

ScintillaNet SandBox



<http://www.keywild.com/SQLite>

Public Domain Software

Author: Lewis Balentine

"Engineering is the art of planning and forethought."



Legal Disclaimer:

The author makes no representation or warranty, either expressed or implied, with respect to the data files and/or software, their quality, accuracy, or fitness for any specific application. Therefore the author shall have no liability to any person or any entity with respect to any liability, loss, or damage caused or alleged to have been caused directly or indirectly by the use of the data files and/or software. This includes, but is not limited to, interruption of service, loss of data, loss of consulting or anticipatory profits or consequential damages from the use of these data files and/or software.

All files unless otherwise noted are the original work product of the author. Unless otherwise noted these files are placed into the Public Domain for the unrestricted use by anyone for any purpose. Placing these files in the public domain shall in no way be construed as an obligation of the author (*or his heirs and/or assigns*) to maintain the web site, web pages, files, data or software. Further it shall in no way limit the author's (*or his heirs and/or assigns*) options to make, produce or use versions of the software, data files, CAD objects or other material posted under the URL in any other commercial or non-commercial venture.

In the event of a legal dispute the court is requested to use a "reasonable person's" interpretation of the "clear intent" of this disclaimer.

The use of these data files and/or software constitutes acceptance of this disclaimer.

"No good deed goes unpunished." — Oscar Wilde

The screenshot shows the ScintillaNET SandBox application window. The title bar reads "ScintillaNET SandBox" and the path is "\\keywild.lan\Public\TempWorkShared\ScintillaNET\". The interface includes a menu bar, a toolbar with buttons for "Zoom In", "Zoom Out", "Reset Fonts", "Next Style", "Show Styles", "All Styles to XML File", and "All Lang to XML File". The main area is a code editor with the following content:

```
1 1234567890 {} [] {} <> / \ + - * = & @ # $ % ^ ! ? # ~ |
2 ABCDEF GHIJKLM "Double quoted string" 'Single Quoted String'
3
4 Select ROWID, Bogus, count(child) as Children, * from [Employees] where [Name] like '%DOE%'
   group by state limit 0,100;
5
6 CREATE TABLE NewTable (x INTEGER, y, z, PRIMARY KEY(x ASC));
7
8 int cFunction ( int x, int y) { return x * y;}
9
10 PRAGMA ignore_check_constraints; -- SQL comment
11
12 /* This is a comment for some SQL system as well as the C language */
13 <!-- just to be fair we have a HTML comment in as well -->
14 ; Ini files and some others use semicolons for comment lines
15 # some language use a hash sign for comment lines
16 /* Freebasic supports block comments */
17 Basic_Line_Label: ' Most basic use single quote for comments.
```

Contents:

ScintillaNet SandBox.....	2
Command Line Options	2
Program Requirements	3
SandBox Operation	3
Creating a ScintillaNet XML Language file	8
ScintillaNet XML Structure	8
SQLite.XML	10
WEB Links :	11
Shameless Self Promotion:.....	11

ScintillaNet SandBox

Scintilla is a syntax highlighting source code editing control available for Windows, Linux and OSX operating systems that supports over 80 programming languages. ScintillaNet is a Net Frameworks wrapper for the Scintilla.dll. I wanted to use this software with a SQLite project. ScintillaNet implemented a custom XML based configuration to allow the control to be used with languages that do not have support built into the basic Scintilla.dll. Unfortunately there is sparse documentation on the creation and use of these XML files. ScintillaNet Sandbox is a PUBLIC DOMAIN Visual Studio 2010 Visual Basic project to extract some understanding of these XML files and how to create new ones. This project targets Net Frameworks X86 4.0.

The Scintilla DLL has 74 language Lexers and up to 255 styles that are used to format the text in the control. The Lexer determines what styles are available and how they are used for a particular programming language. Unless one decides to write a new Lexer from scratch then one must use one of the built in Lexers for their new language XML file. Each XML file may have up to seven Keyword lists associated with the language. One of the big problems in creating a new ScintillaNet XML file is determining what Lexer to use and finding out what styles are available with the Lexer and more specifically what they are used for. This application was created to address that problem.

Command Line Options

There are four command line options for Sandbox:

-Work: This specifies the working directory. The default is **C:\ScintillaNet** but if no files are found in that directory it reverts to the current working directory (*usually the application directory*). That directory may be either a local directory or a network directory (*as shown in the screenshot above*). This can be over ridden with this command line option. The working directory is used to write new XML files to or load XML from. If a subdirectory named "XML" exists in the working directory it will be used. Example:

-work:"c:\somplace Else\working directory" (note the quotation marks)

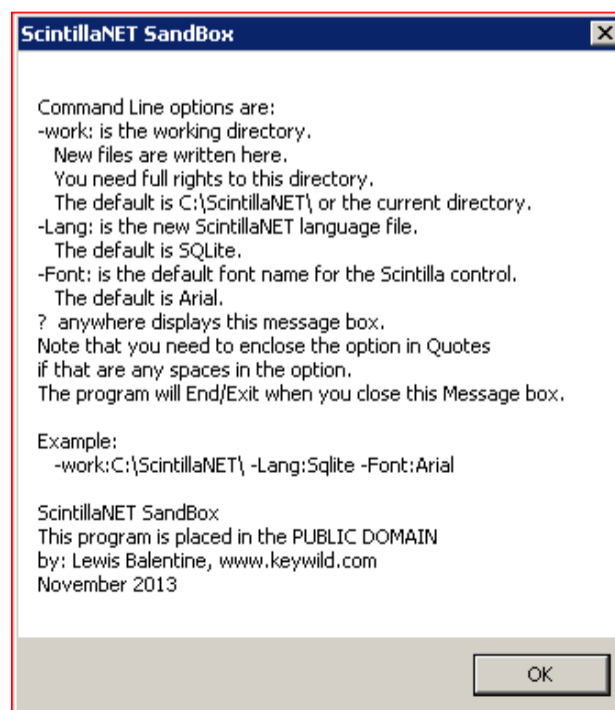
-Lang: This is an option to specify a new language file that is being created. This adds the new language to the END of list of languages supported by ScintillaNet. The default is SQLite. This Example:

-lang:NewLanguage

-Font: This sets the Font used by the "Reset Fonts" button. The default is Arial. This Example:

-lang:otherfont

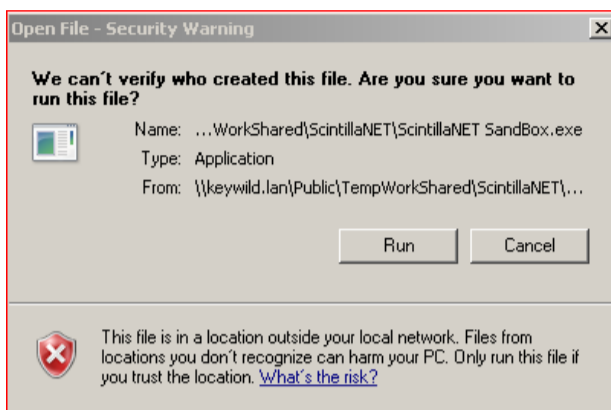
? A question mark anywhere in the command line brings up a message box with a short explanation of these options. The program then exits.



Program Requirements

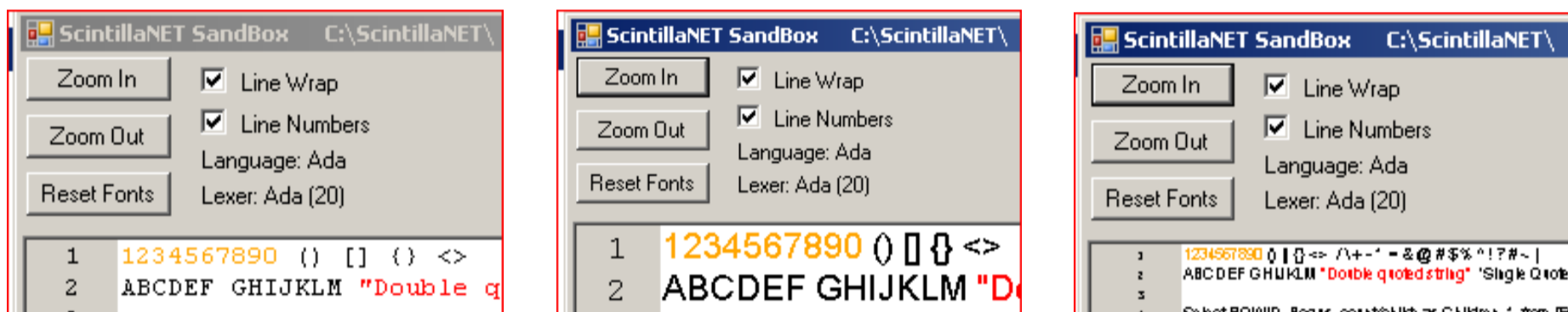
This application requires ScintillaNET.dll and SciLexer.dll. There is also a X64 version of the SciLexer64.dll but this application neither requires it or will take advantage of it. The program expects to find these dll in the same directory as the application but should work with the dll in the %windows%\system32 directory as well (*that configuration has NOT been tested*). The optional XLM subdirectory can be used for the XML configuration files that come with ScintillaNet. The application also requires Microsoft Net Frameworks 4.0.

The program is NOT signed and will generate a security message if run from a network directory:



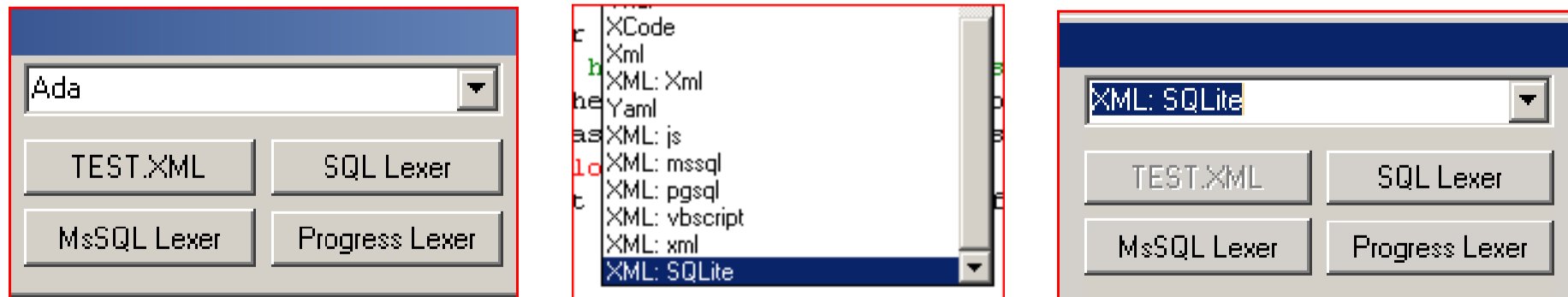
SandBox Operation

When the application is loaded it comes up with a Scintilla Text Box loaded with an assortment of strings one might find in various programming languages. If you do not like the text used then you can create a file named "SandBox-TestText.xml" with something more to your liking. This file should be placed in the working directory. On the extreme left are three buttons to control the text font used in the Editor box:



Personally I abhor Microsoft's "Courier New" font (*the only mono-space font provided by Microsoft*) and that is the font that ScintillaNet uses by default. The "Reset Fonts" will change the fonts to Arial by default or whatever option that is specified on the command line. The "Zoom In" and "Zoom Out" button can be used to control the size of the text in the Editor control. To the right of these buttons are two check boxes used to turn "Line Wrapping" and "Line Numbers" on or off. Below the checkboxes are two labels that indicate the current Language and Lexer are in use. ScintillaNet uses names for the Lexer while ScintillaNet.DLL uses a integer number. A single ScintillaNet.DLL Lexer number may be used with several ScintillaNet languages.

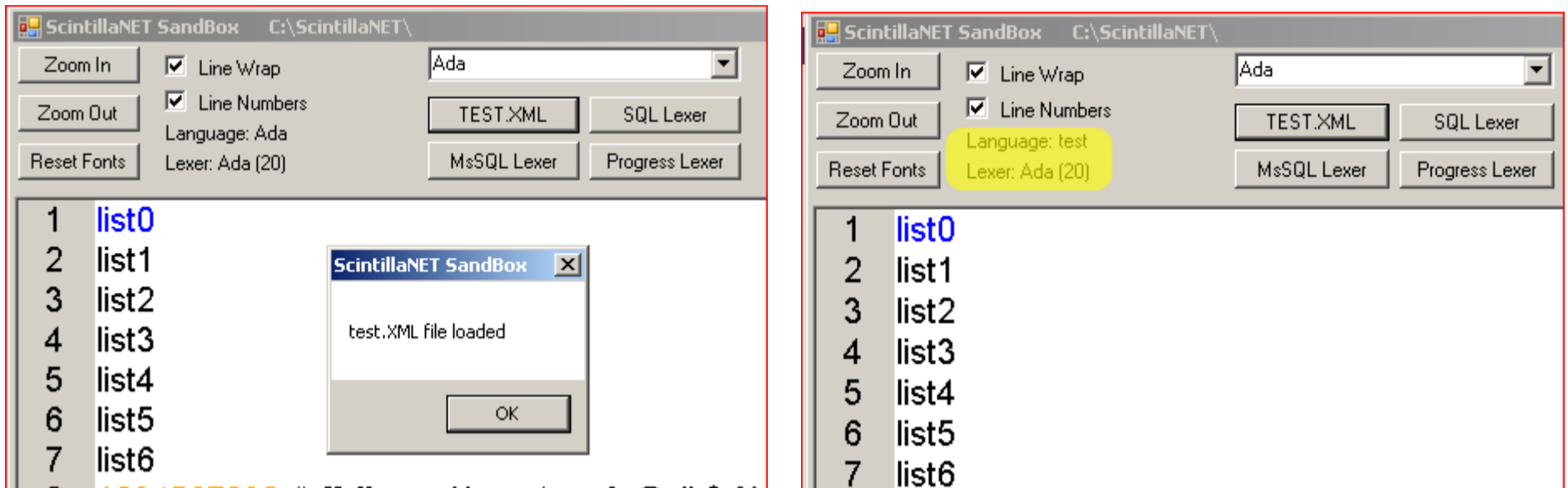
Next in line is the language selection Combo box.



The drop down lists has all the native languages supported by the Scintilla.DLL and the XML defined languages include with ScintillaNet if the XML files were located at startup. Each of the XML languages has the characters "XML:" appended to the front of the language name. Thus you may have two different configurations for each

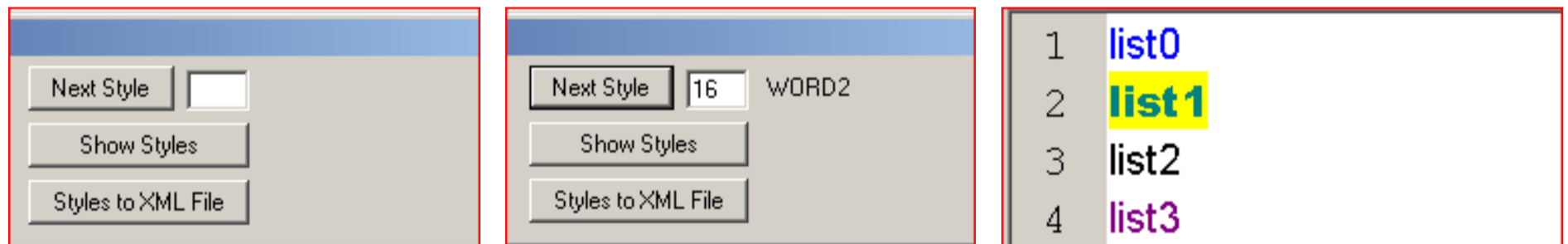
language (actually three is you have XML files in both the working directory and the XML sub directory). The last Language listed is the current XML development that is specified via the command line option.

Under the Language control are four buttons. The first of these is labeled “TEST.XML”. This intended to be used to determine the style used by a particular Lexer. This button is NOT enabled when any XML based language is selected. It constructs a XML file for the language “test” using the Lexer for the language currently indicated by the language control. Then it loads that XML file. It also inserts seven text lines in the top of the editor box:



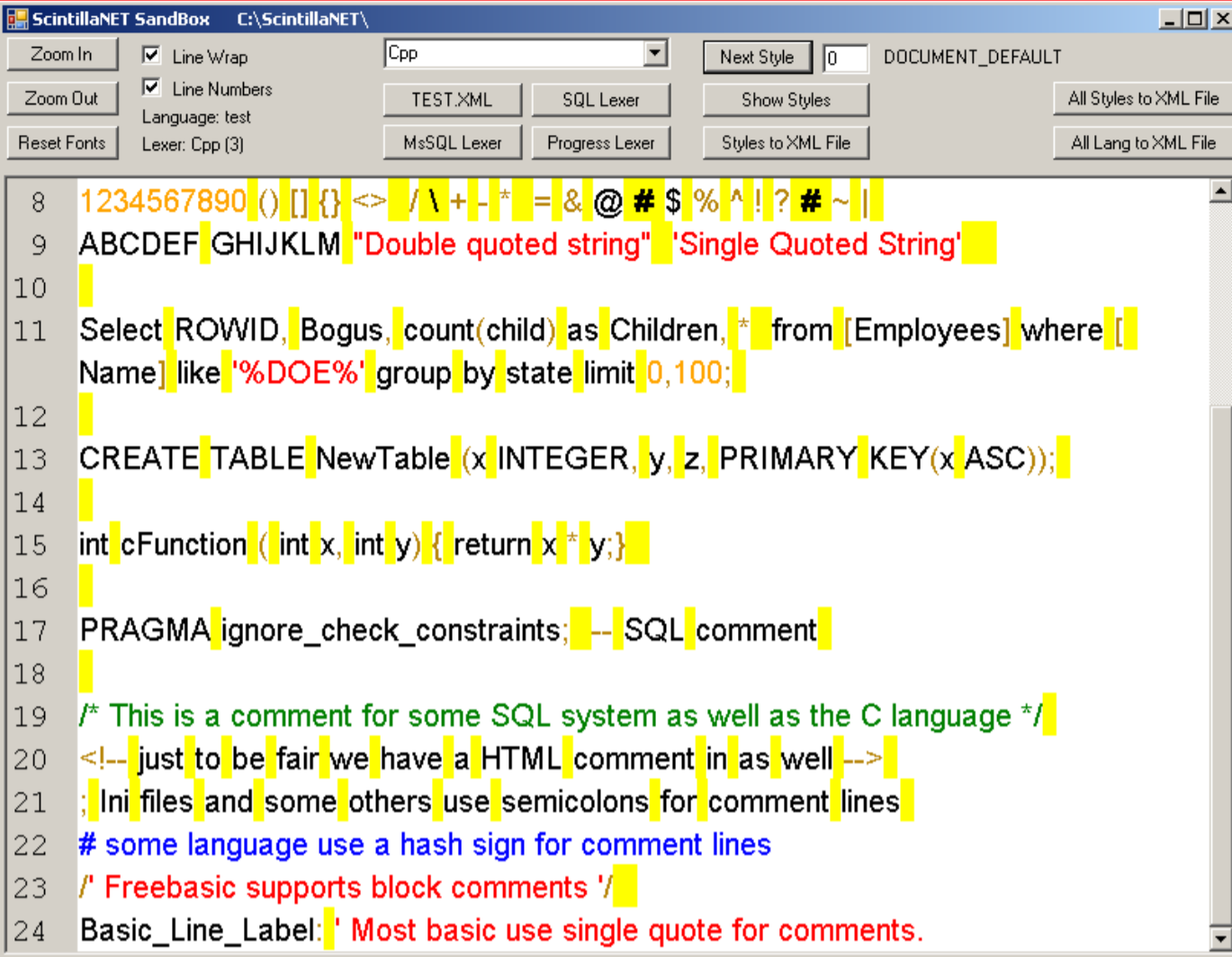
I was working on creating a XML language file for the SQLite database engine. The Scintilla.DLL has three SQL Lexers. I added the three Lexer buttons so that I could switch between these Lexers. In doing so I discovered a small disconnect between ScintillaNet and Scintilla. Whenever Scintilla changes Lexers it automatically loads the Styles for that Lexer. When ScintillaNet set a language it also configures each of the styles. It keeps a list of these styles and their attributes internally. However when ScintillaNet changes the Lexer it does NOT reconfigure the styles and thus its internal style table is totally bogus. Lesson learned: always use the XML file to configure the Lexer.

Now we come to the real heart of the application:



The button “Next Style” steps through each of the styles for the current Language/Lexer changing the the Font to “**Arial Black**” and the background to “**Yellow**”. By observing the changes in the editor box one can determine what each style is used for. In the case shown above style number 16 is named WORD2 and used for Keyword List “1”.

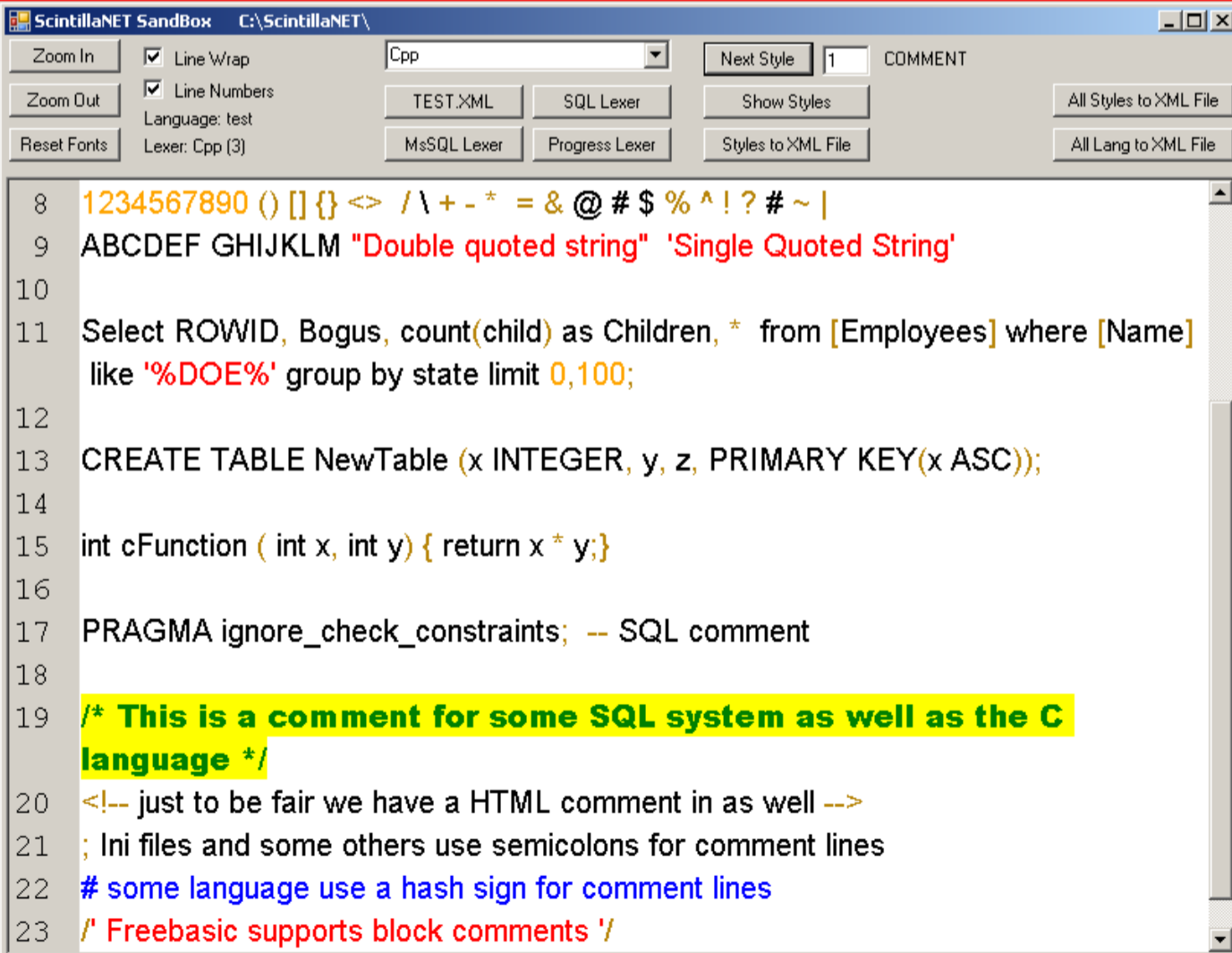
This examples shows Style 0 is named "DOCUMENT_DEFAULT" and used for "white space" and "End of Line" characters. Also note that it catches several special characters as well.



The screenshot shows the ScintillaNET SandBox interface with the following settings: Language: test, Lexer: Cpp (3), Next Style: 0, DOCUMENT_DEFAULT. The code editor displays the following text with various characters highlighted in yellow:

```
8 1234567890 () [] {} <> / \ + - * = & @ # $ % ^ ! ? # ~ |
9 ABCDEF GHIJKLM "Double quoted string" 'Single Quoted String'
10
11 Select ROWID, Bogus, count(child) as Children, * from [Employees] where [Name]
   like '%DOE%' group by state limit 0,100;
12
13 CREATE TABLE NewTable (x INTEGER, y, z, PRIMARY KEY(x ASC));
14
15 int cFunction (int x, int y) { return x * y; }
16
17 PRAGMA ignore_check_constraints; -- SQL comment
18
19 /* This is a comment for some SQL system as well as the C language */
20 <!-- just to be fair we have a HTML comment in as well -->
21 ; Ini files and some others use semicolons for comment lines
22 # some language use a hash sign for comment lines
23 /* Freebasic supports block comments */
24 Basic_Line_Label: ' Most basic use single quote for comments.
```

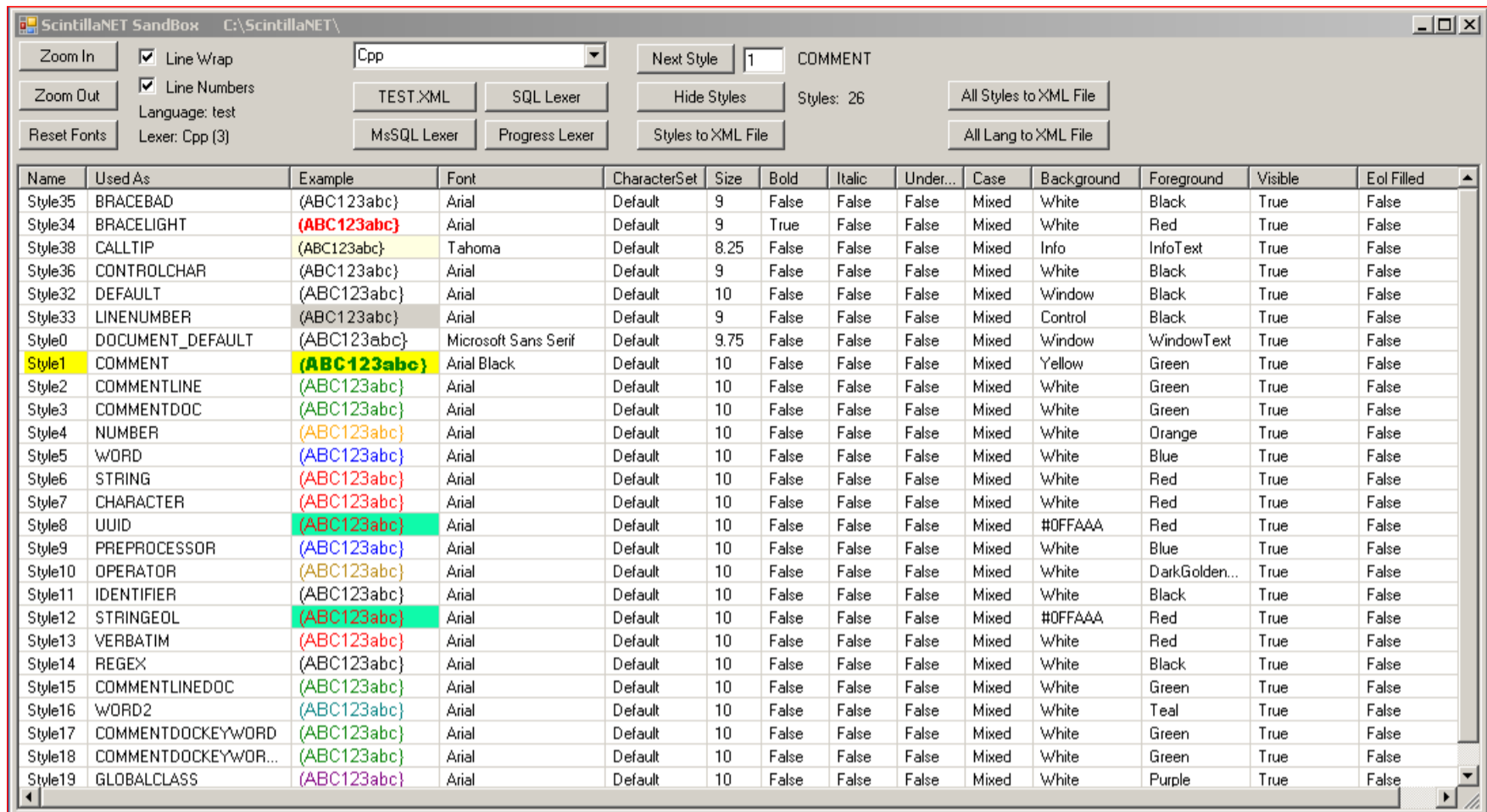
This example shows that Style 1 is named "COMMENT" and used for "C" language style block comments (*the Lexer in this case is "cpp"*).



The screenshot shows the ScintillaNET SandBox interface with the following settings: Language: test, Lexer: Cpp (3), Next Style: 1, COMMENT. The code editor displays the following text with block comments highlighted in yellow:

```
8 1234567890 () [] {} <> / \ + - * = & @ # $ % ^ ! ? # ~ |
9 ABCDEF GHIJKLM "Double quoted string" 'Single Quoted String'
10
11 Select ROWID, Bogus, count(child) as Children, * from [Employees] where [Name]
   like '%DOE%' group by state limit 0,100;
12
13 CREATE TABLE NewTable (x INTEGER, y, z, PRIMARY KEY(x ASC));
14
15 int cFunction ( int x, int y) { return x * y; }
16
17 PRAGMA ignore_check_constraints; -- SQL comment
18
19 /* This is a comment for some SQL system as well as the C
   language */
20 <!-- just to be fair we have a HTML comment in as well -->
21 ; Ini files and some others use semicolons for comment lines
22 # some language use a hash sign for comment lines
23 /* Freebasic supports block comments */
```

The “**Show Styles**” button replaces the edit box with a list view that displays the current styles and their attributes according to the ScintillaNet style table. Note that the current style is indicated by the Yellow background for the Scintilla.dll style name (*that corresponds to its style number*). You may need to expand the application window or scroll to see all the styles and their attributes `<!-- HEY! I just spotted a BUG in my software. I will be right back -->`. The number of style is shown next to the button which is now labeled “**Hide Styles**”.



The last button in this set is labeled “**Styles to XML File**”. It writes all the styles for the current Lexer to an XML file with a filename of the form: “**Lexer(#)-styles.xml**” where “**Lexer**” is the ScintillaNet Lexer name and “**#**” is the Scintilla style number. Note that the Font Name, Size and Character set are not included. There is a reason for that.

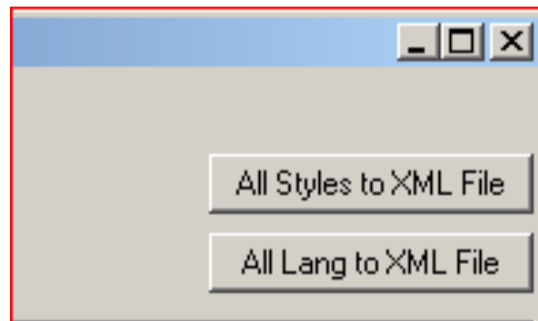
```
<?xml version="1.0" encoding="UTF-8"?>
<!-- These are all of the styles ScintillaNet Lexer: "Cpp" -->
<!-- The Scintilla.dll uses a Lexer number: "3" -->
<!-- The same Scintilla.dll internal lexer may be used for several ScintillaNet languages. -->
<!-- The styles with a dot '.' in the name are skipped because they are inherited. -->
<!-- ALL attributes (except font name, size and chr set) have been filled in. -->
<!-- ScintillaNet Sandbox by: Lewis Balentine, www.keywild.com -->
<ScintillaNET>
  <Language Name="Cpp">
    <Styles>
      <Style Name="BRACEBAD" BackColor="White" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="BRACELIGHT" BackColor="White" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="True"/>
      <Style Name="CALLTIP" BackColor="Info" ForeColor="InfoText" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="CONTROLCHAR" BackColor="White" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="DEFAULT" BackColor="Window" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="LINENUMBER" BackColor="Control" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="DOCUMENT_DEFAULT" BackColor="Window" ForeColor="WindowText" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENT" BackColor="Yellow" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENTLINE" BackColor="White" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENTDOC" BackColor="White" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="NUMBER" BackColor="White" ForeColor="Orange" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="WORD" BackColor="White" ForeColor="Blue" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="STRING" BackColor="White" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="CHARACTER" BackColor="White" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="UUID" BackColor="#OFFAAA" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="PREPROCESSOR" BackColor="White" ForeColor="Blue" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="OPERATOR" BackColor="White" ForeColor="DarkGoldenrod" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="IDENTIFIER" BackColor="White" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="STRINGEOL" BackColor="#OFFAAA" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="VERBATIM" BackColor="White" ForeColor="Red" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="REGEX" BackColor="White" ForeColor="Black" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENTLINEDOC" BackColor="White" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="WORD2" BackColor="White" ForeColor="Teal" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENTDOCKEYWORD" BackColor="White" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="COMMENTDOCKEYWORDERROR" BackColor="White" ForeColor="Green" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
      <Style Name="GLOBALCLASS" BackColor="White" ForeColor="Purple" EolFilled="False" UnderLine="False" Italic="False" Bold="False"/>
    </Styles>
  </Language>
</ScintillaNET>
```

It also places a “tab delimited” copy on the Windows clipboard that can be pasted into Excel or another program for documentation purposes (*you will need to do a little formatting and fill in the notes and sample columns*).

Style No	Style Name	Note	Sample	Back Ground	Fore Ground	Bold	Italic	Under Line	Eol Fill
35	BRACEBAD			White	Black	FALSE	FALSE	FALSE	FALSE
34	BRACELIGHT			White	Red	TRUE	FALSE	FALSE	FALSE
38	CALLTIP			Info	InfoText	FALSE	FALSE	FALSE	FALSE
36	CONTROLCHAR			White	Black	FALSE	FALSE	FALSE	FALSE
32	DEFAULT			Window	Black	FALSE	FALSE	FALSE	FALSE
33	LINENUMBER			Control	Black	FALSE	FALSE	FALSE	FALSE
0	DOCUMENT_DEFAULT	White Space		Window	WindowText	FALSE	FALSE	FALSE	FALSE
1	COMMENT	Block comment	/*string*/	Yellow	Green	FALSE	FALSE	FALSE	FALSE
2	COMMENTLINE	Line Comment	/ String	White	Green	FALSE	FALSE	FALSE	FALSE
3	COMMENTDOC	(no matches)		White	Green	FALSE	FALSE	FALSE	FALSE
4	NUMBER		1234567890	White	Orange	FALSE	FALSE	FALSE	FALSE
5	WORD	List 0		White	Blue	FALSE	FALSE	FALSE	FALSE
6	STRING	Double Quoted	“String”	White	Red	FALSE	FALSE	FALSE	FALSE
7	CHARACTER	Single Quoted	‘String’	White	Red	FALSE	FALSE	FALSE	FALSE
8	UUID	(no matches)		#OFFAAA	Red	FALSE	FALSE	FALSE	FALSE
9	PREPROCESSOR		# anything	White	Blue	FALSE	FALSE	FALSE	FALSE
10	OPERATOR	Special character	()[]<>/+ - *=&%^!~	White	DarkGoldenrod	FALSE	FALSE	FALSE	FALSE
11	IDENTIFIER	List 2,4,5 and 6		White	Black	FALSE	FALSE	FALSE	FALSE
12	STRINGEOL	(no matches)		#OFFAAA	Red	FALSE	FALSE	FALSE	FALSE
13	VERBATIM	(no matches)		White	Red	FALSE	FALSE	FALSE	FALSE
14	REGEX	(no matches)		White	Black	FALSE	FALSE	FALSE	FALSE
15	COMMENTLINEDOC	(no matches)		White	Green	FALSE	FALSE	FALSE	FALSE
16	WORD2	List 1		White	Teal	FALSE	FALSE	FALSE	FALSE
17	COMMENTDOCKEYWORD	(no matches)		White	Green	FALSE	FALSE	FALSE	FALSE
18	COMMENTDOCKEYWORDERROR	(no matches)		White	Green	FALSE	FALSE	FALSE	FALSE
19	GLOBALCLASS	List 3		White	Purple	FALSE	FALSE	FALSE	FALSE

By the way the first six of these are the defaults for Scintilla.dll and will appear in every style group. As you can see from my example here I had a number of “no matches”. That is most likely because my sample text did not have the appropriate keywords or syntax. You can get around this problem for XML based languages by creating a text file for the language. It needs to be named “Lang_TestText.txt” where “Lang” matches the Language name of the XML file. This file will need to be located in the same directory as the Language XML file.

That brings us to the Last Two buttons.



The first “**All Styles to XML File**” writes all styles for all Lexer Languages to an XML file. It does this by going through and setting the current ScintillaNet language for each language and then going through the style table. Thus it must reset the Language when done. The second button “**All Lang to XML File**” does the same thing but writes the Languages and Lexer names to an XML file. The files will be named “DefaultStyles-All.xml” and “Languages-All.xml”.

That is it for the application. If you want to add another feature change the way it operates then the source code is available from my Web site: www.keywild.com.

Creating a ScintillaNet XML Language file

I began this wild goose chase so that I could create a ScintillaNet XML language file for SQLite. Along the way I learned a few things. ScintillaNet uses the Microsoft Visual Studio XML reader. This is NOT a fault tolerant piece of software. An error anywhere in the file will cause it to abort. Among other things if you use the same field name twice it errors out. Example:

```
<Style Name="BRACEBAD" ForeColor="Black" ForeColor="White" />
```

Two dashes in any HTML style comment will cause it to go to never-never land. Example:

```
<!-- this will bring the XML reader to its knees -- or send it out to lunch -->
```

All Keywords must be lower case. I believe that this comes from Scintilla.dll not ScintillaNet.

Scintilla uses style 32 as the default font for all the others. Always include this as the first style in your style list to define the Font Name and Size. For the rest of styles leave the Font name and sizes out unless you want to override the default. DOCUMENT_DEFAULT should be the second style and may be completely blank.

ScintillaNet XML Structure

```
<?xml version="1.0" encoding="UTF-8"?>
```

Required header EXACTLY as it is shown.

```
<!-- This is a language configuration file for ScintillaNET -->
```

Always a good idea to include comments to indicate the purpose of the file (*you may forget in a year or ten years*).

```
<ScintillaNET>
```

Required Tag for ScintillaNET

```
<Language Name="default" >
```

You can have more than one language section in a file but you must have at least one.

```
<AutoComplete ... >
```

```
<List>
```

This is an option Language section that needs research and documentation.

You will have to dig into the original Scintilla Docs and the ScintillaNet program code.

There is an example of this section in the ScintillaNet cpp.xml file.

```
</List>
```

```
</AutoComplete>
```

```
<Indentation TabWidth="4" SmartIndentType="cpp" />
```

Indentation is another optional section that needs research and documentation.

This example is from the ScintillaNet cpp.xml file.

```
<Lexer LexerName="FreeBasic" LineCommentPrefix="' ' >
```

The all-important Lexer section is required. Lexer name required. Comments definition is optional.

This is an optional section if you are just defining styles.

```
<Properties>
```

This is another optional section that needs research and documentation.

There is an example of this section in the ScintillaNet mssql.xml file.

```
</Properties>
```

```
<Keywords List="0" >
```

You can have seven (0 through 6) keyword lists

```
</Keywords>
```

```
</Lexer> End of Lexer section
```

```
<Styles> Define or modify style attributes in this section
```

```
<Style Name="DEFAULT" FontName="Arial" FontSize="10" />
```

I recommend that you always define the default font unless you want to guess what it will be.

```
<Style Name="DOCUMENT_DEFAULT"/>
```

I cannot explain it but this is another style that I would always include. It just seems to work better.

Follow with any other styles you wish to modify.

```
</Styles> End of Style Section
```

```
<Commands Inherit="True">
```

This is another optional section that needs research and documentation.

There is an example of this section in the ScintillaNet html.xml file.

```
</Commands>
```

```
<Snippets>
```

This is another optional section that needs research and documentation.

There is an example of this section in the ScintillaNet cpp.xml file.

```
</Snippets>
```

```
</Language> End of language section
```

```
</ScintillaNET> End of file
```

As you can see there are a lot of optional things that you can do via the XML language file but you will need figure out how as the examples produce more questions than answers.

SQLite.XML

This is the SQLite.xml file that I currently have. It still needs some work but it does function.

```
<?xml version="1.0" encoding="UTF-8"?>
- <ScintillaNET>
  - <Language Name="sqlite">
    <!-- SQLite use both SQL and C style comments -->
    - <Lexer StreamCommentSuffix=" /*" StreamCommentPrefix="/* " LineCommentPrefix="--" LexerName="MsSql">
      - <Properties>
        <Property Name="fold.comment" Value="1"/>
      </Properties>
      <!-- basic SQL statements -->
      <Keywords List="0"> list0 abort action add after all alter analyze and as asc attach autoincrement before begin between
        by cascade case cast check collate column commit conflict constraint create cross current_date current_time
        current_timestamp database default deferrable deferred delete desc detach distinct drop each else end escape
        except exclusive exists explain fail for foreign from full glob group having if ignore immediate in index indexed
        initially inner insert instead intersect into is isnull join key left like limit match natural no not notnull null of offset
        on or order outer plan pragma primary query raise references regexp reindex release rename replace restrict right
        rollback row savepoint select set table temp temporary then to transaction trigger union unique update using
        vacuum values view virtual when where rowid oid </Keywords>
      <!-- data types -->
      <Keywords List="1"> list1 null integer real text glob blob varchar numeric datetime date boolean charater </Keywords>
      <!-- system tables, pragmas -->
      <Keywords List="2"> list2 application_id auto_vacuum automatic_index busy_timeout cache_size cache_spill
        case_sensitive_like checkpoint_fullfsync collation_list compile_options count_changes data_store_directory¹
        database_list default_cache_size defer_foreign_keys empty_result_callbacks encoding foreign_key_check
        foreign_key_list foreign_keys freelist_count full_column_names fullfsync ignore_check_constraints
        incremental_vacuum index_info index_list integrity_check journal_mode journal_size_limit legacy_file_format
        locking_mode max_page_count mmap_size page_count page_size parser_trace query_only quick_check
        read_uncommitted recursive_triggers reverse_unordered_selects schema_version secure_delete
        short_column_names shrink_memory synchronous table_info temp_store temp_store_directory user_version
        vdbe_addoptrace vdbe_debug vdbe_listing vdbe_trace wal_autocheckpoint wal_checkpoint writable_schema
        </Keywords>
      <!-- GLOBAL VARIABLES -->
      <Keywords List="3"> list3 now localtime utc strftime julianday datetime time date unixepoch weekday </Keywords>
      <!-- functions -->
      <Keywords List="4"> list4 </Keywords>
      <!-- stored procedure, no equivalent-->
      <Keywords List="5"> list5 </Keywords>
      <!-- not sure use what they have -->
      <Keywords List="6"> list6 between and or union like is not in like match regexp case exists count average min max
        </Keywords>
    </Lexer>
  - <Styles>
    <Style Name="DEFAULT" BackColor="Window" ForeColor="WindowText" FontSize="10" FontName="Arial"/>
    <Style Name="DOCUMENT_DEFAULT"/>
    <!-- list 0 -->
    <Style Name="STATEMENT" ForeColor="Blue" Bold="true"/>
    <!-- list 1 -->
    <Style Name="DATATYPE" ForeColor="Purple"/>
    <!-- List 2 -->
    <Style Name="SYSTABLE" ForeColor="Red"/>
    <!-- list 3 -->
    <Style Name="IDENTIFIER" ForeColor="Black"/>
    <!-- list 4 -->
    <Style Name="FUNCTION" ForeColor="Fuchsia" Bold="true"/>
    <!-- list 5 -->
    <Style Name="STORED_PROCEDURE" ForeColor="Darkgoldenrod"/>
    <!-- list 6, only because that is all that is left -->
    <Style Name="OPERATOR" ForeColor="Darkgray" Bold="true"/>
    <!-- Comments -->
    <Style Name="COMMENT" ForeColor="Green" Italic="True"/>
    <Style Name="LINE_COMMENT" ForeColor="Green" Italic="True"/>
    <!-- [String] -->
    <Style Name="COLUMN_NAME_2" ForeColor="Black"/>
  </Styles>
</Language>
</ScintillaNET>
```

WEB Links :

[Scintilla](#): Scintilla Home page

[ScintillaNet](#): ScintillaNet Home Page

[SQLite](#): SQLite Home Page

Shameless Self Promotion:

[KeyWild SQLite](#): My SQLite projects (***ScintillaNet SandBox project is on this page***)

[KeyWild Arduino](#): My Arduino projects (*ATMEL ATmega based Nano Digital Thermometer*)

[KeyWild CAD Library](#): A Public Domain Library of DWG files for use with CAD systems (*Fasteners, World Map, etc.*)

[Back Office, Six Mountains](#): A novel that I wrote (*published on Amazon Kindle*)

[Keywild.com home page](#)