

# Run Time Troubleshooting

This article details common errors encountered while operating EditLive! and solutions to solve these problems.

## Printing Errors

The printing architecture for EditLive! creates a new HTML web page containing the content of EditLive!. Through this new web page, users can then click the Print button to print their content based on the current printing settings configured in the user's browser.

### Printing Page Does Not Display

If the printing page does not display, check that the system EditLive! is operating on does not use webpage pop-up blockers.

### Page is Unable to Print

If the page is unable to print, an error exists either in the browser's printing settings or in the connection between the user's system and their printer. Users should consult their browser and operating system help files for more information.

## Image Insertion

Using an EditLive! configuration file, you can specify the location of a web page to list images the user can insert into an instance of EditLive!. For more information on configuring EditLive! to load this page, see the [Image Insertion Dialog's Browser Component](#) article and the `<imageBrowser>` configuration file element.

### Image Insertion Web Page Does Not Display Correctly

Unfortunately, while the page is being displayed in the Image Insertion dialog, JavaScript will not work. This may cause the appearance and functionality of the page to be compromised.

### Image Browser Dialog Does Not Appear

This error can occur if running EditLive! on Java Runtime Environment 1.4.2. This error occurs if a HTML form tag exists in the EditLive! document. The error is triggered if the HTML form tag does not include a target or action attribute. To ensure this error does not occur, include these attributes in any HTML form tag included in the EditLive! document.

## Spell Checking Errors

By default, the spell checking dictionary used with EditLive! is automatically configured to match the locale of the user's machine. However, if a developer wishes to specify a specific spell checking dictionary regardless of the user's locale, this can be done using the jar attribute of the `<spellCheck (Applet)>` configuration file element.

### Spell Checking Does Not Work

One of the most common reasons for spell checking failing is that the location of the dictionary jar file, specified in the `<spellCheck (Applet)>` configuration element, is incorrect. Check the location of the dictionary jar file again and verify this against the value specified in the `<spellCheck (Applet)>` configuration file element. Examples of EditLive! with functional dictionaries can be seen in the [Tutorials](#) section of this SDK.

## Accessibility Checking Errors

EditLive! allows users to check their document against the [W3C Accessibility Compliance Guidelines](#).

### Guideline WCAG 2 Check Name 3.3 Always Appears

This accessibility check refers changing all `<b>` and `<i>` tags in a document to `<strong>` and `<em>` respectively. When the **logicalEmphasis** attribute of the `<wysiwygEditor>` configuration file element is set to *false*, clicking the bold toolbar or menu item will insert `<b>` tags. If this attribute is set to *true*, clicking the bold link will create `<strong>` tags. If you are using accessibility checking, to avoid WCAG 2 Check Name 3.3 constantly appearing set the **logicalEmphasis** attribute to *true*.

## Saving Errors

### Browser Crashes on Form Submit

EditLive! can possibly crash your web browser if the following combination of code exists:

- EditLive!'s [setAutoSubmit Method](#) is called with *true* (this is the default value).
- A javascript call to `form.onSubmit()` is made to the html `<form>` containing EditLive!

To avoid crashing the browser, the following technique can be used:

Code your `onSubmit()` handler to return -1. Then, wherever `onSubmit()` is called, call `form.submit()` directly afterwards.