

EditLive! 9 Upgrade Guide

EditLive! 9 contains a broad range of new functionality including media functionality, HTML5 support, new user interface options and more. This document describes the major changes in EditLive! 9 and how to upgrade your existing EditLive! deployment to take advantage of them.

- [Licensing](#)
- [New Editor Features Including Toolbar Buttons and Menu Items](#)
 - [Social and Embedded Media](#)
 - [HTML5 Media - AUDIO and VIDEO Tags](#)
 - [Page Width and Mobile Preview](#)
 - [HTML5 Semantic Elements](#)
 - [Enhanced HTML Support](#)
 - [New Toolbar Megamenus](#)
- [New User Interface Options](#)
- [New Style Options](#)
- [Platform Support](#)
- [Behavioural Changes](#)
 - [Loading Behaviour](#)
 - [Updated EditLive! UI](#)
 - [List Toolbar Buttons Converted to Megamenus](#)
 - [User-friendly Style Names](#)
 - [Accessibility Checks](#)
 - [Media Preview](#)
 - [Create and Remove Section Commands](#)
 - [HTTP Clients](#)
 - [raiseEvent API Passes the Applet ID](#)
 - [WebDav Functionality Removed in 9.1](#)
 - [JavaScript APIs Removed in 9.1](#)
 - [Java APIs Removed in 9.1](#)

Licensing

All EditLive! users will be required to upgrade their license to use EditLive! 9. An example version 9 license key looks like the following:

```
<ephoxLicenses>
  <license
    domain="LOCALHOST"
    key="6FFF-7C38-9CFD-AE9B"
    licensee="For Evaluation Only"
    release="9.0"
    type="Evaluation License"
    productivityPack="true"
  />
</ephoxLicenses>
```



The version number in your license key must say 9.0.

To get your EditLive! 9 license, please log a support case in the [Tiny support system](#).

New Editor Features Including Toolbar Buttons and Menu Items

Social and Embedded Media

EditLive! 9 includes upgraded media support, with enhanced embedding functionality and support for social media services like YouTube, Vimeo, Flickr, Google Maps, SlideShare and other web-based media aggregator services like [embed.ly](#). This functionality makes embedding social and cloud-based media as simple as creating a hyperlink. Users need only copy the URL of the resource (usually the browser's URL) into EditLive!'s media dialog and then the oEmbed standard is used to obtain the media at that URL and embed it into the content in EditLive!.

This functionality has been designed as a replacement for the process of creating an OBJECT tag - a process that's often too complex for content authors - and avoids the need to access EditLive!'s code view to paste in HTML code.

Content is inserted via the *Insert Media* dialog which supports adding content from services that support the [oEmbed specification](#) or by insertion of an arbitrary HTML fragment to represent rich media.

For more information on Social and Embedded Media support, including configuration information, refer to the Developer Guide [Using Social Media and External Media Services](#).

HTML5 Media - AUDIO and VIDEO Tags

EditLive! 9 introduces support for insertion of HTML5 <video> and <audio> tags. This is achieved through the Video and Audio tabs on the *Insert Media* dialog with placeholder images displayed within EditLive!. This functionality works with content that's already available on web servers.

The following video formats are supported:

- MP4
- OGG
- WebM

The following audio formats are supported:

- WAV
- MP3
- OGG

For more information on HTML 5 Audio and Video support refer to the Developer Guide [Embedding Media Using HTML5](#).



Note

HTML5 media cannot be played back in EditLive!'s Design view. In this view the media will be inserted with an appropriate place holder.

Media can be played back in the Preview tab.

Page Width and Mobile Preview

EditLive! 9 introduces the ability to render your content for a specific page width with the new Page Width function. The functionality is particularly useful when content being created by users may be read on mobile devices in addition to desktop machines. It is preconfigured with common page widths corresponding to mobile and desktop devices.

Setting the page width is via the Page Width toolbar button which provides a selection of common page widths. These include

- Smartphone Portrait
- Smartphone Landscape
- Tablet Portrait
- Tablet Landscape
- Monitor
- Widescreen Monitor
- Size to Fit
- Custom

For more information on including the Page Width on your toolbar refer to the *View Commands* section of the [Menu and Toolbar Item List](#).

HTML5 Semantic Elements

EditLive! 9 supports the broadest range of HTML5 markup of any rich text editor. In addition to the HTML5 media tags listed previously it also supports the full range of HTML5 semantic tags and ensures the validity of any content created containing these tags. This markup provides extra semantic meaning within pages and can be particularly significant for accessibility or providing extra styling flexibility.

Users can create new content containing these tags using the Create Section command and any existing content containing this markup can now be edited by EditLive!. Designers can also create templates for use with EditLive! that include this markup.

EditLive! 9 includes rendering and editing of HTML5 semantic block elements including:

- Section
- Header
- Footer
- Article
- Aside
- Nav
- Figure
- FigCaption

HTML 5 semantic block elements can be inserted using the *Create Section* toolbar button and removed using the *Remove Section* toolbar button.

Enhanced HTML Support

EditLive! 9 has enhanced support for the following HTML tags:

- Blockquote - This tag can now be inserted as a standalone tag or wrapped around existing content using the Blockquote option on the *Create Section* menu.
- DIV - The rendering for DIVs has been improved and they have been added as an option on the *Create Section* menu.
- iframe - EditLive! 9 provides an improved rendering for this tag using a placeholder icon and the height and width of the iframe.

New Toolbar Megamenus

EditLive! 9 introduces megamenus that can be added to the toolbar and menus. When used on the toolbar megamenus are represented by a single toolbar icon with a chevron next to it to indicate more options are available. When used on a menu these megamenus appear as submenus.

Megamenus serve two key purposes:

- they make it easier for users to access related functionality without opening a properties dialog or a menu and;
- they enable developers to optimize the use of space on the toolbar, which can be useful when screen real estate is at a premium - with in-place editing use cases for example.

Megamenus are a collection of functionality that are represented as a single toolbar button that has an expandable menu associated with it. The items on megamenus are not customizable.

The following megamenus are available.

Menu or Tool Tip Text	Function and items	Image
Text Formatting	Megamenu of formatting commands, consisting of: <ul style="list-style-type: none"> • Bold • Italic • Underline • Superscript • Subscript • Strikethrough. 	
Alignment and Indents	Megamenu of Alignment and Indenting commands, consisting of: <ul style="list-style-type: none"> • Align Left • Align Center • Align Right • Decrease Indent • Increase Indent 	
Proofing Tools	Megamenu of Proofing commands, consisting of: <ul style="list-style-type: none"> • Spelling • Disable Spell Checking as you type • Link Checking • Accessibility Report • Enable Accessibility as you type • Word Count 	
Help	Megamenu of help and debugging commands, consisting of: <ul style="list-style-type: none"> • Help • About EditLive! • Enable Debug Logging 	
Page Width	Megamenu providing access to common screen widths for mobile, tablet and desktop devices. The following widths and commands are available: <ul style="list-style-type: none"> • Smartphone Portrait • Smartphone Landscape • Tablet Portrait • Tablet Landscape • Monitor • Widescreen Monitor • Size to Fit - Fit to the width of the current editor instance • Custom - User is prompted for a custom width. 	
Ordered and Unordered Lists	By default these megamenus insert the standard list types. Using the chevron opens a panel that enables the user to select other standard HTML list types.	

Create Section	<p>Megamenu enables the user to insert or wrap content in one of the following block tags:</p> <ul style="list-style-type: none"> • DIV • Blockquote • Section • Header • Footer • Article • Aside • Nav • Figure • FigCaption 	
Remove Section	<p>Megamenu enables the user to remove a blog tag within the HTML structure at the current cursor location:</p> <ul style="list-style-type: none"> • DIV • Blockquote • Section • Header • Footer • Article • Aside • Nav • Figure • FigCaption <p>Removal of the tag does not remove the content contained within the tag</p>	

For more information on the configuration settings for these refer to the appropriate sections in [Menu and Toolbar Item List](#)

New User Interface Options

EditLive! 9 introduces a set of new configuration options and JavaScript APIs that allow you to create a minimalist editing UI for In-place editing. The new UI Configurations include the ability to

- float the main toolbar
- dock contextual toolbars
- remove the Preview, Design and Code view tabs
- minimise the toolbars by using the new megamenu collections described above
- remove the menus

In addition, you can use the new JavaScript APIs to

- close EditLive! when the end user clicks outside of the editable section
- vertically expand and contract content editable sections to better fit the content they contain.

For an example of using these new features refer to the quick start guide [Creating a Minimal Editing UI with In-place Editing](#).

New Style Options

EditLive! 9 introduces the ability to both Blacklist or Whitelist the styles in the style sheet supplied to EditLive!. Whitelisting is where only those styles explicitly marked for inclusion in the stylesheet will appear in the Styles drop down. Blacklisting is where all styles, except those that are explicitly excluded will appear in the Styles drop down.

For more detailed explanation refer to the Developer Guide [Restricting what CSS classes appear in EditLive! styles drop down](#).

Platform Support

EditLive! now requires Java 1.6 as the minimum version of Java.

Google Chrome on OS X 10.7 and above will now use Select Edit to display TinyMCE instead of EditLive!. This is because Chrome on OS X is a 32 bit application, however the Oracle Java plugin is a 64 bit application and will not function with Chrome as it is a 32 bit application.

For more information see the [System Requirements](#) section of this documentation.

Behavioural Changes

Loading Behaviour

EditLive! 9 features an updated loading screen that improves the user experience with more subtle load time visuals. The traditional loading dialogue with a progress meter has been replaced with a dimmed view of the Preview pane and spinner while EditLive! loads in the background. Once EditLive! has loaded the user immediately sees the editor interface.

On-click Tab Switch

To enhance the user experience of EditLive!'s inline editing mode users no longer have to click on the Design tab to load the editing interface for an inline instance of EditLive!. In EditLive! 9 clicking within the Preview pane of an in-line instance of EditLive! will load the editing interface of EditLive!. The exception to this is if the user clicks on an interactive component within the preview - e.g. a Google Map, YouTube video etc - that interaction will not trigger a change from Preview to Design mode.

Updated EditLive! UI

EditLive! 9 includes an updated user interface. While Tiny has worked to ensure the changes to existing UIs is minimal this update does affect some of the widths and heights of items in EditLive! and you are advised to test this with your configuration prior to updating your production environment. Changes in buttons widths in particular may cause your toolbar to wrap unexpectedly. The following lists some of the key changes:

- The introduction of chevrons to represent megamenus on the list, table and colour buttons increase the space these buttons require on the toolbar.
- The widths of the style, font and font size drop downs have been reduced. Long style or font names may now be truncated.
- The height of toolbars have been reduced.
- The spacing between toolbar button groups has been reduced.
- EditLive! now uses the Open Sans font by default. Changes to the font may affect the appearance of custom dialogs developed using the Advanced Java API.

List Toolbar Buttons Converted to Megamenus

The Ordered and Unordered List toolbar buttons have been converted to megamenus - a convention used by most word processors today. These megamenus provide access to the different HTML list types available for each list. Previously changing the list type required the user to open the List Properties dialog.

User-friendly Style Names

Class name that use naming conventions with camelCase, underscores or hyphens will be converted into user-friendly names. For more information refer to the *User Friendly Style Names* section of [Using CSS in EditLive!](#)

Accessibility Checks

The W3C accessibility guidelines have been updated from WCAG 1.0 to WCAG 2.0. As per the standard, WCAG 1.0 "1" and "2" ratings have been replaced by the WCAG 2.0 "A" and "AA" ratings.

New accessibility as-you-type checks have been added for media elements including HTML5 media and embedded media.

The accessibility as-you-type check for absolute vs relative widths on DIVs and tables has been removed as it is no longer valid for WCAG 2.0.

Media Preview

HTML5 and interactive embeddable media elements are now able to be previewed in the EditLive! Preview pane. They can also be interacted with in the Preview pane.

Create and Remove Section Commands

Create Section (DIV) and Remove Section (DIV) have been enhanced to now provide the ability to insert and remove HTML 5 Semantic tags along with DIV and Blockquote elements. The configuration setting to add to the Toolbar and Menu remains the same as previous version.

HTTP Clients

EditLive! 9.0 uses the Oracle (previously known as the "Sun") HTTP client layer to make requests by default. Most users will not notice any changes associated with this.

EditLive! 9.1 only uses the Oracle HTTP client layer. The "apache" layer has been removed.

raiseEvent API Passes the Applet ID

raiseEvent applet callbacks are now passed the applet ID (the string used to create the EditLiveJava object) as the first function parameter. This enables developers to identify the editor instance that invoked the JavaScript call.

WebDav Functionality Removed in 9.1

WebDav functionality has been removed, as of EditLive! 9.1. The following related configuration items no longer have any effect:

- [<webdav>](#)
- [<repository>](#)
- [<authentication>](#)
- [<realm>](#)

JavaScript APIs Removed in 9.1

Any calls to this function will need to be removed:

- [setHttpLayerManager Method](#)

Java APIs Removed in 9.1

See: [Java API](#)