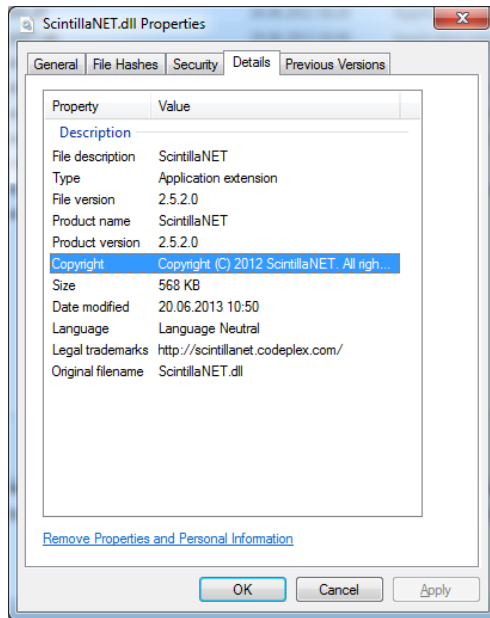# Scintilla.NET can't be loaded in Terminals

Introduction

Bug in Scintilla.NET version 2.5.2 Scintilla.cs line 669:


```
private static void LoadModule()
{
    if (_moduleHandle == IntPtr.Zero)
    {
        // Load the Scintilla module into memory
        if ((_moduleHandle = NativeMethods.LoadLibrary(_moduleName)) == IntPtr.Zero)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_CannotLoadModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }

        // Get the direct function. We use GetProcAddress instead of DllImport
        // because we don't know the name of the module ahead of time.
        _directFunction = Marshal.GetDelegateForFunctionPointer(
            NativeMethods.GetProcAddress(_moduleHandle, "Scintilla_DirectFunction"),
            typeof(NativeMethods.Scintilla_DirectFunction)) as NativeMethods.Scintilla_DirectFunction;

        if (_directFunction == null)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_InvalidModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }
    }
}
```

Symptoms

Terminals plugin is unable to load the scintilla native component and produces an error in the Terminals log like:

```
ERROR 2013-06-20 15:32:50,561 51427ms Terminals                 Error
- System.ComponentModel.Win32Exception: Cannot load the 'SciLexer.dll'
module into memory. ---> System.ComponentModel.Win32Exception: The
specified module could not be found
   --- End of inner exception stack trace ---
   at ScintillaNET.Scintilla.LoadModule() in
C:\Users\jslusser\Projects\ScintillaNET\Releases\2.5\ScintillaNET\Scintilla
.cs:line 669
   at ScintillaNET.Scintilla.get_CreateParams() in
C:\Users\jslusser\Projects\ScintillaNET\Releases\2.5\ScintillaNET\Scintilla
.cs:line 2508
   at System.Windows.Forms.Control..ctor(Boolean autoInstallSyncContext)
   at System.Windows.Forms.Control..ctor()
   at ScintillaNET.Scintilla..ctor() in
C:\Users\jslusser\Projects\ScintillaNET\Releases\2.5\ScintillaNET\Scintilla
.cs:line 3688
   at CodeEditor.Edit.InitializeComponent() in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.Designer.cs:line 123
   at CodeEditor.Edit..ctor() in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.cs:line 696
ERROR 2013-06-20 15:32:50,570 51436ms Terminals                 Error
- System.NullReferenceException: Object reference not set to an instance of
an object.
   at CodeEditor.Edit.SetScintillaToCurrentOptions(Scintilla scintilla) in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.cs:line 563
   at CodeEditor.Edit.NewDocument() in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.cs:line 324
   at CodeEditor.Edit..ctor() in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.cs:line 705
ERROR 2013-06-20 15:32:52,437 53303ms Terminals                 Error
- System.NullReferenceException: Object reference not set to an instance of
an object.
   at CodeEditor.Edit.SetLanguage(String extension) in
d:\Kohl\Projects\Code\Terminals\CodeEditor\Edit.cs:line 77
```

Solution

If the build in solution Terminals uses doesn't work please be so kind to report the error and use one of the below mentioned solutions (solution one to three) - every solution except solution number four.

Solution 1:

Change the line 669 in Scintilla.cs:

from

```
private static void LoadModule()
{
    if (_moduleHandle == IntPtr.Zero)
    {
        // Load the Scintilla module into memory
        if ((_moduleHandle = NativeMethods.LoadLibrary(_moduleName)) == IntPtr.Zero)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_CannotLoadModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }

        // Get the direct function. We use GetProcAddress instead of DllImport
        // because we don't know the name of the module ahead of time.
        _directFunction = Marshal.GetDelegateForFunctionPointer(
            NativeMethods.GetProcAddress(_moduleHandle, "Scintilla_DirectFunction"),
            typeof(NativeMethods.Scintilla_DirectFunction)) as NativeMethods.Scintilla_DirectFunction;

        if (_directFunction == null)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_InvalidModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }
    }
}
```

to

```
private static void LoadModule()
{
    if (_moduleHandle == IntPtr.Zero)
    {
        // Load the Scintilla module into memory
        if ((_moduleHandle = NativeMethods.LoadLibrary(System.IO.Path.Combine(Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location), _moduleName))) == IntPtr.Zero)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_CannotLoadModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }

        // Get the direct function. We use GetProcAddress instead of DllImport
        // because we don't know the name of the module ahead of time.
        _directFunction = Marshal.GetDelegateForFunctionPointer(
            NativeMethods.GetProcAddress(_moduleHandle, "Scintilla_DirectFunction"),
            typeof(NativeMethods.Scintilla_DirectFunction)) as NativeMethods.Scintilla_DirectFunction;

        if (_directFunction == null)
        {
            string message = string.Format(Resources.Culture, Resources.Exception_InvalidModule, _moduleName);
            throw new Win32Exception(message, new Win32Exception(Marshal.GetLastWin32Error()));
        }
    }
}
```
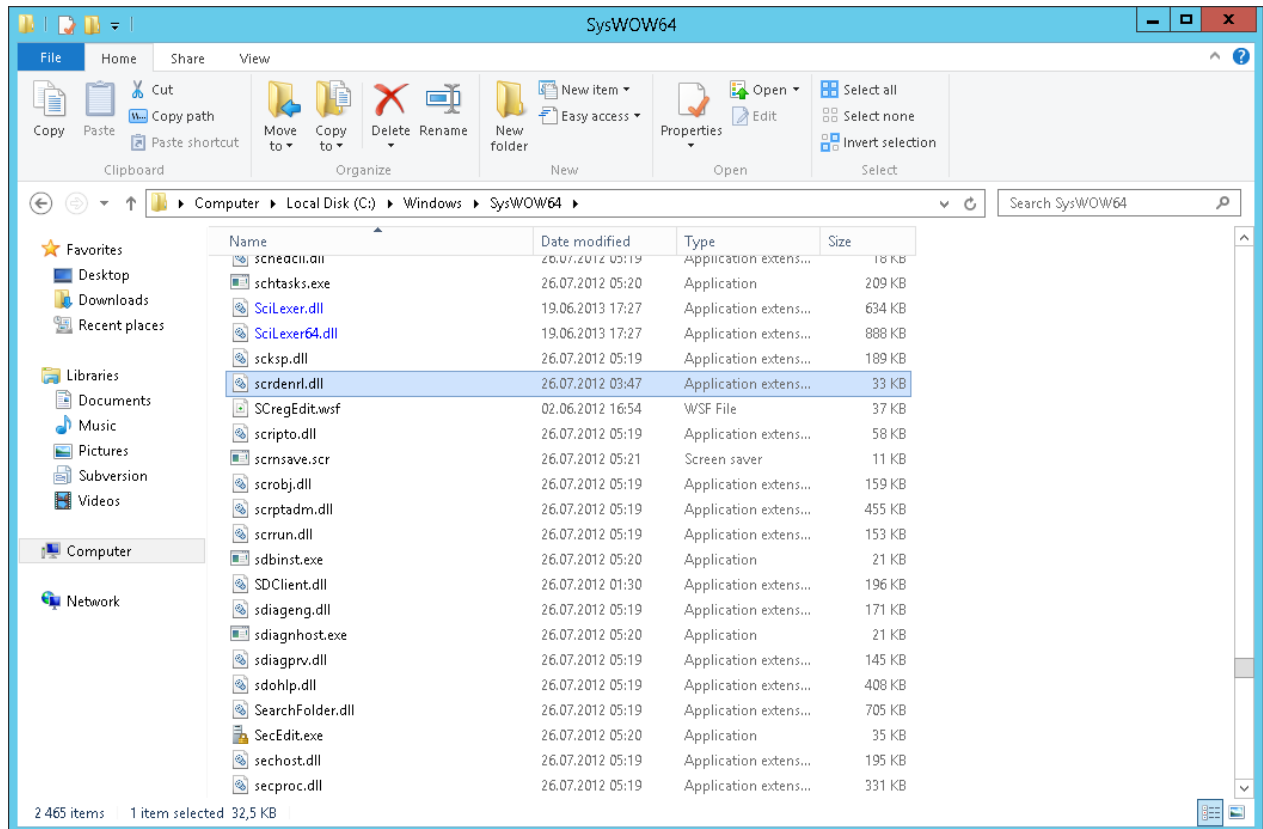
Solution 2:

Move or copy the Scintilla.NET dependent libraries to the Terminals root directory:

Solution 3:

Move or copy the Scintilla.NET dependent libraries to the System32 and SysWow64 directories.

📁 System32	20.06.2013 11:32	File folder
📁 SysWOW64	20.06.2013 10:44	File folder

Solution 4 (default, built-in)

Terminals try to work around this problem by using the following code in the "Terminals code editor"

→ Workaround: Load SciLexer or SciLexer64 before the Scintilla.NET library does.

```csharp
[DllImport("kernel32.dll")]
private static extern IntPtr LoadLibrary(string dllToLoad);

static Edit()
{
    string nativeDllName = IntPtr.Size == 4 ? "SciLexer.dll" : "SciLexer64.dll";
    string path = Path.GetDirectoryName(System.Reflection.Assembly.GetExecutingAssembly().Location);
    string name = System.IO.Path.Combine(path, nativeDllName);

    Log.Debug("Loading native dll '" + name + "' dynamically.");

    if (LoadLibrary(name) == IntPtr.Zero)
    {
        Log.Fatal("Error loading native dll " + nativeDllName + " at '" + path + "'.");
    }
}
```

Notifying the developer

Either navigate to the website

http://www.kohl.bz

or

write an e-mail to

oliver@kohl.bz

and report the issue, possible root causes and symptoms.