOS Terminal prompt:*

```
$ brew install wget
```

Homebrew installs packages to their own directory and then symlinks their files into `/usr/local`

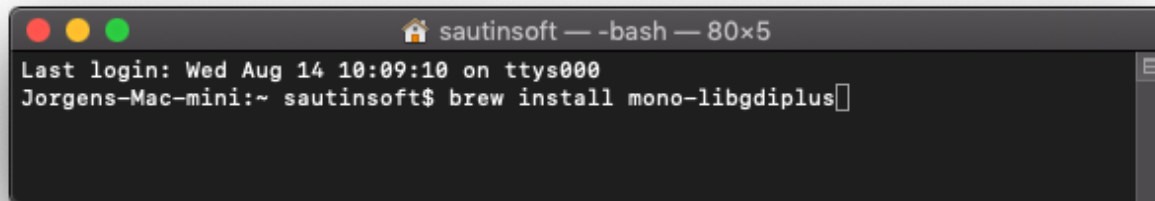
A screenshot of a macOS Terminal window. The title bar shows a home icon, the name 'sautinsoft', and the prompt '-bash — 80x5'. The terminal text shows 'Last login: Wed Aug 14 10:08:52 on ttys000' and the command 'Jorgens-Mac-mini:~ sautinsoft\$ brew install wget' being entered at the prompt.

Homebrew won't install files outside its prefix and you can place a Homebrew installation wherever you like. Homebrew complements macOS (or your Linux system). Install your RubyGems with `gem` and their dependencies with `brew`.

Now, we need to install [mono-libgdiplus](#). GdiPlus – compatible API on non-Windows operating systems.

*Paste that in a macOS Terminal prompt:*

```
$ brew install mono-libgdiplus
```

A screenshot of a macOS Terminal window. The title bar shows a home icon, the name 'sautinsoft', and the prompt '-bash — 80x5'. The terminal text shows 'Last login: Wed Aug 14 10:09:10 on ttys000' and the command 'Jorgens-Mac-mini:~ sautinsoft\$ brew install mono-libgdiplus' being entered at the prompt.

Congratulations, you have installed all the dependencies needed to run .NET components.

## 2. Dependencies and samples

Create a simple project by Visual Code or an another Environment for C# or VB.NET. To get started with our product, you need to include the following NuGet packages in your project file (and don't forget to add our library to your project's references):

```
<PackageReference Include="System.IO.Packaging" Version="4.5.0" />
<PackageReference Include="System.Text.Encoding.CodePages" Version="4.5.0" />
<PackageReference Include="SkiaSharp" Version="2.88.7" />
<PackageReference Include="SkiaSharp.NativeAssets.macOS" Version="2.88.7" />
<PackageReference Include="Svg.Skia" Version="1.0.0.18" />
```

Great! Now you can fully harness our product and bring all your ideas to life.

Need inspiration or guidance? Check out a lot of code samples on our website:

[www.sautinsoft.com](http://www.sautinsoft.com) or GitHub (SautinSoft).

[SautinSoft](http://www.sautinsoft.com)

[support@sautinsoft.com](mailto:support@sautinsoft.com)