Specifying Character Sets in the Swing SDK

Tiny EditLive! for Java Swing supports multiple character sets which allow it to be used in an international environment. Specifying the character set used by an instance of EditLive! for Java Swing defines which characters the editor can correctly render.

For a comprehensive list of the character sets supported by EditLive! for Java Swing, please see the Character Sets Supported article.

Default Character Set

If a character set is not specified, the default character set for the user's JVM will be applied to the editor. This means the editor will exhibit different behaviour depending on the system language of the current user. On English Windows systems, the default character set is ISO8859_1 (also known as CP1252). If, for example, you attempt to load content urlencoded as UTF-8 into the editor without specifying the UTF-8 characters set, a user loading the editor on English Windows would corrupt all unicode characters - eg "å" will be turned into "Ã¥". This corruption is not reversible.

Specifying EditLive! for Java Swing's Character Set

The character set used by EditLive! for Java Swing can be specified by the following:

- Using the setDocument() method of ELJBean
  A <meta> tag containing the character set specification can be added when specifying the XHTML document to load into an instance of EditLive! for Java Swing.
- Specifying the Character Set in the EditLive! for Java Swing Configuration File
  A <meta> tag containing character set specification can be nested in the <head> element of the EditLive! for Java Swing configuration file.

Using the setDocument method of ELJBean

The character set to be used within EditLive! for Java Swing can be specified in the document using the setDocument() method of ELJBean. In order to specify the character set in this way, a <meta> tag must be included into the <head> of the document to be loaded into EditLive! for Java Swing.

Example
The following instantiation of EditLive! for Java Swing shows how to specify a character set of ASCII.

```
ELJBean editlive = new ELJBean();

...

editLive.setDocument("<html><head><meta http-equiv=\"Content-Type\" content=\"text/html; charset=ASCII\" /><
/head><body><h1>Body Text</h1></body></html>");
```

> If character encoding is specified by using the setDocument() method then this will be the final character encoding used. If an instance of EditLive! for Java Swing utilizes a configuration file that specifies character encoding, the character encoding will still be set to the value specified in the document loaded into the editor via the setDocument() method.

- Specifying the Character Set in the EditLive! Configuration File
  A <meta> tag containing character set specification can be nested in the <head> element of the EditLive! for Java Swing configuration file.

Example
To specify character encoding of ASCII, the following tag should be entered between the <head> tags of the configuration file as such:

```
<editLive>
   <document>
      <html>
         <head>
            <meta content="text/html; charset=ASCII" http-equiv="Content-Type" />
            ...
         </head>
      ...
      </html>
   </document>
   ...
</editLive>
```

Specifying the Character Set In the EditLive! for Java Swing Configuration File

A meta tag explicitly specifying the character encoding can be entered into the configuration file for EditLive!.

Example
To specify ASCII character encoding, the following tag should be entered between the <head> tags of the configuration file as such:

```
<editLive>
   <document>
      <html>
         <head>
            <meta content="text/html; charset=ASCII" http-equiv="Content-Type" />
            ...
         </head>
      ...
      </html>
   </document>
   ...
</editLive>
```

Specifying the Output Character Set

The methods above specify how to define which character set is used to load and render the content in EditLive! for Java Swing. When extracting content from EditLive! for Java Swing, the default operation is to encode the outgoing content with the same character set specified for loading the content. If no character set has been specified for loading, the UTF-8 character set will be used for encoding outgoing content. Note the input vs output distinction; in such a scenario the system default would still be used for loading, which gives unpredictable results depending on user machine configurations.

Encoding characters with UTF-8 will correctly extract the content of EditLive! for Java Swing for most rendering character sets.

If you are experiencing problems regarding the content extracted from EditLive! for Java Swing not matching the content displayed in the editor, this may be due to an incompatibility between your rendering character set and the output character set. To explicitly set your output character set, use the setOutputCharset Method.

See Also

- Character Sets Supported
- <meta> Configuration File Element