Smart Indent (Draft)

By Sam Harwell

Getting Started

Without any form of indentation assistance (e.g. Notepad), pressing enter results in the caret positioning itself at column 1 of the next line, and the only change to the text buffer is the addition of a newline character. Smart Indent can alter two aspects of this:

- The content of either of the two lines involved in the split.
- The position of the caret.

Smart Indent is invoked when one of the following occurs:

- A new, blank, line is inserted in the document via:
    - Pressing enter at the end of a line.
    - For Smart Indent purposes, the following “special” insertions are treated as though enter were pressed at the very end of the line preceding the inserted line.
        - Pressing insert line above (Ctrl+Shift+Enter in Visual Studio).
        - Insert line below (Ctrl+Enter in Visual Studio).
        - Random insertion?
- A line is split by pressing enter at some point before the end of the line. The split can occur:
    - At the beginning of the line (similar to insert line above, but the cursor stays on the line with text instead of moving to the inserted blank line).
    - After the beginning of the line, but before the first non-whitespace character.
    - After the first non-whitespace character, but before the last non-whitespace character.
    - After the last non-whitespace character, but before the end of the line.
    - For splits occurring within a comment, there is an additional notion of the set of characters from first non-whitespace character to the last character before the comment-specific text (CI). In the following lines, these characters are highlighted:

```
/* Here is a block comment
 * that spans multiple lines
 */
// Here is a line comment that does not.
```

Definitions

Line Insertion occurs when a new blank line is created in the document with the cursor located on that line.

Line Split occurs when a line is inserted in the document resulting in the cursor being positioned on a line containing text.

Virtual Caret Placement

For any case where the caret is directed to be placed at the end of a line containing only indenting whitespace, the cursor should instead be placed in virtual editing space at that column and no whitespace characters actually entered in the text buffer until the user types a key.

Comment Continuation Mode

The conditions for using comment continuation mode are easily identified by the current state of the syntax highlighter. This mode is used when either of the following two conditions holds:

- A line split occurred within a single-line comment. This region of code is underlined in the following example:

  ```
  // Here is a line comment.
  ```

  - The text on line 1 of the split is not changed.
  - For line 2:
    - The whitespace is stripped from the beginning of the line.
    - The prefix of the previous line (includes leading whitespace and the comment intro as highlighted in the Getting Started section) is identified and prepended to line 2.
    - The cursor is placed after the last inserted prefix character.
- A line split occurred within a block comment.
  - Line 1 is unchanged.
  - If line 2, trimmed-start, starts with the block comment termination character(s), as would occur if enter were pressed in any of the following underlined regions, then:

    ```
        /* Here is a block comment
         * that spans multiple lines
    _____*/

    /* Here is a different one._*/

    int line; /* Yet one more.
    ___________*/
    ```

    - The whitespace is stripped from the beginning of the line.
    - Identify the column in which the comment is first opened and build a prefix containing whitespace up to this column.
    - If the user prefers, add one more space to align asterisks and form comments like are shown in the example above.
    - Prepend the calculated line prefix to the trimmed-start line 2.
    - The cursor is placed after the inserted whitespace.
  - If line 2, trimmed start, starts with a trimmed non-terminating vertical continuation prefix, as would occur if enter were pressed in the following underlined region, then:

    ```
        /* Here is a block comment
    _____* that spans multiple lines
         */
    ```

    - The whitespace is stripped from the beginning of the line.

        - Identify the column in which the comment is first opened and build a line prefix containing whitespace up to this column.
        - If the user-specified vertical continuation prefix includes leading whitespace, such as [space]-[asterisk] like I've been using, append the continuation prefix's leading whitespace to the calculated line prefix.
        - Prepend the calculated line prefix to the trimmed-start line 2.
        - The cursor is placed after the inserted whitespace.
    - If line 2, trimmed-start, does not terminate the block comment, as would occur if enter were pressed in any of the following underlined regions, then:

```
    /* Here is a block comment
     * that spans multiple lines
     */

/* Here is a different one. */

int line; /* Yet one more.
           */
```

        - The whitespace is trimmed from the beginning of the line.
        - Identify the column in which the comment is first opened and build a line prefix containing whitespace up to this column (A).
            - To this line prefix, append the vertical continuation prefix (B) followed by a user-specified number of padding spaces (C) (in the example above, this is always 1).
        - Identify the last (if any) line containing one or more non-whitespace characters that is located after the comment is opened but at or before Line 1 of the split. If this line starts with the calculated line prefix, then if any spaces exist between the prefix and the next non-whitespace character, include those in the calculated prefix (D).
        - Prepend the calculated line prefix to the trimmed-start line 2.
        - The cursor is placed after the inserted whitespace.
        - The following is an example inserting characters after enter was pressed in the underlined location. The added characters are highlighted based on their addition points above: A, B, C, D.

```
    /* Here is a block comment.
     *   And it has an indented section,_a long one at that.
     */

    /* Here is a block comment.
     *   And it has an indented section,
     *   a long one at that.
     */
```

- A line insertion happens, and the line before the inserted line ends as an un-terminated block comment.
    - The inserted line is treated like Line 2 in the previous case where the trimmed-start line 2 is an empty string.

Standard Mode

Standard Mode deals with indenting blocks of code.

I am so not even here yet.