



# **InDesign EPUB Changes - CS6 To CC**

---

Douglas Waterfall  
Architect, InDesign Engineering

## General

- Table Of Contents (TOC) Stories And Content Now Supported
- Index Stories Now Supported
- Export Option "Generate CSS" restored as in CS5.5
- Restored Scripting Support for EPUB Book Export as in CS5.5
- iBooks Reader Device and Embedded Fonts Now Supported
- Turned Off ZIP Compression of Embedded Image/Audio/Video Files
- Added Version String and Build Number to generator Metadata
- New Export Warning/Error Dialog
- Removed "EPUB 3.0 With Layout" from EPUB Export Dialog
- Fixed Issue With Empty <title> Metadata
- Fixed Issue When Generating Document Title Containing Japanese Text
- Changed Handling Of Filenames With Spaces
- Removed DTDs From NCX File in EPUB 3.0 Export
- Fixed Issue With Generated CSS Filename Containing Japanese Text
- Fixed Issue With Generating NCX/NAV Entries With Numbered Paragraphs
- Fixed Crash in Export Scripting
- Fixed Crash When Exporting A Story Containing Text Converted To Outlines

## UI

- Object Export Options Now Supported By Object Styles
- Style To Tag/Class Mapping Now Supported By Object Styles
- Multiple Class Names Per Style Now Supported In Style Export Tagging
- New Control Added To Style Export Tagging To Resolve Class Name Collisions
- Removed "Static Ordered List" Text Option from Export Dialog

## HTML Markup

- Lists Greatly Improved
- Changed Custom Anchored Objects and Tables Dividing Paragraphs Into Two
- Reverted the Inserting Of Non-Breaking Spaces Into Empty Paragraphs As In CS5.5
- Changed Handling Of Empty Paragraphs
- Class Names Auto-Generated From Style Names Now Fully Qualified By Style Group
- Improved the Naming of InDesign Generated Classes To Better Describe Their Purpose
- Fixed Issue Resulting in Empty Japanese Ruby Element
- Fixed Footnote Markup Issue Involving Swapped Prefix/Suffix Control

## CSS

- Paragraph Keeps Attributes Now Mapped to Appropriate CSS Properties
- Restored `margin-top` Property On Paragraph CSS
- Removed `-epub-ruby-position` From Non-Japanese Exports
- Changed CSS Generation For Japanese Ruby Tag
- CSS Classes For Object Styles Now Containing Object Export Option Properties
- Fixed Width Calculation For Images With Effects
- Removed Unused CSS Property `-adobe-float-keep`

## General

### Table Of Contents (TOC) Stories and Content Now Supported

In CS6 any Table Of Contents Story included in the export was quietly ignored and behind the scenes a new TOC Story was generated based on the TOC Style and this is the Story which was exported to the EPUB.

Users which had not hand-edited their generated TOC, or had copied any of the paragraphs and pasted them somewhere else would not have noticed this substitution.

The reason it was done this way in CS6 was that the (hidden) hyperlinks in the TOC entries are actually hyperlinks to PDF pages, NOT hyperlinks back the originating paragraph, and so it was not possible to create links to them.

In CC this has been redone so that the hyperlinks contain BOTH a PDF Page target and a Text Anchor target which allows us flexibility during the appropriate export.

This new scheme also means that if any portions of the TOC which are copied and pasted elsewhere in the document will also be exported to EPUB correctly.

**Note:** *TOC Stories generated in CS6 and before will need to be re-generated in CC in order for the links to be live back to their source paragraphs.*

The NCX (navigation) File, required in the EPUB Package, cannot use an existing TOC and will be generated using the TOC Style as in CS6.

If you do not want Page Numbers in the TOC then the TOC must be generated without Page Numbers. We have considered providing some heuristic to automatically remove these Page Numbers on EPUB Export but that will not happen in CC.

### Indexing Stories Supported

In CS6 when a Story generated from the Index feature was included in the export it was silently ignored during export

These Stories are now supported and the links from the Page Numbers to the Index markers stay live.

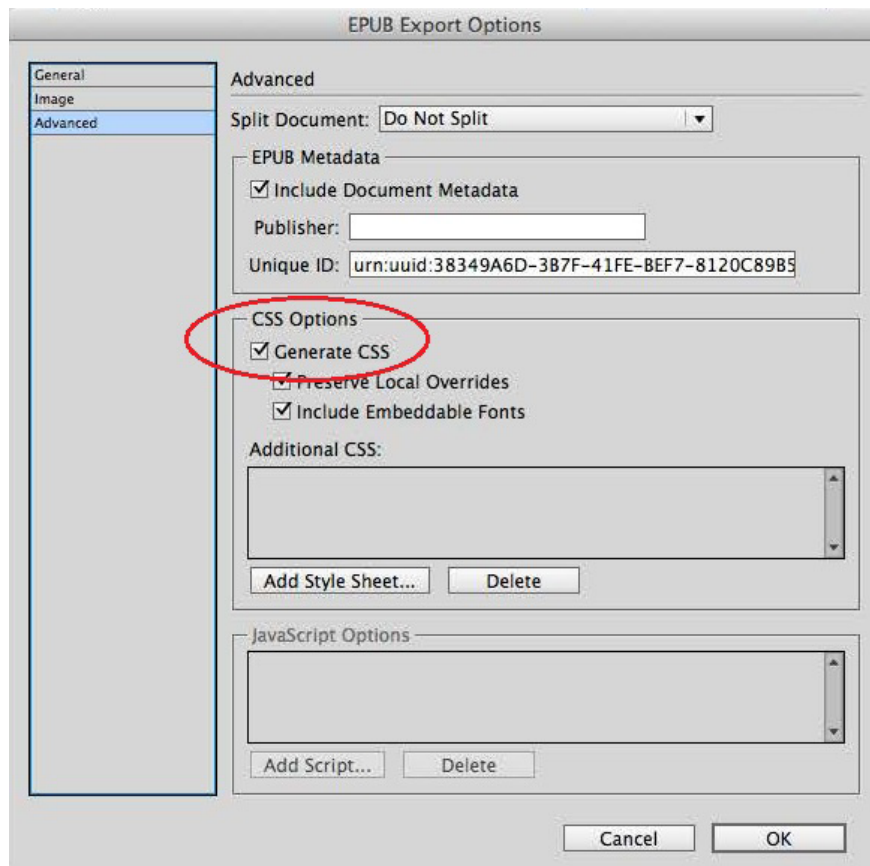
**Note:** *Index Stories generated in CS6 and before will need to be re-generated in CC in order for the links to be live.*

### Export Option "Generate CSS" restored as in CS5.5

In CS6 you could disable defining CSS for Styles but you could not prevent the creation of a CSS file with some boilerplate values.

This was a problem for users/developers who wanted to define their own CSS without any help from InDesign.

This functionality has been restored with the additional refinement that **NOT** "Generate CSS" *really* means InDesign will do nothing related to generating any CSS.



For example, there are a variety of cases where InDesign will create what we have called "generated Classes" over and above the Class markup defined by the Style in order to improve the visual fidelity of the export.

Drop caps, height and width of images, em-based point size, alternating strokes and fills for tables, etc are all cases where InDesign has to emit additional CSS over what the Styles and User-Overrides say.

When "Generate CSS" is disabled, none of these Classes will be generated and they will not be added to the HTML markup.

In addition, users will need to provide his or her own fully capable CSS file as InDesign will no longer generate one at all.

## Restored Scripting Support for EPUB Book Export as in CS5.5

This was lost CS6.

## iBooks Reader Device and Embedded Fonts Now Supported

iBooks Reader supports a wide number of device fonts (<http://iosfonts.com/>) which by default are not enabled when referenced via @font-face unless the special XML option file:

`META-INF/com.apple.ibooks.display-options.xml` is added to the package which we have not done so for EPUB 2 and EPUB 3 exports.

For embedded fonts, we have figured out why iBooks Reader was not properly reading our embedded fonts and we've made a change (obfuscate before compress) which allows this to now work on iBooks Reader while still working for RMSDK based devices.

EPUBCheck was recently changed (Nov 2012) to accept this font format as being acceptable, in

contrast with the April 2012 release which flagged it as an error.

Font embedding on the Kindle Fire continues to not work.

## Turned off zip compression of image/audio/video files

These are already compressed formats.

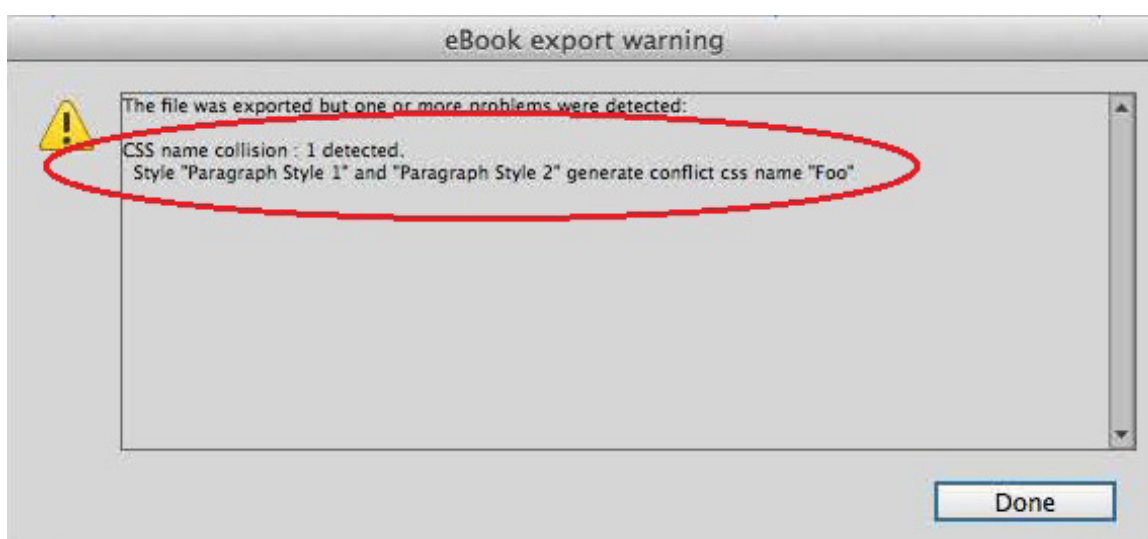
## Added Version String To generator Metadata

```
<meta name="generator" content="Adobe InDesign 9.0 Build 171" />
```

*Note: Build number will only be present in prerelease builds.*

## Added Export Warning/Error Message Dialog

This makes it easier to read multiple warning/error message from the export.



## Removed "EPUB 3.0 With Layout" from EPUB Export Dialog

The Adobe Digital Publishing Service has elected to not support this format at this time.

## Fixed Issue With Empty <title> Metadata

To pass EPUBCHECK one needs to have <title> metadata tag, even if it is empty. Users who failed to specify a Document Title in the InDesign/File/FileInfo would get warnings.

We now always emit the <title> tags, even if empty, to avoid the warning.

## Fixed Issue When Generating Document Title Containing Japanese Text

## Changed Handling Of Filenames With Spaces

Filenames with spaces were creating problems in the EPUB package.

*This change was in the in CS6 dot release.*

## Removed DTDs From NCX File in EPUB 3.0 Export

This was triggering an EPUBCheck failure.

This was legal for EPUB 2.0 export but not EPUB 3.0 export.

## Fixed Issue With Generated CSS Filename Containing Japanese Text

## Fixed Issue With Generating NCX/NAV Entries With Numbered Paragraphs

When the NAV/NCX file is generated from the TOC Style and that Paragraph Style has Numbering applied these numbers never made it to the NAV/NCX entries. This has now been resolved.

*Note: Bullets are still dropped at this time.*

## Fixed Crash in Export Scripting

Setting `epubExportPreferences.tocStyleName="Default"` caused InDesign to crash.

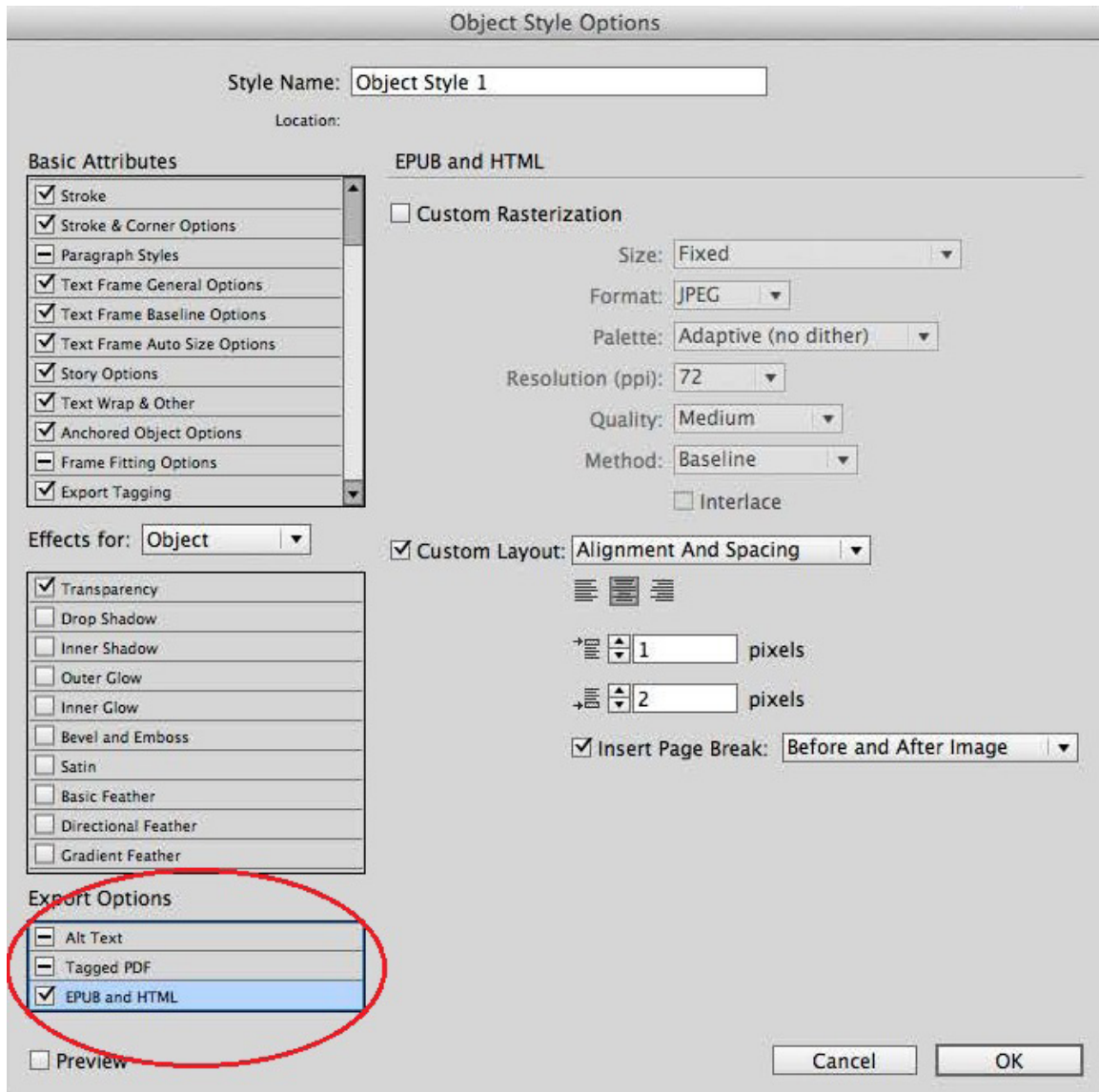
## Fixed Crash When Exporting A Story Containing Text Converted To Outlines

The outlines needed to have been converted to Inline TextFrames.

# UI

## Object Export Options Supported By Object Styles

Object Export Options were implemented in CS 5.5 to give Users control on a per-Object basis for rasterization and Custom Layout (amongst others).

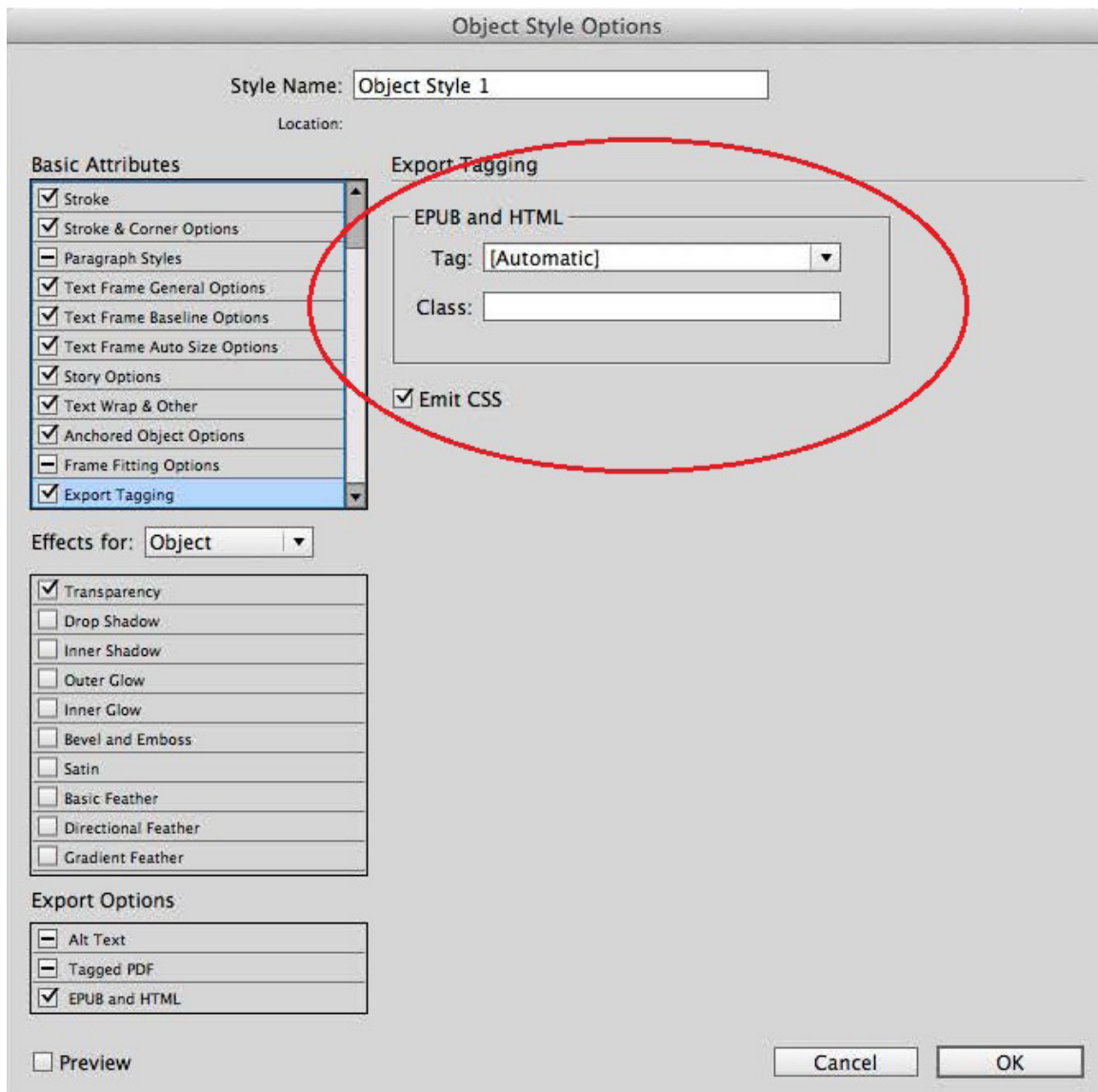


This capability has been integrated into Object Styles under Export Options at the bottom of the Object Styles dialog and will work as expected for any other attribute applied to an Object/Pageltem including IDML, scripting, overrides to the style, etc.

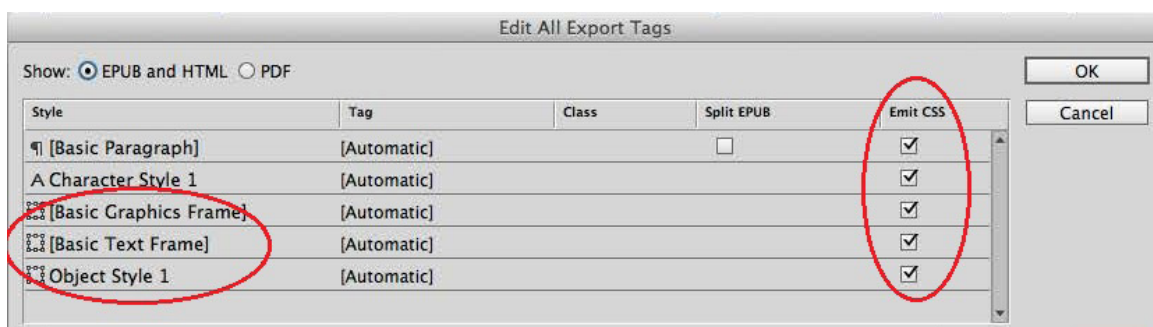
## Style To Tag/Class Mapping Supported By Object Styles

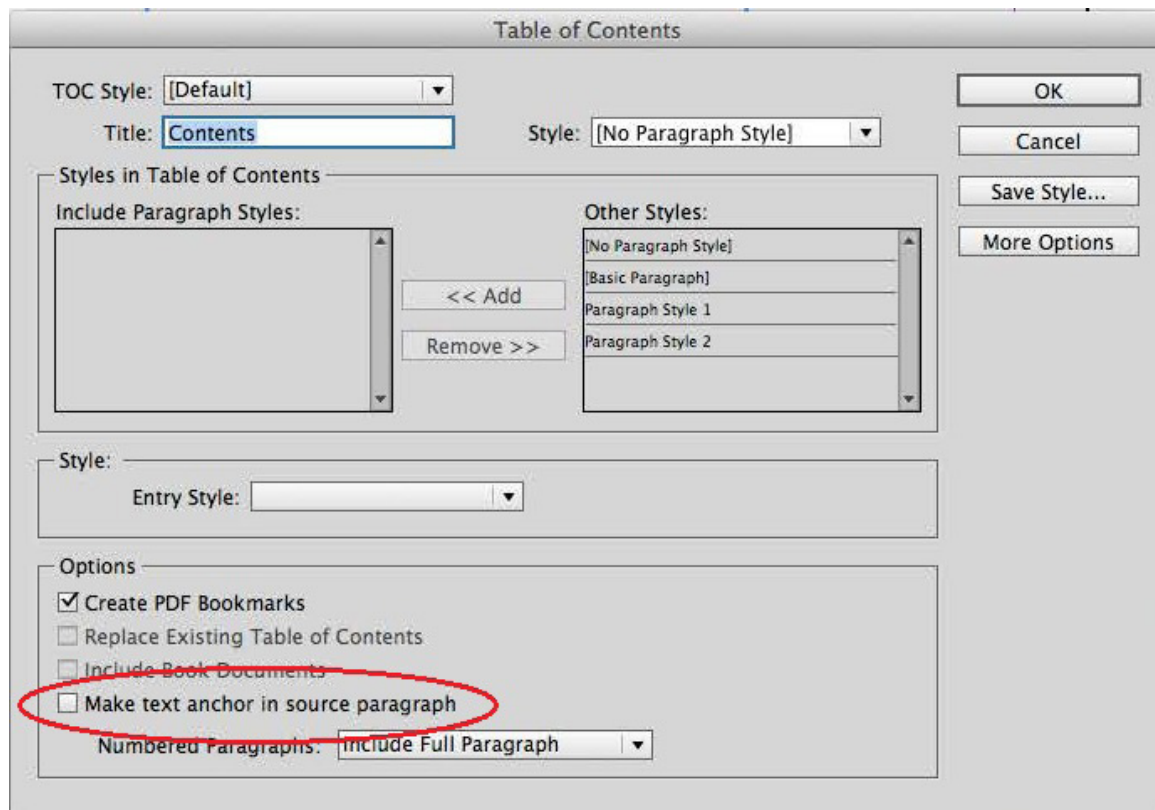
This can be found under "Export Tagging" in the Object Styles Dialog and follows the same model as with Paragraph and Character Styles.





Object Styles also now appear in the "Edit All Export Tags" dialog which is found on the Paragraph Style and Character Style Palette flyout menu.





## Added support for more than one CSS Class name in Style To Tag/Class Fields of Paragraph, Character and Object Styles

Multiple Class names must be separated by one or more spaces and the first Class name will be used to generate the CSS for Style, if Emit CSS is enabled for this Style.

For example for a Paragraph Style that says this:

Class:

Results in:

```
<p class="InDesign is the best">foo</p>
```

```
p.InDesign { ... }
```

The intention of this feature is to allow Users to override or extend CSS properties via the "Additional CSS File" mechanism in the Advanced tab of the EPUB Export Dialog.

No additional checking is done for the values specified and during export InDesign simply carries them along for the purposes of markup and otherwise ignores them.

## New Control Added To Style Export Tagging To Resolve Class Name Collisions

There are situations where a user may declare that two or more Styles should map to the same Class and although in terms of HTML markup this is an unambiguous situation, the problem is that InDesign does not know from which Style the Class should be created when it was asked to create CSS.

Paragraph/Character/Object Style Name	CSS Class Name	Which Style should InDesign use to create the CSS for MyClass?
My Style 1	MyClass	
My Style 2	MyClass	

To get around this ambiguity InDesign would silently rename one of the Classes to be a different name and then generate Class properties for both. Using the above example for mapping to 'MyClass':

Paragraph/Character/Object Style Name	CSS Class Name
My Style 1	MyClass
My Style 2	MyClass-1

In CC this policy has been changed - there is no longer any renaming and conflicts will result in a warning message on export.

**Note: If Generate CSS is off, then there is no ambiguity and no warning is raised.**

Export from Books has a slight nuance that needs explanation. On export the document declared the Style Source will define the Style to Class mapping.

If, while processing a document it comes across a Style which was not defined in the Style Source document (call it Style A), then "Style A" will be silently added as if it was part of the Style Source document.

While processing a later document, which also defines "Style A", then this document (and all later processed documents) will simply defer to this first definition of this Style and thus possibly result in a different styling.

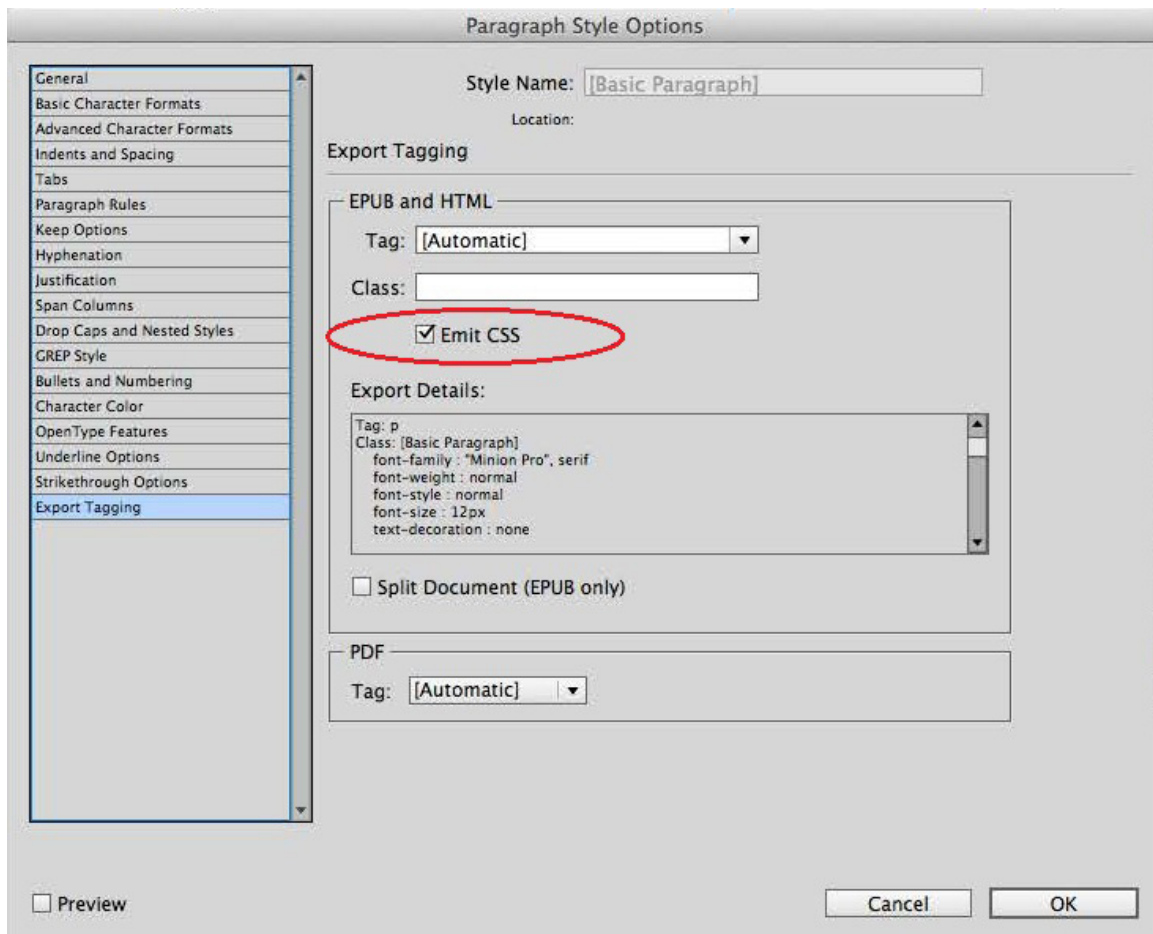
**Note: No warning will be given for these duplicate Styles which may not be the same.**

To avoid this potential issue, make sure your Style Source document contains all the Styles used in all the documents in the Book.

To still allow users to map two or more Styles to the same Class, we have introduced a new control in the Style Export Tagging to allow the user to declare which Style should be the one to define the CSS for the Class, thus removing the ambiguity.

This control has been added to the Paragraph, Character and Object Styles Edit Dialogs.

**Note that it is possible that all related Styles say they do not want to define CSS in which case no CSS will be generated for the Class. This opens up the possibility that a user-supplied CSS file could define the CSS for this Class.**



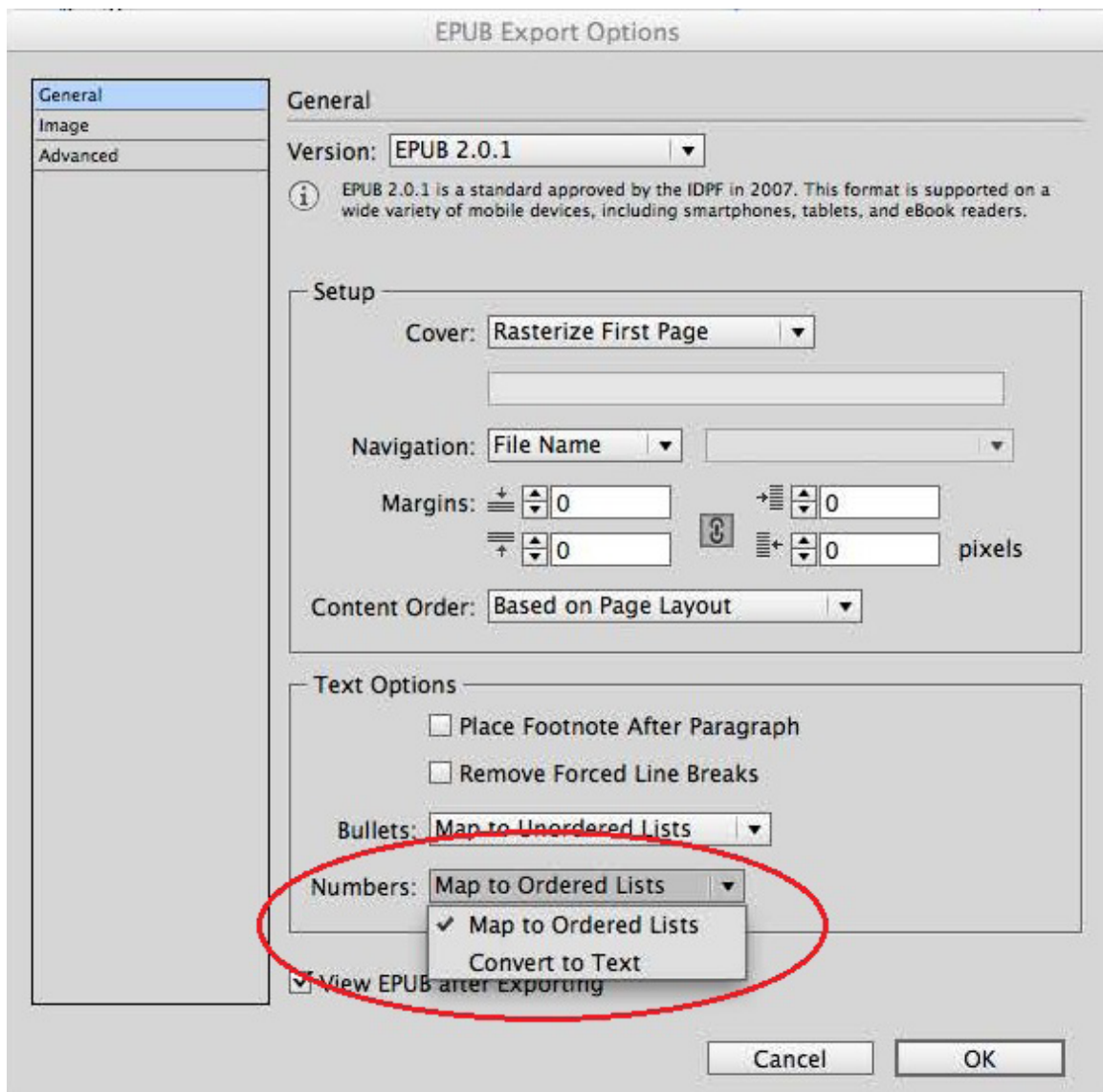
# HTML Markup

## Improved Handling Of Lists

- Removed "Static Ordered List" option from EPUB Export Dialog

This was a holdover from the pre-CS5.5 which forced each numbered list paragraph to have a specific value but still be mapped to a HTML list.

Now that we offer a true (stripped) markup to lists respecting the starting number values this functionality is no longer deemed worthwhile to maintain.



- Improved the HTML markup and CSS of the "Convert To Text" list option to more closely map the look in InDesign

With the "Convert To Text" export option bullets and numbers are marked up as a simple paragraph and no longer an HTML lists and to achieve a composed look that more closely matches what InDesign does we insert the necessary bullet/number characters and override styling.

EPUB Export Text Option "Convert To Text"	
	<ul style="list-style-type: none"> <li>Bullet</li> <li>2. Number</li> </ul>
CS6	<p>&lt;p class="Bullet"&gt;&lt;span&gt;•&amp;#9;&lt;/span&gt;Bullet&lt;/p&gt;</p> <p>&lt;p class="Number"&gt;&lt;span&gt;2.&amp;#9;&lt;/span&gt;Number&lt;/p&gt;</p>
CC	<p>&lt;p class="Bullet"&gt;&lt;span class="_idGenBNMarker-1"&gt;•&lt;/span&gt;Bullet&lt;/p&gt;</p> <p>&lt;p class="Number"&gt;&lt;span class="_idGenBNMarker-1"&gt;2.&lt;/span&gt;Number&lt;/p&gt;</p> <pre>span._idGenBNMarker-1 {     display:inline-block;     text-indent:0;     width:18px; }</pre>

*Note: As not all attributes in InDesign map well to CSS you mileage will vary.*

- Removed generated characters and CSS Classes From Ordered/Numbered Lists Exports

Under the EBook Export/General/Text Options "Map to Unordered List" and "Map to Ordered List" we have changed the generated markup to be free of any extra/added Classes.

In CS6 we would insert the bullet/number characters, and override spans and control the composition of the list so that it would appear to look more like InDesign even though the markup was an HTML list.

We have changed our policy on this behavior and now no longer insert any characters or create any spans and simply let the browser/device be responsible for composing the unordered/ordered list.

We know that InDesign has more controls with respect to lists than HTML/CSS support but that's just fine - we do the best mapping we can while leaving the semantic mapping between InDesign and HTML as pure as possible.



EPUB Export Text Option “Map to Unordered/Ordered List”	
	<ul style="list-style-type: none"> <li>• Bullet</li> </ul> <ol style="list-style-type: none"> <li>2. Number</li> </ol>
CS6	<pre> &lt;ul&gt; &lt;li class="Bullet"&gt; &lt;span class="char-style-override-1"&gt;•&amp;#9;&lt;/span&gt;Bullet&lt;/li&gt; &lt;/ul&gt; &lt;ol&gt; &lt;li class="Number"&gt;&lt;span class="char-style-override-1"&gt;2.&amp;#9;&lt;/span&gt;Number&lt;/li&gt; &lt;/ol&gt;  span.char-style-override-1 {     display:inline-block;     text-indent:0;     white-space:pre-wrap;     width:18px; } </pre>
CC	<pre> &lt;ul&gt; &lt;li class="Bullet"&gt;Bullet&lt;/li&gt; &lt;/ul&gt; &lt;ol&gt; &lt;li class="Number" value="2"&gt;Number&lt;/li&gt; &lt;/ol&gt; </pre>

- Improved heuristic for determining bullet and numbering semantic structure

CS6 used a very simply algorithm that resulted in the failure to detect both nested lists but also cases where lists were interrupted by paragraphs which were visually part of the same list element.

The prior policy around inserting bullet/numbers into list paragraphs masked this situation until some Users wanted to do their own styling at which point it became clear that anything more than a simple lists was not being represented properly.

In CC we have significantly improved this heuristic to enable detecting the semantic structure of the bullets and numbers and properly expressing this in HTML.

We can only claim this is a heuristic because InDesign does not have a formal mechanism to declare relationships between paragraphs in the same list (Numbers have some support of this) so we have to make some educated guesses in order to back into the implied structure.

For bullets, the rule is that we treat two paragraphs which share ALL the same bullet related attributes, including left indent, as being part of the same list.

In addition, a paragraph which is not a bullet, but which follows a bullet paragraphs and has the same left indent, is considered as a "nested" paragraph and becomes part of the list.

For numbers, because users can set the attribute for the "List" which it is part of, this setting has the highest priority relative to left indent or different number related attributes.

*Note: Numbered list which cross Table and Story boundaries are not possibly to express in*

*HTML, although we do support setting the value attribute on the <li> tag.*

- Improved mapping of bullet character to list-style-type property

With the change to leave composition of the list to the browser/device we must do our best to map InDesign's Format/Bullet Character attribute to the appropriate CSS property.

ID Unicode value	CSS Property
0x2022 (bullet)	disc
0x25CB (white circle)	circle
0x25A0 (black square)	square

If any Unicode value is found other than these, the CSS will NOT specify it, leaving it to the device/browser default (typically "disc").

ITextAttrNumberingStyle	CSS Property
1, 2, 3, 4	decimal
01, 02, 03,	decimal-leading-zero
I, II, III,	upper-roman
i, ii, iii,	lower-roman
A, B, C, D	upper-alpha
a, b, c, d	lower-alpha
001, 002, 003,	decimal-leading-zero
0001, 0002, 0003,	decimal-leading-zero

- Changed handling of value attribute for Ordered List <li> tags

This attribute controls the starting number of Ordered Lists (defaults to 1) but for EPUB 2.1 this attribute is **not** legal and will cause EPUBCheck to fail, even though popular viewers/devices do render this as expected.

*Note: Arguably EPUBCheck should not fail because this attribute was only marked deprecated in HTML4 and, as it happens, un-deprecated in HTML5. Regardless, we have decided that passing EPUBCheck is more important and so this attribute will NOT be exported for EPUB 2.1.*

For EPUB 3.0 exports this attribute is fully supported.

*Note: We try to be intelligent about only exporting it when the value is different then what we think the browser will calculate.*

- Improved the CSS generation for <li> tags

This is part of the large effort around lists discussed above.

With the change in the HTML markup all CSS properties which are defined as being relative to their parent (in HTML terms) needed generated override Classes to behave as they did in InDesign.

## Changed Custom Anchored Object And Tables Dividing Paragraphs In Two

CS6 would do this because it was trying to increase the visual fidelity between EPUB and InDesign Composition at the expense of semantic fidelity.

This has now been reversed in CC – we will maintain semantic markup of the Paragraph and the AnchoredObject with Custom or AboveLine settings, and Tables will be move outside of the



Paragraph markers.

## Reverted the Inserting Of Non-Breaking Spaces Into Empty Paragraphs As In CS5.5

In InDesign an empty paragraph is not really empty - there is always a carriage return which has attributes and gets composed providing vertical spacing.

In CS6, in an attempt to preserve the composition effect on export to HTML a non-breaking space was inserted into the empty paragraph so it would become non-empty.

This policy has now been reversed and we no longer insert this extra character.

*This change was in the in CS6 dot release.*

## Changed Handling Of Empty Paragraphs

Empty Paragraphs in InDesign are not really empty - they contain a carriage return and are composed with vertical height.

In HTML we map our Paragraphs to <p> and <li> tags as appropriate and the carriage return is not part of the mapping can lead to empty tags.

Likewise there are certain transformations involving Anchored Objects and Tables that result in Paragraphs which did contained content but no longer have any once in HTML.

We now look for these empty tag ranges and remove them.

## Class Names Auto-Generated From Style Names Now Fully Qualified By Style Group

When a Style has not defined a Class name InDesign will generate one from the name of the Style, replacing spaces and other illegal characters with dashes as required to create a legal name in CSS.

In CC we will now precede this generated Class name with the name of the Style Group(s) (if any) the Style is part of using dashes a separators between the Style Group(s) and the Class name.

## Improved the Naming of InDesign Generated Classes To Better Describe Their Purpose

As mentioned above, InDesign generates a variety of extra Classes for the purposes of improving the visual fidelity and conceptual mapping between InDesign and EPUB.

We have now broken these out and renamed them to better clarify their purpose.

Generated CSS Class Prefix	Description
CharOverride-#	Mapping of user applied Character Attribute override to CSS properties
ParaOverride	Mapping of user applied Paragraph Attribute override to CSS properties
TableOverride-#	Mapping of user applied Table Attribute override to CSS properties
CellOverride-#	Mapping of user applied Cell Attribute override to CSS properties

ObjectOverride-#	Mapping of user applied Object (PageItem) Attribute override to CSS properties
_idGenParaOverride-#	To deal with CSS property inheritance (such as left indent)
_idGenCharOverride-#	To deal with CSS property inheritance (such as point size)
_idGenDropcap-#	To make the dropcap look more like ID
_idGenBNMarker-#	To make the Bullet & Number characters look more like ID
_idGenRuby-#	Needed to alter the Ruby CSS
_idGenStoryDirection	To support story direction
_idGenTableRowColumn	To support alternating table strokes and fills
_idGenPageitem-#	To carry the width and height of the PageItem/Object

## Fixed Issue Resulting in Empty Japanese Ruby Element

## Fixed Footnote Markup Issue Involving Swapped Prefix/Suffix Control

The markup of footnote prefix/suffix for incorrectly swapped between the reference/marker

## Paragraph Keeps Attributes Now Mapped to Appropriate CSS Properties

In CS6 the Keeps Option "Start Paragraph" attribute is mapped to `break-before` which was a problem as it is a CSS3 property and invalid for a EPUB 2.1 export.

This has been changed for both EPUB 2.1 and EPUB 3.0 exports to map any value for this attribute other than 'Text::kStartAnywhere' to `page-break-before:always`.

*Note: Break characters remain unsupported at this time.*

The other Keeps Attributes are mapped as thus:

Paragraph Attribute	CSS Property
Keep With Previous	<code>page-break-before:avoid</code>
Keep All Lines Together	<code>orphans: 99</code>
Keep First N	<code>orphans: #</code>
Keep Last N	<code>widows: #</code>
Keep With Next N	<code>page-break-after:avoid</code>

*Note: Since the "Start Paragraph" attribute and "Keep With Previous" map to the same property, in cases of conflict the "Start Paragraph" mapping will win, as it does in InDesign.*

*However, we are simply mapping the attributes to properties without regard to trying to preserve the same or similar composition outcome in situations where <div>s are next to paragraphs which there were not before.*

*Our mapping of Keep With Next N ignores any values other than one.*

## Restored margin-top property On Paragraph CSS

Paragraph Space Before attribute is now mapped to the margin-top property, back to what it was in CS5.5.

*This change was in the in CS6 dot release.*

## Removed `-epub-ruby-position` from non-Japanese exports

This was being emitted for all CSS paragraph classes, now it is only emit for the J feature set.

*This change was in the in CS6 dot release.*

## Changed CSS Generated For Japanese Ruby Tag

We now generate a separate CSS rule containing just the ruby related attributes.

CS6	<pre>&lt;p class="Basic-Paragraph"&gt;&lt;ruby class="Ruby-1"&gt;Base&lt;rp&gt;(&lt;/rp&gt;&lt;rt class="char-style-override-1"&gt;abcd&lt;/rt&gt;&lt;rp&gt;)&lt;/rp&gt;&lt;/ruby&gt;&lt;span class="Ruby-1"&gt;&lt;/span&gt;  rt.char-style-override-1 {     color:#00aeef;     font-family:"Kozuka Mincho Pro R", serif;     font-size:1.085em;     font-style:normal;     font-weight:normal; }</pre>
CC	<pre>&lt;p class="Basic-Paragraph"&gt;&lt;ruby class="Ruby-1"&gt;Base&lt;rp&gt;(&lt;/rp&gt;&lt;rt id="_idGenRuby-1"&gt;abcd&lt;/rt&gt;&lt;rp&gt;)&lt;/rp&gt;&lt;/ruby&gt;&lt;/p&gt;  p.Basic-Paragraph rt {     -epub-ruby-position:over; } #_idGenRuby-1 {     font-size:1.085em; }</pre>

## CSS Classes For Object Styles Now Containing Object Export Option Properties

In CS6 we did not create CSS Classes for Object Styles but now that Object Export Options has been enabled in Object Styles we are now emitting these as the first properties for our Classes.

CC	<pre>div.Object-Style-1 {     margin:1px auto 2px auto;     text-align:center;     page-break-before:always;     page-break-after:always; }</pre>
----	---

*Note: We have heard the requests to support more properties in our Object CSS.*

## Fixed Width Calculation For Images With Effects

Images with effects appeared smaller to accommodate both the image and the effect in the fixed size.

## Removed Unused CSS Property `-adobe-float-keep`

Appeared on dropcaps. This was part of the EPUB 3.0 With Layout Support and should not have



appeared in other export paths.

© 2013 Adobe Systems Incorporated. All rights reserved.

Adobe, InCopy, InDesign and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries. Macintosh and Mac OS are trademarks of Apple Computer Inc., registered in the United States and other countries. Microsoft, Windows, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Intel is a registered trademark of Intel Corporation in the U.S. and other countries. All other trademarks are property of their respective owners. Updated information/additional third party code information available at <http://www.adobe.com/go/thirdparty>