

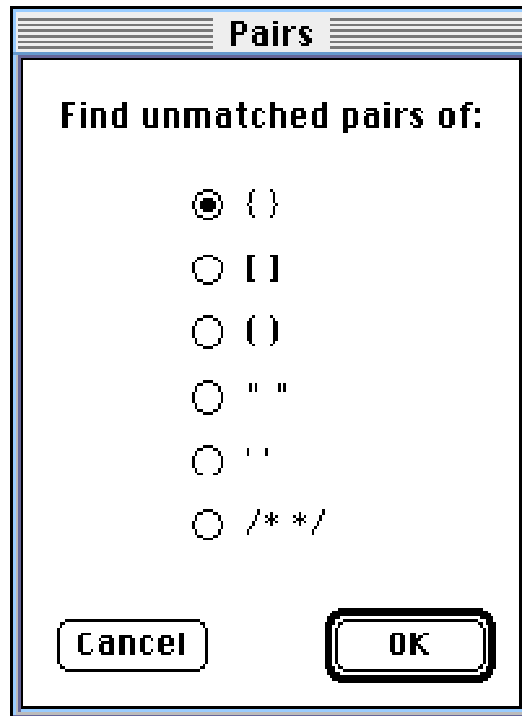
## Documentation for “Pairs” extension for BBEdit

Pairs is an extension for BBEdit that will look for unmatched pairs of characters or tokens in C or C++ source code that may cause unexpected program behavior or may prevent the code from compiling. The character or token pairs that can be checked are:

```
{ }  
[ ]  
( )  
" "  
' '  
/* */
```

Pairs will check any of BBEdit's editable windows containing text. If the window contains selected text, only the selection will be checked. For example, you can select the text of a single function, and check just it. If there is no selection, the document will be checked from the current insertion point to the end.

After dragging the Pairs extension to the “BBEdit Extensions” folder in your BBEdit folder, the Pairs extension will show up in BBEdit’s “Extensions” menu. Selecting it brings up the Pairs dialog:



This dialog will be centered over the window being checked. It can be dragged using the dialog's title bar, and the underlying BBEdit windows can be repositioned by first pressing the Command key and then dragging these windows by their title bars. Unlike most movable modal dialogs, Pairs will **not** allow you to switch to another running application, so the application and Apple menus remain grayed out.

The pairs to be checked are indicated by the radio buttons running down the center of the dialog. The return and escape keys provide shortcuts for selection OK and Cancel, respectively.

Once OK is selected, the pair checking begins. If no problems are found, an alert like the one below appears:



The alert is placed in standard alert position over the window being checked. As with the Pairs dialog, BBEdit's underlying windows can be repositioned by command dragging. The alert, of course, can not be repositioned.

Of course, mismatches do occur. What is deemed a mismatch or error depends on the kind of check being performed. Pairs always tries to be intelligent about quoted text, so that a single brace found with single or double quoted text is not flagged. The same is true of text commented using either the standard C comment characters `/* */` or C++ style line comments `/**/`.

Pairs will always complain about nested C style comments. If you check text containing something like the following:

```
int i,j;
for (i = 0; i < 10; i++)
{
    j = i; /* I don't know why I'm doing this /* a nested quote */ */
}
```

Pairs will complain:

```
for (i = 0; i < 10; i++)  
{  
  j = i; /* I don't know why I'm doing this /* a nested quote */ */  
}
```



This happens no matter what pair is being checked. The offending comment start is selected. Similarly, if an extra C style end comment “\*/” is found, the message “Mismatched C-style “/\* \*/” comment found!” will be displayed.

Apart from this exception, a check for a mismatched token pair will only complain about mismatches of that type. If the file contains a mismatched parenthesis, but you only check for curly brackets, the mismatched parenthesis won't be flagged.

Quote checking is special in that Pairs will complain if there is a mismatched quote within a single line of text. The assumption is that you're checking C code, and not War and Peace. As always, Pairs tries to be intelligent about quoting. It knows about escaping single and double quotes. For example:

```
"Double quoted text with an embedded single quote ' here is OK"  
'Single quoted text with an embedded double quote " here is OK'
```

```
"Text with an embedded, escaped double quote \" here is OK"  
'Text with an embedded, escaped single quote \' here is OK'
```

```
This is a "double" but not a 'single' quote error"  
This is a 'single' but not a "double" quote error'
```

When a mismatch is found, the line containing the mismatch is selected.

#### **Requirements:**

Pairs requires BBEdit 2.2 or later, and System 7.0 or later.

#### **Legal stuff:**

Pairs is supplied as a BBEdit extension plus full source code in the form of a Metrowerks CodeWarrior project. Although every effort has been made to make Pairs bug free, the user of the software takes full responsibility for its use. The extension and the source code may be freely distributed and modified so long as its is not sold for profit, and credit is given to the original author (me).

Brian Clark  
b-clark@nwu.edu