# MES : Mozile Editing Scheme

## *Mozile 0.6 Series for Firefox 1.5*

Author: Max d'Ayala
Date: 27 January 2006
Content: MES Configuration and Commands

The MES file defines the toolbar commands and other options for an editable page. The MES file to load is defined in the page RSD Configuration file. The MES options are applied for every editable region on the page, and for that page only. It is not anticipated in the Mozile 0.6 Series that a single page will be configurable with more than one MES.

A default MES, providing a toolbar for XHTML page editing, is loaded if a custom MES file is not specified.

Refer to the main Mozile documentation for details of the tags and attributes required to define the toolbar buttons and command menus.

# Editing Overview

Editing works on the principle of converting displayed text into lines based on the CSS *"display"* property. The easiest way to understand the basics is to look at some XHTML examples.

## *CSS Lines*

### Contained and Bounded Lines

In this example the `<div>` is the top container. The top container can **never** be modified, only its contents may be edited. There are three CSS Lines.

```
<div class="editable">
      <p>This is a Contained Line.</p>
      This is a <strong>Bounded Line</strong>.
      <p>Another <em>Contained</em> Line.</p>
</div>
```

The first paragraph is a Contained Line because the text has a container. i.e the line is contained within the `<p>...</p>` tags. These have a CSS display value of "block".

The second line of text does not have its own container but instead is bounded at either end by the two paragraphs. The blue text is an example of a Bounded Line.

Note that the `<strong>` and `<em>` tags have a CSS *display* value of "inline" and do not form lines of their own. They are inline elements within the CSS Line.

Contained lines can be of three basic types based on their CSS *display* value.

- Block (e.g. p, h1)
- List-item (e.g. li )
- Table Cell (e.g. td, th)

It is worth looking at one other example that may occur if you start from an empty editable area, or if all the content is deleted.

```
<div class="editable">This is a Contained Line.</div>
```

In this case the text is still contained, this time by the block level `<div>.`

### Whitespace

The CSS *white-space* property determines how whitespace (spaces, tabs, newlines, etc) is displayed. Usually *white-space* has the value *normal* and any whitespace used to make the XHTML markup readable is filtered out. When *white-space* has the value *pre* the whitespace is displayed exactly as it occurs in the markup. In XHTML *white-space: pre* is normally applied to the `<pre>` element by default.

Note: the eDOM handling of *white-space: pre* is probably not as well tested as *whitespace: normal*, so please report any bugs.

### Editing Token

Block level elements must contain editable text or an Editing Token to make them editable. The Editing Token used is a non-break space character, Unicode 160 decimal, A0 hex, or   as an XHTML character entity. Currently the editing token can not be easily changed to another value.

This correctly defines an empty editable area:

```
      <div class="editable">&amp;nbsp;</div>
```

If you miss out the Editing Token you won't be able to select the div to begin editing:

```
<div class="editable"></div>
```

## Normalization

Mozile works on the Document Object Model (DOM) of the document once the page has been loaded, not from the raw source of the original file. This means there may by some changes to the whitespace of the document and to the order of some attributes. In general these small changes are not important, but if you wish to preserve the non-editable sections of the document exactly as you wrote them you will need to save just the editable areas, or extract them if saving the entire document.

The eDOM functionality relies on Text nodes remaining normalized. That is, when new Text nodes are added they should always be merged with any adjacent Text nodes. Failure to do this will often lead to Mozile generating an error, sometimes when you *next* try to execute a command on these text nodes. If you get any unexpected errors like this it is worth using the DOM inspector to check for Text nodes that have not been normalized. If you find any, try and work out what command you executed to produce the error and then report the bug.

Inline elements, such as the `<span>` element used to style text in XHTML documents, are not nested. Nested and crossing styles are produced by splitting the span at each style boundary.

The following text displayed in the browser;

A sentence with **bold text and a nested *italic* word.**

might be generated by Mozile like this:

```
<p>A sentence with
 <span style="font-weight: bold;">bold text and a nested </span>
 <span style="font-weight: bold; font-style: italic;">italic</span>
 <span style="font-weight: bold;"> word.<span>
</p>
```

When changes are made to the document similar Styles may be merged, with either inline elements or the line container, to minimise the number of Style declarations.

## Comments

In theory Comment nodes should be filtered out by Mozile and should not affect editing. In practice this is not always the case and comments in the editable markup can cause errors. For this reason it is a good idea not to include comments in the editable region.

# MES Configuration : Commands Summary

| | Attribute | |
|---|---|---|
| Element : command<br>E.g. <command type="toggleStyle" name="bold" key="b"... etc ... /> | | |
| **Common attributes** | | |
| | name | Command name; must be unique. |
| | title | Text displayed in a drop down menu. |
| | tooltip | Text displayed when you hover over the button. |
| | icon | Image to use for the button. |
| | modifiers | Short cut modifier key.<br>Not currently used - always Ctrl/Meta key. |
| | key | Short cut key(s).<br>E.g. "b" for Ctrl+b, "B" for Ctrl+shift+b, "Bb" for both. |
| | type | Command type.<br>Additional command specific attributes are detailed below. |
| **Type attribute** | | |
| **setLinesContainer** | | Sets the *container* for Contained or Bounded Lines. |
| | namespace | Optional |
| | tag | |
| **unformat** | | Removes the *container* from all Contained Lines of type Block. |
| **wrapText** | | Wraps the selected text with *tag*. |
| | namespace | Optional |
| | tag | |
| **toggleAttribute** | | Toggles attribute between value and default.<br>If default is an empty string the attribute is removed. |
| | attribute | e.g. class |
| | value | |
| | default | |
| **styleLines** | | Applies CSS *property* and *value* to Contained and Bounded Lines**.** |
| | property | |
| | value | |
| **styleText** | | Applies CSS *property* and *value* to the selected Text. |
| | property | |
| | value | |
| **toggleStyle** | | Toggles CSS *property* between *value* and *default* for the selected Text. Multiple properties can be specified as a comma separated list. |
| | property | e.g. font-weight |
| | value | bold |
| | default | normal |
| | activeproperty | Optional. The number of properties to use when updating the active toolbar. |
| **indent** | | Adds to the *margin-left* Style. |
| | value | Optional. Default = +40px |
| **outdent** | | Subtracts from the *margin-left* Style. |
| | value | Optional. Default = -40px |

| | | |
|---|---|---|
| **toggleListLines** | | Toggles between Lines and Lists ( "ul" or "ol" ). |
| | tag | List container to create. ("ul" or "ol") |
| | old | List containers to be changed to *tag*. ("ul" or "ol") |
| **indentLists** | | Indents lists. |
| **outdentLists** | | Outdents lists. |
| **link** | | Applies a link to the selected text or modifies an existing link. ( <a href="url">...</a> ) |
| **unlink** | | Removes links from the selected text. |
| **table** | | Inserts a table. |
| **image** | | Inserts an image. |
| **hr** | | Inserts a horizontal rule. |
| **toolbar** | | Fixed toolbar button configuration.<br>The Common attributes are not currently used. |
| | name | command name |
| | disabled | true |

# Individual MES Commands

## *setLinesContainer*

**Sets the *container* for Contained or Bounded Lines.**
**Arguments: *namespace, tag***

The *namespace* argument is optional.

Works for a single line if the selection is collapsed, or changes all lines within a selection.

If it is a Contained Line and the *container* is top, a table element or a list element the existing *container* is not changed but a new container (of type *tag*) is created.

Warning : Does not check that *tag* is a valid descendent of container.

## *unformat*

**Removes the *container* from all Contained Lines of type Block.**
**Arguments: none.**

Works for a single line if the selection is collapsed, or changes all lines within a selection.

Removes the line container for a Contained Line of type Block.

Has no effect on Bounded Lines, or Contained Lines of type Table-cell or List-item.

## *wrapText*

**Wraps the selected text with *tag*.**
**Arguments: *namespace, tag***

The *namespace* argument is optional.

Works across all text lines within a selection (wraps the text of each line), or an individual text fragment within a line. Does nothing if the selection is collapsed.

Warning: does not check that wrapping text with *tag* is valid.

## *toggleAttribute*

Toggles *attribute* between *value* and *default*.
**Arguments: *attribute, value, default***

This is a rather basic command in how it is applied. Essentially it toggles *attribute* between *value* and *default*. *Attribute* is only applied once, effectively to the first element of the selection.

If default is an empty string the *attribute* is removed if *attribute* equals *value*.

# XHTML Specific Commands

## *styleLines*

**Applies CSS *property* and *value* to Contained and Bounded Lines.**
**Arguments: *property, value***

Works for a single line if the selection is collapsed, or styles all lines within a selection.

Bounded Lines are converted to Default Contained Lines before being styled. Adds the XHTML style attribute to the line container.

## *styleText*

**Applies CSS *property* and *value* to the selected Text.**
**Arguments: *property, value***

Works across all text lines within a selection (Styles the text of each line), or an individual text fragment within a line. Does nothing if the selection is collapsed.

Applies  tags as required.

Note: applying a Style that the text already inherits will result in *property* being removed.

## *toggleStyle*

**Toggles CSS *property* between *value* and *default* for the selected Text.**
**Arguments: *property, value, default, activeproperty***

Works similarly to styleText.

If property is not equal to value, property is set to value.
If property is equal to value, property is set to default.

Multiple properties can be toggled using a list of comma separated arguments.

## *indent*

**Adds to the *margin-left* Style.**
**Arguments: *value***

*value* is optional, if not supplied a default value of +40px is applied.

Works for a single line if the selection is collapsed, or all lines within a selection.

Applies the style to the Line *container*.

Indents Contained Lines.
Bounded lines are converted to Contained Lines and then indented.
List containers (ul and ol) are indented.

## *outdent*

**Subtracts from the *margin-left* Style.**
**Arguments: *value***

Similar to indent, although margin-left value can't go negative.
*value* is optional, if not supplied a default value of -40px is applied.

## *toggleListLines*

**Toggles between Lines and Lists ( "ul" or "ol" ).**
**Arguments: *tag, old***

Either tag=ul and old =ol, or, tag=ol and old=ul.

Works across all Lines within a selection.

Contained and Bounded Lines that are not already list lines are changed to list-items.

List Containers of type *old* are changed to type *tag.*

Adjacent lists of the same type, provided they are in the selected range, are merged.

If there are only list item lines of type *tag* in the range the
list-items are outdented one level. Adjacent lists are NOT merged.

## *indentLists*

**Indents lists.**
**Arguments: none**
Code is part written but incomplete. Creates nested lists.

## *outdentLists*

**Outdents lists.**
**Arguments: none**
Code is part written but incomplete. Removes nested lists.

## *link*

**Applies a link to the selected text or modifies an existing link.**
**Arguments: none**

Applies the link element across the selected text or Lines.

If the selection is collapsed within an existing link it can be edited or removed.

## *unlink*

**Removes links from the selected text.**
**Arguments: none**

Removes all links from the selection. If the selection is collapsed and there is a parent link it is removed.

The following commands work in limited circumstances but could do with being improved or expanded.

### *table*

**Inserts a table**
**Arguments: none**

You are prompted to enter the number of rows and columns for the table.
There are no other table specific commands to manipulate tables.

### *image*

**Inserts an image**
**Arguments: none**

### *hr*

**Inserts a horizontal rule**
**Arguments: none**

Often nests incorrectly in block level elements such as paragraphs.
(General problem of inserting empty elements with the eDOM.)

**Other undocumented development features that may or may not be useful...**

### *x-element*

**inserts an xhtml element**
**Arguments: *tag***

Might work if you wish to insert an individual xhtml element.

### *mozileCommand*

**Execute one of the other commands.**
**Arguments: commandname**

Possibly useful if you want to have completely custom toolbar/menu system.

# MES Configuration : eDOM

The default container for XHTML editing can be specified.
Without this option the default container is set to .

## *MES Example : Set default container to <p>*

For a typical editing scheme you may have the editable region set up within the xhtml  tag.
Within this editable region you may wish to have the default container set to be a paragraph tag <p>.

```
<editingScheme ... >

      ... MES commands ...

        <eDOM>
                <xhtml>
                        <container default="p"/>
                </xhtml>
        </eDOM>

</editingScheme>
```

# MES Configuration : Fixed Toolbar Items

This allows you to disable (hide) any of the fixed toolbar items.

The following command names are recognised;

1. cut
2. copy
3. paste
4. undo
5. redo
6. view-tags
7. edit-source
8. special-characters
9. save
10. bug
11. command-menu

## *MES Example : Hide fixed toolbar buttons.*

```
<editingScheme ... >

        <commands>

                <command name="view-tags" type="toolbar" disabled="true"/>
                <command name="edit-source" type="toolbar" disabled="true"/>
                <command name="bug" type="toolbar" disabled="true"/>
                <command name="command-menu" type="toolbar" disabled="true"/>

                ... other command tags...

        </commands>

</editingScheme>
```

Note that the fixed commands such as Cut, Copy, Paste, Undo, Redo and Save have hard-coded keyboard shortcuts (in the mozileFirefox.xul overlay file) that can be used instead. The remaining items do not have keyboard shortcuts and are completely disabled. It wouldn't be too much work to add a "key" attribute to the above to make all the shortcut keys configurable via the MES.

# MES Configuration : Special Characters

## *MES Example*

This is a simple example of a custom special character definition added to a MES file.
It will add a single named block to the predefined character blocks.

```
<editingScheme ... >

      ... MES commands ...

      <specialchars insert="add">
            <specialentry entry="none"/>
            <specialblock name="MES-Test 1" columns="16">
                  <characters>77,69,83,S,84,69,83,84</characters>
                  <range start="160" end="255" wrap="16"/>
            </specialblock>
      </specialchars>

</editingScheme>
```

| Element | Attribute name<br>value | Notes |
|---|---|---|
| specialchars | | Should only occur once. |
| | insert | Optional. Default: insert="add" |
| | replace | Replace the predefined special character blocks. |
| | add | Add to the predefined character blocks. |
| | | |
| specialentry | | Should only occur once.<br>Optional. Default: entry="any" |
| | entry | |
| | none | Don't allow manual Unicode entry. |
| | any | Allow entry of any Unicode value. |
| | | |
| specialblock | | Optional. Up to 30 blocks can be defined. |
| | name | This is the name shown in the drop down menu.<br>The displayed name is truncated to 30 characters in length. |
| | columns | Number of columns as an integer.<br>Maximum of 50. |
| | | |
| You must include at least one <characters> or <range> element. You can include multiple <character> and <range> elements in any order. Each element starts a new row of characters. | | |
| | **Text content** | |
| characters | | Comma separated list of Unicode characters that will wrap at the "columns" value specified in <specialblock>. |
| | n | integer - Unicode value |
| | "S" | space |
| | **Attribute name** | |
| range | | Contiguous range of Unicode characters. |
| | start | Start value (integer) |
| | end | End value (integer) |
| | wrap | Length of lines. (Must be <= columns) |

# KEYBOARD : Shortcuts and special keys

| |
|---|
| ***Ctrl/Meta Key : Hard Coded Shortcuts***<br>Note that the Caps Lock key is ignored for Ctrl key combinations.<br>So "x" means the x key with or without Caps-Lock pressed. |
| **Defined in the overlay file mozileFirefox.xul** |

| | |
|---|---|
| Ctrl-x | Cut |
| Ctrl-y | Paste |
| Ctrl-z | Undo |
| Ctrl-shift-z | Redo |
| Ctrl-shift-s | Show save dialog. |

| |
|---|
| **Hard-coded in mozileKeyboard.js** |

| | |
|---|---|
| Ctrl-s | Save with current settings. |

| |
|---|
| ***MES Ctrl/Meta Key shortcuts*** |

| | |
|---|---|
| User Defined | As specified by the MES key attribute in <command> definitions. *** |

| |
|---|
| ***Other Key Combinations*** |

| | |
|---|---|
| spacebar | single space character |
| spacebar repeated | string of alternate space and non-break space characters. |
| Shift+spacebar | single non-break space |
| Tab | single tab character for text styled with "whitespace: pre". |
| Shift+Return | xhtml line break <br/><br>[Note eDOM handling of break is very buggy.] |

***On my system not all keyboard shortcuts work. For example, pressing Shift and "= "on my keyboard gives the plus sign "+" during normal typing. However, pressing Control-Shift and "=" doesn't produce any sort of keypress event in the browser. Pressing Control and "+" on the numeric keypad does. All normal letter keys seem to work correctly when Ctrl-shifted. The odd non-letter key works correctly when shifted, but most don't. I don't know if this is a "feature" of the keyboard controller or the browser.