# Mozile Development

**Mozile 0.6.26 - 25 May 2006**

## *Known Bugs and Issues*

Testing with Firefox 1.5.0.3 on Windows.

### Caret Movement and Navigation

1. **General Caret Movement Problems**
   The eDOM cursor manipulation is not very good. E.g. It can't move the cursor correctly to the start/end of a line, it always goes to the start/end of the Line container. The normal browser caret navigation has improved with Firefox 1.5, and using the Home/End keys now moves the cursor correctly to the start/end of line. The only problem with this is that if you have tag view enabled using the browser navigation Home/End keys move to the start/end of the tags, not the start/end of the editable text. This causes an error with Mozile because the cursor is not then in an editable position. It is maybe better to live with this than go to the start/end of a block. A similar argument applies to the Up/Down keys. You can choose which method you want by commenting out the relevant lines in mozileKeyboard.js. Ideally, at some point the eDOM navigation needs to be improved to hook into the existing browser caret movement functions, if this is possible. Don't forget, that the eDOM should ideally constrain the cursor movement to the editable area and not let you move out of it. Just using the built in browser navigation doesn't let you do this. In most cases, if you are editing a large block of text it doesn't matter. But for anyone who wants to edit small chunks of text (e.g. just the odd word here and there) constraining the cursor to a fixed area is much more important.

2. **Caret Movement around end of line in "*white-space: pre*" sections**
   The caret is incorrectly placed at the end of the previous line when you try to place it at the start of a line using Javascript. This bug has been submitted to Bugzilla and it should be fixed in future versions of Firefox. It should be possible to add a workaround to the Mozile code. (To do.)

3. **Caret movement around the XHTML <br/> element.**
   Can produce a similar error to the white-space:pre problem. Needs to be checked to see if it as an eDOM or browser issue. Again, an annoying bug that makes using breaks unpleasant. (Also need to handle the automatic creation/deletion of the Editing Token when <br/> is at the end of a line.)

4. **Caret disappears**
   Sometimes the caret disappears for no good reason and you have to click around to try and get it back. It will appear in some pieces of text but not others. If you know where it should be you can usually still carry on typing and everything else appears to work correctly. Sometimes closing the source edit window if it is open makes the caret reappear.

### Detection and Wakeup

1. Toolbar detection has been improved when switching between multiple browser tabs, but it is still not complete and probably never will be. Ideally, switching to a new tab would always reset the Mozile toolbar to match the configuration for that page. If you are editing multiple documents in multiple tabs, each document could have a different MES and hence a different toolbar. At the moment, most of the time you would probably have to click in an editable area to reset the toolbar to its correct state. In reality this is only a minor issue, because not many people edit multiple documents with different MES at the same time.

2. If Mozile "detect and edit" is switched OFF autohiding will no longer work. Hence, to get rid of the toolbar you have to go via a roundabout route. From the main Firefox menu select View >

Toolbars and then deselect "Mozile Toolbar" to hide the toolbar. Similarly if the Mozile toolbar is hidden and you want to disable it you first have to show the Mozile toolbar to access the options. Or you could access the Mozile Options from the Tools > Extensions menu without having to show the Mozile Toolbar. [I don't think this will change, autohiding could be made to work when Mozile is OFF, but it would kind of defeat the whole purpose of having an ON/OFF switch.]

## MES

1. Key overrides not yet implemented. The purpose of these is to re-map standard keyboard commands such as enter, tab, etc to other commands to give complete control over the editing environment. The data is currently parsed from the MES, but nothing is done with it.

2. More of the basic editing commands could be made available via the MES. E.g. insert character, split line, etc. Also, there aren't any of the simple "advanced" commands that might be useful such as split an element, delete an element, etc.

3. Only one MES can be used per page. Technically it wouldn't be difficult to do because essentially it would just involve parsing and storing more data, then selecting the correct MES for each editable area. The "class" attribute could be used for xhtml, but this wouldn't work for xml in general. [If implemented, it would be useful to tie in with saving of multiple areas.]

4. **font-weight: bold**
   Note that the CSS 2.1 spec does not provide for an equivalent mapping between the font-weight scale (100, 200, ...,900) and the *bolder* value. The default value for the bold tag is *bolder*, whereas for editing you are likely to use the absolute scale. Hence, if you want the bold tag <b> to show up as equivalent to *font-weight: bold*, which itself is equivalent to *font-weight: 700*, you must redefine the default style for the bold tag. i.e. Put this in your CSS file:
   ```
   b {
     font-weight: bold;
   }
   ```

## Cut and Paste

1. Only *text* cut and paste is currently working. Most of the code for *xml* and *xhtml* cut and paste seems to be there but needs finishing off.

## Saving

1. Saving all editable areas, or all those that have been modified, is not implemented.

## Source Edit

1. When the content-type is "text/html" the source window always shows the code in HTML format. (i.e. lonely tags such as <img> don't have a trailing slash.) Ideally if the DTD is an XHTML version it should show the code in xhtml (compatible) format.

## Browser bugs

1. **DOM Inspector / Venkman** has changed for Firefox 1.5. All the getters and setters are now being evaluated for some reason. (Or at least, that is how it appears to me.) Maybe this is a feature, although I doubt it, because why would you need to execute all the setters? If the getters are all being executed without any write buffering then there is the potential to change properties without meaning to. This is because some getters modify properties; they just execute functions and are not necessarily read only. Having said that, there aren't too many eDOM getters that modify properties so the chances of errors occurring may not be that great. Presumably, if you don't open the DOM Inspector or Venkman then everything should be ok. Haven't had time to get to the bottom of this but any thoughts about this issue welcome.

2. **Javascript *instanceof* in XUL**
   Another bug that has appeared with Firefox 1.5. For some reason the Javascript code:
   myarray *instanceof* Array
   has stopped working in two of the .JS files used with XUL dialogs (info.js and save.js). It is now returning false all the time, but the code worked fine in Fx 1.0. Any ideas on this one?

3. **Special Chars XUL Dialog**
   Window doesn't always redraw correctly on the right hand side when you make a new selection from the dropdown box. Looks like a 1 pixel strip on the inside edge of the right border is not being filled in.

4. **Mozile Debug**
   Messages are not being displayed in the status bar for some reason.

5. **CSS Numbering** can cause problems if you try to select text next to the number with the mouse. The selection appears and then disappears.

## Javascript warnings and errors

1. Always on the lookout for these! Mozile should work in strict Javascript mode and not produce any warnings or errors. Please report any reproducible warnings or errors.

## eDOM Limitations

1. Problems navigating *white-space: pre* end of lines.
2. XHTML <br/> not fully implemented; problems with navigation.
3. Problems when the editable container is a paragraph or inline element rather than a *div*. Not all commands check when adding or splitting nodes that what they are doing is allowable according to the XHTML DTD. (Some commands have been fixed, others to do.)
4. Editing Token - can not be easily changed.
5. Backspace deletes the container of a Token Contained Line if it is the top line. There should be an option to prevent the deletion of the container.
6. eDOM JSUnit test need rewriting.

## Namespace issues

1. Report any namespace issues.

## XML Specific Issues

1. Report any XML specific issues.

## XHTML features not implemented

1. Forms.
2. Tables, other than basic table insertion.
3. Definition lists
4. Nested lists (ol/ul) not complete.
5. Adding unique page bookmarks/ids.
6. Editing of page title.
7. Editing and insertion/deletion of meta tags.
8. Images - very basic. Can't link, edit properties, upload image files.

9. Proper handling of the class attribute. i.e. commands to style using class, and also to correctly normalize class attributes.
10. Proper handling of inline elements such as <b>, <strong>, etc.

## Browsers

1. Only works in Firefox 1.5 but should work in Firefox 1 and Seamonkey with minor changes to the browser overlay file and the install script.

## Minor Bugs

1. Some minor problems converting between Bounded and Contained Lines. The Contained Lines sometimes end up with invalid whitespace at the start / end of the line.

2. Switching between Bounded and Contained Lines - cursor sometimes moves - due to whitespace handling issues.

3. Inserting an image at the end of a Contained Line after another image - cursor is left before the inserted image instead of at the end of the line.

4. mozileDataTransport.js - mozTransportConvertToUnicode
   var u = UC.ConvertToUnicode( content );
   Saving to dialog. Sometimes produces an error. E.g. When you use the XHTML mime type and UTF-8 charset the conversion to the required display format doesn't happen and the unconverted data is displayed. I don't think it affects the actual save data, which has to be converted back again to display it correctly in the dialog. (Or something like that, needs a closer look. Maybe it is because the data is already in Unicode format, so the call to ConvertToUnicode is not actually required? If this is the case then there is nothing really to worry about. )

[Inserting images is not really sorted out but it appears to work well enough in most cases. Not sure what happens if you set the CSS *display* value for images to *block* or *inline*, or if Mozile is meant to handle this automatically - there doesn't appear to be any code for this. The eDOM code used to insert images says it shouldn't be used for images and similar xhtml empty elements...]

### Bug Reports

Please report any bugs you find. They might not get fixed very quickly, but if you don't report them they probably won't get fixed at all. There are several different versions of Mozile. The 0.5 series is no longer under development. This document refers to the 0.6 series. The 0.7 series is a complete redesign of the 0.6 series. In general, bugs will only be considered for the latest version of the Mozile 0.6 Series. If you report a bug for an old version, the first thing you will be asked to do is verify that bug still exists in the current version. There are also a few cross-platform issues, so it helps if you say what sort of OS you are using.

Max d'Ayala