

Getting the Document in the Applet Tutorial

Getting Started

Required Skills

The following skills are required prior to working with this tutorial:

- Basic client-side JavaScript

Required Tutorials Completed

The following tutorials are required to be undertaken before attempting this tutorial:

- [Instantiation Tutorial](#)
- [Setting the Document in the Applet Tutorial](#)

Tutorial

Step 1. Create an Instance of EditLive! in a Webpage and Set the Document

As shown in the [Setting the Document in the Applet Tutorial](#), create an instance of EditLive! in a webpage and set the Document.

```
<html>
  <body>
    <script src="../../redistributables/editlivejava/editlivejava.js" language="JavaScript"></script>
    <script>
      var editlivejava = new EditLiveJava("ELJApplet", "700", "400");
      editlivejava.setConfigurationFile("../../redistributables/editlivejava/sample_eljconfig.
xml");
      editlivejava.setBody(encodeURIComponent("<p>Original <i>HTML</i> loaded into EditLive!<
/p>"));
      editlivejava.show();
    </script>
  </body>
</html>
```

Save this webpage as *getDocument.html*.

Step 2. Create a HTML Textarea and a Button.

As seen in the [Setting the Document in the Applet Tutorial](#), create a textarea and a button on the page. The purpose of the button in this tutorial will be to copy the contents of the HTML Document in EditLive! into the textarea.

```
<html>
  <body>
    <p><textarea id="documentContents" cols="80" rows="5"></textarea>
      <br/><input type="button" value="Get Document Contents" onclick="buttonPress()"></p>
    <script src="../../redistributables/editlivejava/editlivejava.js" language="JavaScript"></script>
    <script>
      var editlivejava = new EditLiveJava("ELJApplet", "700", "400");
      editlivejava.setConfigurationFile("../../redistributables/editlivejava/sample_eljconfig.
xml");
      editlivejava.setBody(encodeURIComponent("<p>Original <i>HTML</i> loaded into EditLive!<
/p>"));
      editlivejava.setUseMathML(true);
      editlivejava.show();
    </script>
  </body>
</html>
```

Step 3. Create a Javascript Function to Get the Document of EditLive! with the Textarea Contents

In order to extract the HTML Document stored in EditLive!, the [getDocument Method](#) is used.

The GetDocument run-time function requires a string parameter passed to it. This parameter is required to be the name of an existing Javascript method in the page. Once the [getDocument Method](#) is called, the Javascript property passed will then be called, passing the HTML document as a string parameter.

This tutorial contains a javascript function called *getEditLiveDocument*. When called, the string parameter passed to this function is then assigned to value of the textarea.

Hence, the [getDocument Method](#) passes the string 'getEditLiveDocument', which then calls the *getEditLiveDocument* method, passing the HTML contents of EditLive! as the src parameter.

```
<html>
  <body>
    <form name="exampleForm">
      <p><textarea id="documentContents" cols="80" rows="5"></textarea>
        <br/><input type="button" value="Get Document Contents" onclick="buttonPress()"><
      /p>
      <script src="../../redistributables/editlivejava/editlivejava.js" language="JavaScript"><
    /script>
      <script>
        var editlivejava = new EditLiveJava("ELJApplet", "700", "400");
        editlivejava.setConfigurationFile("../../redistributables/editlivejava
/sample_eljconfig.xml");
        editlivejava.setBody(encodeURIComponent("<p>Original <i>HTML</i> loaded into
EditLive!</p>"));
        editlivejava.show();

        /** Function extracts the contents of the EditLive! HTML Document
         * and displays this content in the textarea
         */
        function buttonPress() {
          // the parameter passed to the GetDocuement method is the callback
method the applet will call, passing
          // the contents of the HTML Document stored in EditLive!
          editlive.getBody('getEditLiveDocument');
        }

        function getEditLiveDocument(src) {
          document.exampleForm.documentContents.value = src;
        }
      </script>
    </form>
  </body>
</html>
```