IBM Connections Wikis

General guidelines for working with pasted content in Connections Wikis

Table of Contents

Introduction

The purpose of this short document is to demonstrate how some lesser-known features of the Rich Text editor could be used to improve the quality and consistency of documents, especially in the context of importing content from external sources.

The features discussed in this document include:

1. Paste from Microsoft Word (automatic)
2. Paste as Plain Text
3. Clear Formatting
4. Show Blocks

This document will also cover some general guidelines for using the HTML source tab.

Target Audience

The intended target audience of this document is any user of Connections Wikis who wishes to create Wiki pages, with an interest in reusing content from external sources.

It is assumed that the reader has a very basic understanding of HTML. Some HTML principles may be explained in this document, when an understanding of these is required to explain a given feature.

Definitions

For the purpose of this document, some terms are defined below:

Word	Definition
Block element	A HTML element that usually functions as a container for other elements and that normally renders on its own line e.g. a heading, a paragraph, a table etc.
Clipboard	The operating system's temporary storage for content that has been copied e.g. using CTRL+C
Clipboard operations	Copying and pasting e.g. using CTRL+C and CTRL+V
Plain text	Content that only contains textual data with no formatting applied. It is the simplest type of content.
Rich text	HTML content that may contain styles and formatting. Rich Text and HTML are used synonymously in this document.
Source application	The application from which content is copied with the intention of pasting it into a Wikis page.

Background on clipboard operations (copy and paste)

This information is only some useful background and could be skipped.

It may be useful to give a brief overview of the confusion many users encounter when they attempt to import content from external document sources, e.g.; Symphony, Word, Excel, PDF, HTML pages etc.

When content is copied from a source application (e.g. Microsoft Word), it is up to the source application to make that content available to the operating system clipboard. Content added to the clipboard can come in different flavors. For the time being, the two most relevant flavors are HTML and plain text. Plain text is almost always available in the clipboard after a copy operation (CTRL+C).

Not all applications are able to produce or accept all data flavors. The table below gives an example of some common applications and the way that they handle HTML and plain text content flavors.

Application	HTML		Plain Text	
	Provides	Accepts	Provides	Accepts
All browsers (inc. Rich Text editor)	Yes	Yes	Yes	Yes
Word	Yes	Yes	Yes	Yes
Excel	Yes	Yes	Yes	Yes
Lotus Symphony	Yes	Yes	Yes	Yes
Adobe Acrobat Reader	No	-	Yes	-
Windows Notepad	No	No	Yes	Yes

If pasting into an application like the Rich Text editor, and there is no HTML flavor available in the clipboard, the plain text flavor will be used.

Many applications that do provide HTML content to the clipboard do not actually use HTML content as their own native data format. These applications perform a conversion from their own data format to HTML first and provide that as the HTML equivalent flavor in the clipboard.

Understanding this concept will help the author of wikis content to understand the process that occurs when copying content from one application to another.

Working with the Rich Text editor

This section will focus on styles, formatting and general document structures.

Document structure and semantics

Even though many applications are able to transfer HTML content through the clipboard, they do not always produce semantically correct content. Source applications can also add additional meta-information and styling to content that makes it difficult for the Rich Text editor to understand or work with. The cleaner and more structurally correct the content is, the easier it will be to work with it in the Rich Text editor.

Cleaning Microsoft Word content (automatic)

There are many examples where content copied from Microsoft Word has incorrect semantic structures e.g. Microsoft Word might use styled paragraphs to represent lists rather than use proper HTML list semantics. Microsoft Word also adds custom additional styling information embedded in the content that is not easily handled by other applications such as the Rich Text editor.

Due to the issues with copied Microsoft Word content, a dedicated Microsoft Word content filter is invoked automatically by the Rich Text editor whenever a paste operation is detected. This filter removes a lot of the extraneous styling information and attempts to fix any semantic problems with the content, especially with lists.

This filter can also be triggered manually by selecting the “Paste Special” command from the Paste menu button. It will bring up a dialog into which the user can paste their content.

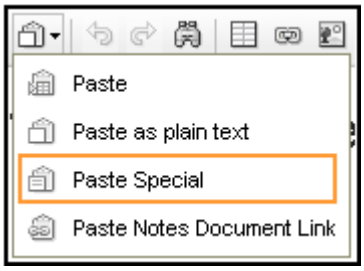


Figure 1 Paste Special

Note: Sometimes styling will be lost during the operation when the filter is activated. In some cases this is unavoidable as the filter will prefer fixing the semantics of the content over preserving strict styling information.

Paste as Plain Text

There may be cases where the content being pasted into the editor is so difficult to work with that the Paste Special command can not properly clean it. In this case it is probably better to paste the content into the editor as plain text, i.e. to force the plain text data flavor into the editor. The outcome of performing this kind of operation is that all semantic structures and styling will be removed from the content and will need to be manually re-applied. The up-side to this approach is that the document will contain all relevant content as plain text, making it much easier to work with and alter.

This is probably a last resort option, when no other technique works.

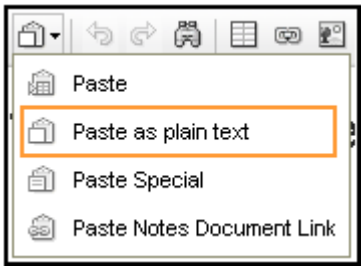


Figure 2 Paste as Plain text

NOTE: Using this feature is equivalent to first pasting into a simple text editor e.g. Notepad and then copying from Notepad into the Rich Text editor.

Remove Format

Perhaps most of the semantic information in the content is valid, but there is styling within the content that the user wants to remove. This can be achieved using the “Clear Formatting” button in the toolbar.

Figure 3 Remove Format button

There are many reasons why a user may want to use this feature to normalize their content. Sometimes source applications will apply styling to content that the Rich Text editor does not recognize or cannot directly manipulate without editing the HTML source directly. For example, a line-height may visually appear to have been applied to some paragraphs in the copied content, but the editor does not recognize these as such. In this case it is desirable to remove these line heights from the document after it has been pasted into the Rich Text editor.

In the example below, the sample content has had some line spacing applied to its paragraphs within the source application (Microsoft Word in this case):

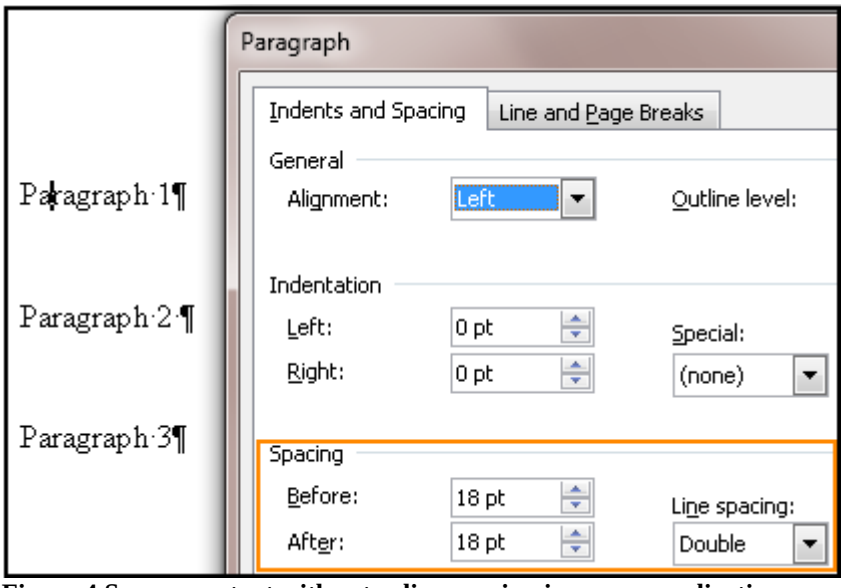


Figure 4 Source content with extra line spacing in source application

When the content is copied and pasted into the Rich Text editor the user will see the line spacing, but be unable to remove it without manually editing HTML. In order to preserve a consistent presentation with the rest of the content in the same document, the user can select the new content and then use the "Remove Format" button to remove the extra line spacing styles. This will effectively remove all other styling.

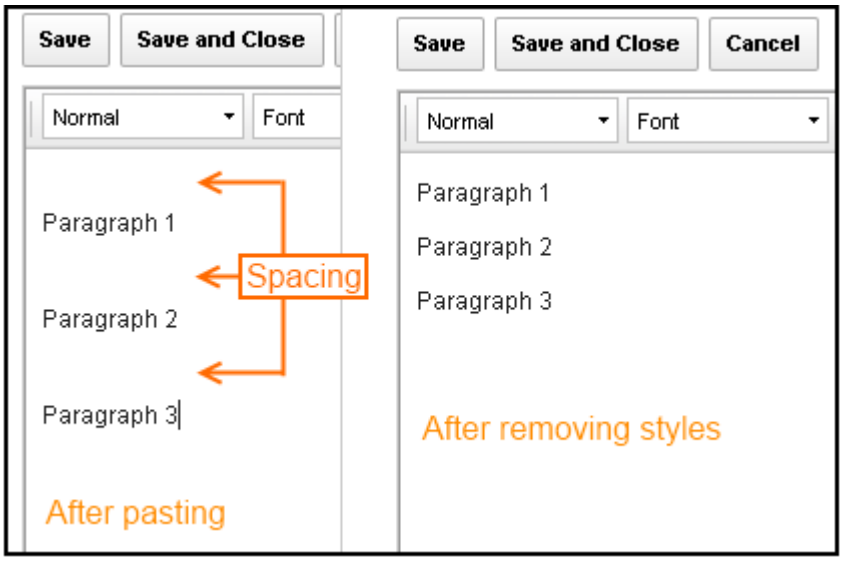
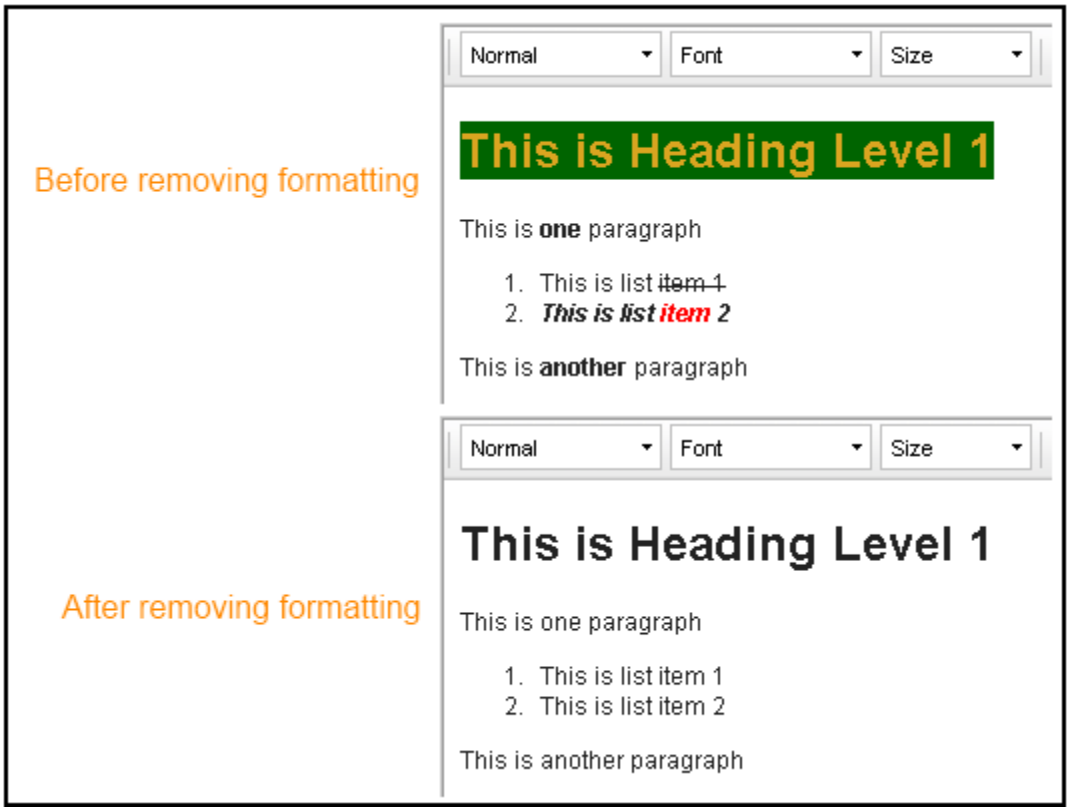


Figure 5 Result of pasting and clearing styles

Remove Format will not change the semantic structure of the document. This means that headings, lists and other structures will remain; only their styling will be removed. For this reason, this approach is better than pasting the whole content as plain text.



A note about tables: When the formatting of a table is removed using Remove Format, its width may also be reset. This is because the width of a table is controlled via styles.

Show Blocks

The Show Blocks feature in the Rich Text editor is an editing aid that does not manipulate the document. Its purpose is to highlight where the block elements in the document are. The feature can be turned on and off using the toolbar button:

Figure 6 Show blocks toolbar button

The following is the current lists of all block elements that this feature will show.

Block label	Description
address	A block for containing addresses
blockquote	A block for containing quotes
div	Similar to a Paragraph, but can contain other block elements and does not have any default margin applied in the editor.
h1, h2, h3, h4, h5, h6	Heading elements. Displays for levels 1 - 6.
p	The basic paragraph element. Paragraphs cannot contain other block elements (unlike the div). By default browsers apply a

	small padding around each paragraph.
pre	Formatted content.

These are the same elements that are also available in the Paragraph Format panel in the toolbar:

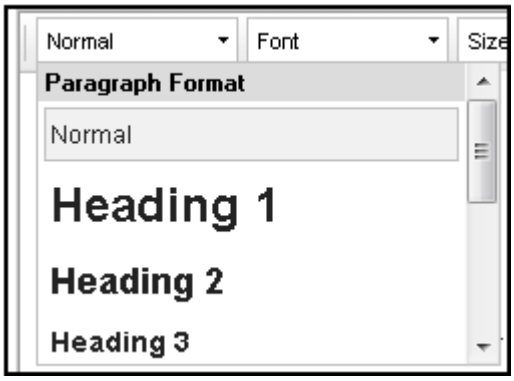


Figure 7 Paragraph foramat toolbar panel

Other block elements are not highlighted by this feature, but are usually much easier to recognize, e.g. tables and lists.

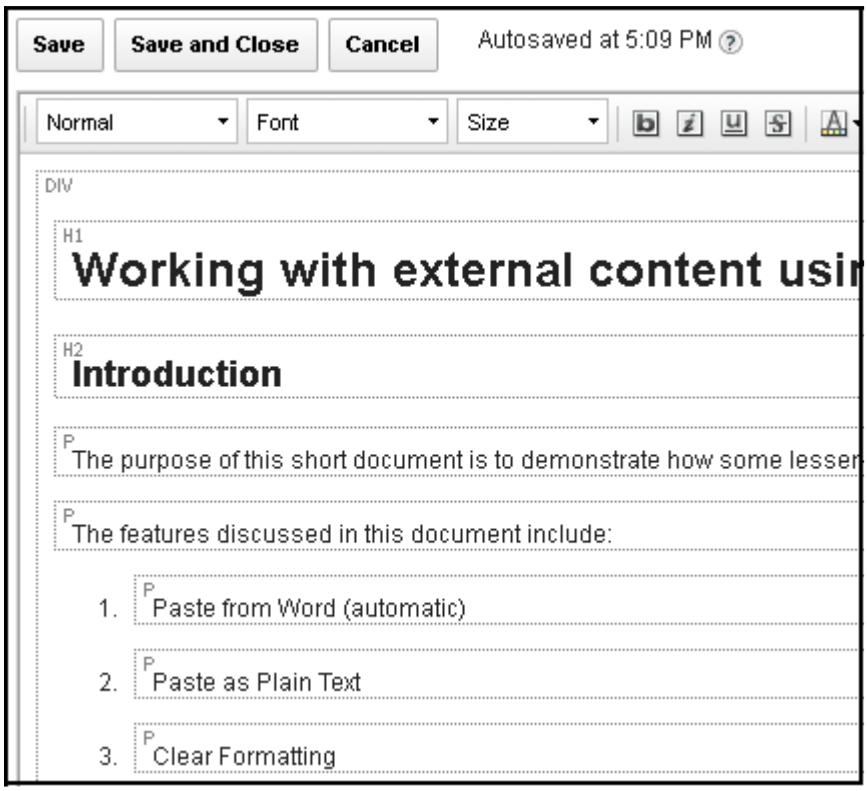


Figure 8 Show Blocks turned on

The Show Block feature helps to ensure that the content is semantically correct.

Working with HTML source

The HTML source tab in Connections Wikis can be used to edit the source of a document. This may sometimes be done by advanced users when a format or style can not be achieved through the Rich Text editor.

Figure 9 HTML Source tab activated

Using the HTML Source tab

There are a few restrictions that users should be aware of if they want to take advantage of this feature.

DTD validation and supported HTML markup

The HTML content that is accepted by Wikis and other Connections components must be valid XML.

The HTML content is not validated against any DTD and therefore any valid XML/XHTML content is accepted. It is worth noting that the wiki page content will be rendered within the context of the Connections UI just like all other content in Connections.

For reference only, IBM Connections uses the following doctype declaration:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

The following is a list of things a HTML author should be aware of; some of them are relevant to XML documents generally:

- Content is encoded in UTF-8 during save.
- All tags must have either matching open and close tags or be self-closing. For example `<BR>` is not valid, but `<BR/>` is.

- Duplicate attribute names on a single element will not be accepted e.g.
  `<a href="link.html" href="link.html">Link</a>`

- Most HTML entities are not supported and their equivalent UTF-8 characters must be used. For example, this means that the copyright symbol entity **`&copy;`** will not be accepted, but the character © will be. HTML entities that have been entered into the HTML source view can be automatically converted to UTF-8 by switching to Rich Text and then back again.

  An exception is the **` `** (non-breaking space) entity.

- The standard XML entities will be accepted e.g.
  `<` - less than (<)
  `>` - greater than (>)
  `&` - ampersand (&)

- URLs should be properly encoded e.g. the following will fail:
  `https://www.google.com/search?q=test&start=200`
  but this will work:
  `https://www.google.com/search?q=test&start=200`

- Any tag name may be used as long as it adheres to the above restrictions e.g.
  `<test>My Test tag</test>`
  is valid and accepted (although meaningless on its own).
  Users should be careful that tag names are correctly spelled.

- Some restricted content will be removed if it is deemed potentially unsafe (see below)

Fixing content by switching between HTML Source and Rich Text

Most issues can be resolved by switching from the HTML Source tab to Rich Text and then back again to HTML Source. When this action is performed, the Rich Text editor will attempt to fix content and ensure that it is valid. In some cases this will result in the content being re-arranged, especially if the HTML structure is nested in an unsupported way.

Restricted content – Active Content Filter

All content that is saved in Connections is processed by an Active Content Filter (ACF). The ACF is in place to inhibit the creation of malicious content. The user should be aware that any content that is considered to be unsafe, for example JavaScript, will be removed from the document without providing any feedback to the user that this has happened.

In some cases the content will be converted into a different structure that should be functionally equivalent but safe.