
Subject: [PROPOSAL] CppCheck support
Posted by [Oblivion](#) on Sun, 22 Jan 2023 17:00:46 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,

I've been using CppCheck for a long time, and found it very useful. It is a very popular free static analyzer tool with lots of features and it is easy to integrate.

While its own parser is very good, it also provides an experimental clang AST backend, so it is very flexible.

I am currently integrating it with theIDE, and the integration is very smooth. So I am proposing to add cppcheck support to TheIDE.

My Plan is to make it available to public before U++ 2023 1, in a usable/stable state.

What it requires.

0. CppCheck binary.

1. A single file in TheIDE's codebase (CppCheck.cpp)
2. Detection of CppCheck binary (HasCppCheck() function)
3. A settings pane, in theIDE's settings dialog (will be available only if the binary is detected).
4. Color entries in (Settings/Syntax Highlighting pane) for CppCheck error severity types (style, warning, performance, portability, etc.)

5. It will have the ability to check

- a) A single file
- b) A package
- c) Workspace (all packages)

This will require adding menu and toolbar entries when cppcheck binary is detected.
(Similar to existing "Compile \$FILENAME" or "Build \$PACKAGE" entries -> "Analyze \$PACKAGE")

See the below screenshot for a working version (initial)

I will create a branch (theide_cppcheck) in my fork of upp and make the source code available so others can participate.

What do you think?

Best regards
Oblivion,

File Attachments

1) [Ekran Görüntüsü - 2023-01-22 19-41-52.png](#) , downloaded
928 times

UWord

- Core
- CtrlCore
- CtrlLib
- Draw
- Painter
- PdfDraw
- RichEdit
- RichText
- plugin/bmp
- plugin/jpg
- plugin/png
- plugin/z
- <prj-aux>
- <ide-aux>
- <temp-aux>
- <meta>

- # RichText.h
- Object.cpp
- RichImage.cpp
- # Para.h
- ParaData.cpp
- ParaType.cpp
- ParaPaint.cpp
- # Txt.h
- HeaderFooter.cpp
- TxtData.cpp
- TxtPaint.cpp
- TxtOp.cpp
- Format.cpp
- # Table.h
- TableCell.cpp
- TableLayout.cpp
- TablePaint.cpp
- TableData.cpp
- # Text.h
- TextPaint.cpp
- TextStyle.cpp
- TextData.cpp
- TextTable.cpp
- EncodeQt.cpp
- ParseQt.cpp
- EncodeHTML.cpp

```

95     case PARA:
96         if(parti == r_parti) break;
97     case FROM:
98         r_parti = min(parti, r_parti);
99         r_type = FROM;
100        break;
101    }
102 }
103
104 void RichTxt::SetRefreshFrom(int parti)
105 {
106     r_type = FROM;
107     if(r_type == NONE)
108         r_parti = parti;
109     else
110         r_parti = min(parti, r_parti);
111 }
112
113 void RichTxt::Put(int i, const RichPara& p)
114 {
115     if(i >= part.GetCount() || !IsPara(i))
116         part.At(i).Create<Para>();
117     Para& pp = part[i].Get<Para>();
118     int numbering = p.format.GetNumberLevel();
119     if(pp.numbering != numbering)

```

File	Line	Message
RichText/Para.h	100	variable 'angh' is assign
RichText/Para.h	108	Variable 'fillchar' is assi
RichText/RichText.h	392	Variable 'next' is assign
RichText/TxtData.cpp	18	Variable 'styleid' is assi
RichText/TxtData.cpp	19	Variable 'content' is ass
RichText/RichText.h	33	Struct 'Zoom' has a con
RichText/RichText.h	109	Struct 'PageRect' has a
RichText/RichText.h	448	Struct 'QtRichObject' ha
RichText/RichText.h	532	Struct 'SimplePageDraw
RichText/RichText.h	544	Struct 'PrintPageDraw' h
RichText/RichText.h	530	The function 'Page' over
RichText/RichText.h	541	The function 'Page' over
RichText/TxtData.cpp	422	The function 'operator()
RichText/TxtData.cpp	107	Condition 'r_type==NO
RichText/TxtData.cpp	212	Local variable 'length' s
RichText/TxtData.cpp	243	Local variable 'i' shadow
RichText/TxtData.cpp	265	Local variable 'i' shadow
RichText/TxtData.cpp	367	Local variable 'i' shadow