

# Using TinyMCE



Select Edit works with TinyMCE 3.x only.

The TinyMCE JavaScript editor is supported by Tiny. While EditLive! contains more extensive and powerful functionality than TinyMCE, TinyMCE does not require Java to be installed on the user's machine.

In order to allow developers to easily integrate either EditLive! or TinyMCE into their application, the EditLive! APIs have been extended to easily facilitate loading either editor. These extensions to the EditLive! APIs are referred to as the Select Edit APIs.



Before attempting to integrate TinyMCE into your application using the API calls below, please consult the [Installing TinyMCE](#) article in the Install Guide component of the EditLive! for Java SDK.

If you are upgrading from EditLive! 7.1, you will need to upgrade the legacy Tiny Express Edit JavaScript editor with TinyMCE. You will need to follow the [Installing TinyMCE](#) instructions and the integration steps listed below.

## Integrating TinyMCE using the Select Edit APIs

To enable TinyMCE in your application, take a standard implementation of EditLive! and perform the following steps:

1. Reference the `expressEdit/tiny_mce/jscripts/tiny_mce/tiny_mce.js` file located in your `redistributables/editlivejava` directory.

```
...
<script language="Javascript" src="../../redistributables/editlivejava/expressEdit/tiny_mce/jscripts
/tiny_mce/tiny_mce.js"></script>
<script language="Javascript" src="../../redistributables/editlivejava/editlivejava.js"></script>

...

<script language="Javascript">
    editlivejs = new EditLiveJava("ELApplet", 640, 400);
    editlivejs.setConfigurationFile("../../redistributables/editlivejava/sample_eljconfig.xml");
    editlivejs.show();
</script>

...
```

2. Add a call to the [SetExpressEdit](#) load-time API. In the example below, TinyMCE will load as long as the client's platform and browser is compatible with TinyMCE. (For more information on TinyMCE compatibility, please see the [TinyMCE Compatibility Chart](#))

```
...
<script language="Javascript" src="../../redistributables/editlivejava/expressEdit/tiny_mce/jscripts
/tiny_mce/tiny_mce.js"></script>
<script language="Javascript" src="../../redistributables/editlivejava/editlivejava.js"></script>

...

<script language="Javascript">
    editlivejs = new EditLiveJava("ELApplet", 640, 400);
    editlivejs.setConfigurationFile("../../redistributables/editlivejava/sample_eljconfig.xml");
    editlivejs.setExpressEdit("always");
    editlivejs.show();
</script>

...
```

## Setting Styles and Configuring Toolbar Buttons for TinyMCE

When using the Select Edit APIs to load TinyMCE, the styles and toolbar buttons specified in the EditLive! configuration file will also be applied to the TinyMCE instance. Not all toolbar buttons available to EditLive! will be applicable for TinyMCE. Please consult the [TinyMCE Known Issues](#) article for a complete list of TinyMCE-compatible EditLive! toolbar buttons.

If no style information has been specified in the EditLive! configuration file, the Select Edit APIs will search for style information from the **head** element of the document loaded into TinyMCE. If no style information exists here, style information will be loaded via the [SetStyles](#) property.

## Setting and Retrieving Content from TinyMCE

The majority of load-time and run-time APIs in EditLive! for Java are supported when integrating TinyMCE (e.g. GetBody, SetBody). For a comprehensive list of supported APIs please consult the [TinyMCE Known Issues](#) article.

### See Also

- [setExpressEdit Method](#)
- [TinyMCE Known Issues](#)