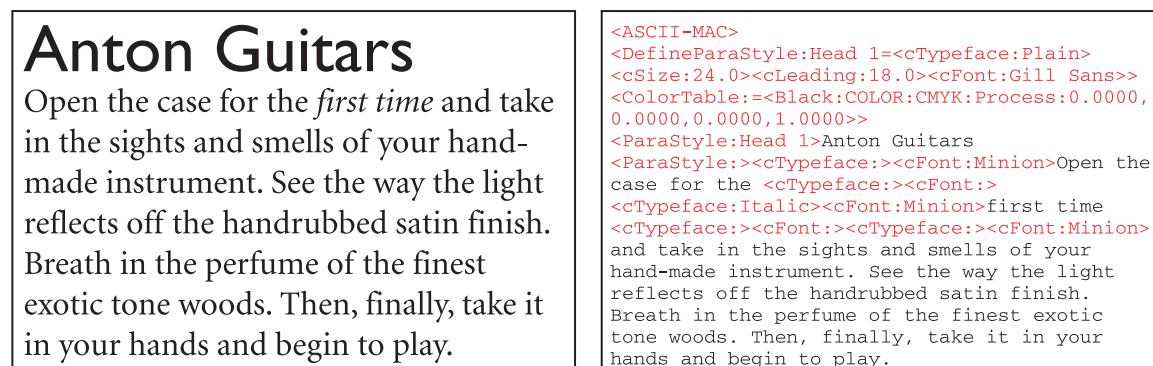# Adobe InDesign Tagged Text

You can use tags in Adobe® InDesign® to import and export formatting attributes with text-only files. These tags, or codes, indicate the character- and paragraph-level attributes you want to apply to text. For a complete list of InDesign tags, see "InDesign tags" on page 3.

When you import a tagged text file, InDesign reads the tag codes and applies the attributes automatically. When you export formatted text, you can instruct InDesign to generate the tag codes so that you can later import the information with formatting intact.

Anton Guitars

Open the case for the *first time* and take in the sights and smells of your hand-made instrument. See the way the light reflects off the handrubbed satin finish. Breath in the perfume of the finest exotic tone woods. Then, finally, take it in your hands and begin to play.

```
<ASCII-MAC>
<DefineParaStyle:Head 1=<cTypeface:Plain>
<cSize:24.0><cLeading:18.0><cFont:Gill Sans>>
<ColorTable:=<Black:COLOR:CMYK:Process:0.0000,
0.0000,0.0000,1.0000>>
<ParaStyle:Head 1>Anton Guitars
<ParaStyle:><cTypeface:><cFont:Minion>Open the
case for the <cTypeface:><cFont:>
<cTypeface:Italic><cFont:Minion>first time
<cTypeface:><cFont:><cTypeface:><cFont:Minion>
and take in the sights and smells of your
hand-made instrument. See the way the light
reflects off the handrubbed satin finish.
Breath in the perfume of the finest exotic
tone woods. Then, finally, take it in your
hands and begin to play.
```

Formatted InDesign document (left) and exported tagged text file (right), with tags shown in red

Only InDesign can read InDesign tags. InDesign cannot read tagged text files formatted in Adobe PageMaker® and QuarkXPress®.

Tagging basics

All InDesign tags begin with the less-than character (<) and end with the greater-than character (>). Type the tag immediately before the text you want to format. For paragraph-level attributes, type the tag at the start of the paragraph. For character-level attributes, type the tag at the start of the text you want to change. The type attributes you specify remain in effect until you use codes to cancel them or until you apply other attributes. For example, to apply underlining to one word in a paragraph, you type <cUnderline:1> before the word and <cUnderline:> after the word.

To learn about using tag codes, export text with tags from a formatted InDesign document. Then view the exported text in a word-processing application or text editor to see how InDesign transforms type attributes into tag codes. You can then edit and add tags before you import the text file into InDesign.

To export InDesign text as tagged text:

1 Using the Type tool, click an insertion point in the story you want to export, or select the range of text you want to export.

2 Choose File > Export.

3 For Save as Type (Windows) or Format (Mac OS), select Adobe InDesign Tagged Text.

4 Specify the name and location in which to save the file, and click Save.

5 In the Adobe InDesign Tags Export Options dialog box, specify the following options:

- Select Verbose to show the tags in long form, such as <cStrokeColor:Green> or Abbreviated to show the tags in short form, such as <csc:Green>.

- Select the encoding format, such as ASCII (the most common format for representing English characters as numbers), ANSI (a collection of many international characters and special punctuation marks that can be inserted using the Alt/Option key), Unicode (a new standard that supports most language characters, including non-European languages), and Shift-JIS (MS-Kanji for Japanese characters).

6 Click OK. The story text is exported as a text file. You can use a word-processing application or a text editor to open and edit the tagged text file.

To import tagged text:

1 Open the InDesign document into which you will place the tagged text, and choose File > Place.

2 Locate and select the text-only tagged file.

3 If desired, select Show Import Options.

4 Choose Open (Windows) or Choose (Mac OS).

5 If you selected Show Import Options in the Place dialog box, select any of the following:

- Use Typographer's Quotes to ensure that imported text includes left and right quotation marks (" ") and apostrophes (') instead of straight quotation marks (" ") and apostrophes (').

- Remove Text Formatting to remove formatting such as typeface, type color, and type style from the imported text.

- For Resolve Text Style Conflicts Using, specify which character or paragraph style is applied when there is a conflict between the style in the tagged text file and the style in your InDesign document. Select Publication Definition to use the definition that already exists for that style name in the document. Select Tagged File Definition to use the style as defined in the tagged text. This creates another instance of the style with "copy" appended to its name in the Character Style or Paragraph Style palette.

- Select Show List of Problem Tags Before Place to display a list of unrecognized tags. If a list appears, you can choose to cancel or continue the import. If you continue, the file may not look as expected.

6 Click OK. The text appears at the insertion point or replaces selected text. If there is no text selection or insertion point, a loaded text icon appears. You can click inside an existing frame to place the text inside it, or you can click or drag to create a new text frame.

InDesign tags

The following tables provide the type specifications for the InDesign tags.

Start file, color table, and style definition tags

When you export InDesign text as tagged text, the text file includes a start tag that describes the encoding format (such as <ASCII-WIN>) and a color table tag (<ColorTable>) that defines the colors, tints, and gradients in your document. If you use paragraph and character styles in your exported text, <DefineParaStyle>, <DefineCharStyle>, and <StrokeStyleTable> tags appear as well.

Note: If you are creating a tagged text file from scratch, you must include a tag that describes the encoding format (such as <ASCII-MAC>) at the beginning of the text file. You must type the following text at the top of a text file you want InCopy to treat as tagged: <ASCII-WIN> for files created in Windows or <ASCII-MAC> for files created in Mac OS. If you use an encoding format other than ASCII, use that format instead of ASCII (such as <UNICODE-MAC>).

Attribute	Format	Example and notes
Start file tag	*<Encoding format-Platform>*	<ASCII-MAC> Specify the encoding format (ASCII, ANSI, UNICODE, or SJIS), followed by the platform (MAC or WIN).
Color table	<ColorTable:=<*Swatch name 1:* *Swatch Type* (COLOR, TINT, GRAD, or MIXED): *Color Type* (CMYK, RGB, LAB, or MIXED): *Color Mode* (Process, Spot, or Mixed): *Values of color, tint, or gradient>* *<Swatch name 2>...>* A COLOR swatch requires 4 values for CMYK and 3 values for RGB and LAB. GRADIENT syntax: <*Gradient name*: GRAD:*Fill type:StopPosition1;StopColor1;* *Midpoint1	StopPosition2;StopColor2;* *Midpoint2>* TINT syntax: <*Base Color*:TINT:*Tint value>* MIXED Ink Syntax: <*MixedInk* *Name:MIXED:MIXED:Mixed:Component1Name,* *Component2Name:Component1value,Component2* *value>*	<ColorTable:= <Black:COLOR:CMYK:Process:0,0,0,1> <Green:COLOR:CMYK:Process:1,0,1,0> <Blue:COLOR:CMYK:Process:1,1,0,0> <G/B:GRAD:Linear:0.0;Green;0.5	1.0;Blue;0.5> <Blue:TINT:Blue;20>> <MixedInk1:MIXED:MIXED:Mixed:Green,Blue:1.00,0.5> The color table is a list of all colors, tints, and gradients defined in the Swatches palette and used in the text. You can use the abbreviation <Table>.
Paragraph style definition	<DefineParaStyle:*Paragraph style name*= <*attr1*><*attr2*>...> A paragraph style definition includes the character- and paragraph-level attributes listed later in this document, as well as <Nextstyle:*style name*> and <BasedOn:*style name*>	<DefineParaStyle:Heading 1=<Nextstyle:Body Text> <cFont:AdobeGaramond><cSize:18> <pBodyAlignment:Center>> The paragraph style definition includes a list of all styles defined in the Paragraph Styles palette and used in the text.
Paragraph style applied	<ParaStyle:*Paragraph style name*>	<ParaStyle:Heading 1> The tag <ParaStyle:> starts a new paragraph with no paragraph style. You can use the abbreviation <pStyle>.
Character style definition	<DefineCharStyle>*Char style name*= <*char attr1*><*char attr2*>...> A character style definition includes the character-attributes listed later in this document, as well as <Nextstyle:*style name*>.	<DefineCharStyle:Emphasis=<Nextstyle:Emphasis> <cTypeface:Italic>> The character style definition includes a list of all styles defined in the Character Styles palette and used in the text.

Attribute	Format	Example and notes
Character style applied	<CharStyle:*Character style name*>	<CharStyle:Emphasis> The tag <CharStyle:> ends the character style definition.
Stroke style table	<StrokeStyleTable:=<StrokeName1: StrokeStyleType (DASH or STRIPE): Dash/stripe values> <StrokeName2>...> Dash syntax: <*Dash name:DASH:LineCap(0,1,2): AdjustDashOrGaps:segment lengths list separated by commas*> Stripe syntax: <*Strip name:STRIPE:segmentlengths list separated by commas*>	<StrokeStyleTable:=<StrokeStyle1:DASH:0: AdjustDashes:3.810811,5.675676,11.432432,3.081081> <StrokeStyle2:STRIPE:0.130000,0.470000,0.220000,0.180000><StrokeStyle3:DASH:1:Adjust-Gaps:0.000000,6.000000,0.000000,18.000000>> Here StrokeStyle1 is a custom dash, StrokeStyle2 is a custom stripe; and StrokeStyle3 is a dotted dash.

Tags for formatting characters and paragraphs

The tables in this section list each text attribute you can set in InDesign, its corresponding tag name and abbreviation, example, and pertinent notes. Note the following tips for specifying numeric values or text within an InDesign tag:

• In the "Tag name" column for the following tables, each tag includes the value type. *String* includes a list of characters, such as the name of a color. (When the information is a text string, the case of the characters does not matter, unless you are typing a specific color or style name.) *Integer* includes only whole numbers for values, such as the number of drop-cap characters. *Real* can include any number, such as "23.578." *Boolean* includes on/off toggle values: "0" for off and "1" for on. *Enum* includes only specific string values, such as "Small Caps."

• Numeric measurements are expressed as points.

• While creating the tagged text file in a word-processing application or text editor, any characters, tabs, and spaces you type outside the angle brackets (< >) will appear in the InDesign text frame when the file is placed. To begin a new paragraph with no paragraph style assigned, type **<ParaStyle:>** on a new line.

• To return an attribute to its default state, type the attribute name followed by a colon (:). For example, to return to the default leading value, type <cLeading:>.

Character-level tags

Attribute	Tag name	Tag abbr.	Examples and notes
Alternate glyph	cAlternateGlyph:*Integer*	caltg	<cAlternateGlyph:3> Alternate glyphs for OpenType or Asian fonts such as Tekton Pro MM appear in the Glyphs palette in InDesign. In this example, 3 represents the third alernate glyph in the pop-up palette.
Auto pair kerning	cAutoPairKern:*String*	capk	<cAutoPairKern:Optical> This tag can include Optical or Metrics (default).
Baseline shift	cBaselineShift:*Real*	cbs	<cBaselineShift:3> This value can be between -5000 and 5000 points.

Attribute	Tag name	Tag abbr.	Examples and notes
Case	cCase:*Enum*	ccase	<cCase:Small Caps> Valid values include Small Caps, All Caps, Caps To Small Caps, or Normal (default). Caps To Small Caps is an OpenType font attribute.
Character alignment	cHang:*Enum*	ch	<cHang:Top> Valid values include EmTop, EmCenter (default), EmBottom, Baseline, ICFTop, and ICFBottom.
Character skew	cSkew:*Real*	csk	<cSkew:-25> The value represents the percentage of character skew from 85 to -85 degrees.
Color: color tint	cColorTint:*Real*	cct	<cColorTint:80> Specify a tint percentage.
Color: character fill	cColor:*String* or *Color Definition*	cc	<cColor:Green> or <cColor:COLOR:RGB:Process: 0.5,1,0> This tag can include tints, gradients, and unnamed colors. The default color is Black.
Color: character stroke	cStrokeColor:*String* or *Color Definition*	csc	<cStrokeColor:Green> or <cColor:COLOR:RGB: Process:0.5,1,0> This tag can include tints, gradients, and unnamed colors. The default color is None.
Color: fill gradient angle	cGradientAngle:*Real*	cga	<cGradientAngle:-45> Angle of linear gradient fill from -180 to 180 degrees.
Color: fill gradient center	cGradientCenter:*Real,Real*	cgc	<cGradientCenter:-145,-145> This tag indicates the x and y values of the center point of a radial gradient or the starting point of a linear gradient.
Color: fill gradient length	cGradientLength:*Real*	cgl	<cGradientLength:2> This tag indicates the length of a linear gradient ramp or radius of a radial gradient.
Color: stroke gradient angle	cStrokeGradientAngle:*Real*	csga	<cStrokeGradientAngle:-45> Angle of linear gradient fill from -180 to 180 degrees.
Color: stroke gradient center	cStrokeGradient-Center:*Real,Real*	csgs	<cStrokeGradientCenter:-145,-145> This tag indicates the x and y values of the center point of a radial gradient or the starting point of a linear gradient.
Color: stroke gradient length	cStrokeGradientLength:*Real*	csgl	<cStrokeGradientLength:2> This tag indicates the length of a linear gradient ramp or radius of a radial gradient.
Color: stroke tint	cStrokeTint:*Real*	cst	<cStrokeTint:80> Specify a tint percentage.
Color: stroke weight (thickness)	cStrokeWeight:*Real*	csw	<cStrokeWeight:0.5> Specify a value between 0 and 800.

Attribute	Tag name	Tag abbr.	Examples and notes
Font: family	cFont:*String*	cf	<cFont:Adobe Garamond> Specify only the family name. To complete the font information, also use <cSize> and <cTypeface>.
Font: point size	cSize:*Real*	cs	<cSize:18> Specify a point size between 0.1 and 1296.
Font: typeface	cTypeface:*String*	ct	<cTypeface:Italic> The values for this tag depend on which font is used. For example, Times New Roman uses Regular while Helvetica uses Plain.
ILG shift	cILGShift:*Real*	cilgs	<cILGShift:200> Specify a value between -8640 and 8640.
Kerning	cKerning:*Real*	ck	<cKerning:50>
Language	cLanguage:*String*	clang	<cLanguage:Portuguese: Brazilian> To use a valid string, type the name of the language as it appears in the Character palette.
Leading	cLeading:*Real*	cl	<cLeading:18> This tag lets you specify a leading value between 0 and 5000. To use default auto leading, leave value empty (<cLeading:>).
Ligatures	cLigatures:*Boolean*	clig	<cLigatures:1> 1=on (default), 0=off
No break	cNoBreak:*Boolean*	cnb	<cNoBreak:1> 1=on, 0=off (default)
Next break character	cNextXChars:*Enum*	cnxs	<cNextXChars:Column> Valid values include NoForcedBreak (default), ColumnBreak, PageBreak, BoxBreak, OddPageBreak, and EvenPageBreak.
Old style figures	cOldStyleFigures:*Boolean*	cosf	<cOldStyleFigures:1> 1=on, 0=off (default)
OpenType: figure style	cFigureStyle:*Enum*	cfs	<cFigureStyle:Proportional Oldstyle> Valid values include Tabular Lining (default), Proportional Oldstyle, Proportional Lining, Figure Style, and Tabular Oldstyle.
OpenType: contextual alternate	cOTFContAlt:*Boolean*	cotfsalt	<cOTFContAlt:0> 1=on (default), 0=off
OpenType: discretionary ligatures	cOTFDiscLig:*Boolean*	cotfdl	<cOTFDiscLig:1> 1=on, 0=off (default)
OpenType: feature list	cOTFeatureList:*String*	cotfl	<cOTFeatureList:zero\,1>
OpenType: fractions	cOTFFractions:*Boolean*	cotff	<cOTFFractions:1> 1=on, 0=off (default)

Attribute	Tag name	Tag abbr.	Examples and notes
OpenType: position	cPosition:*Enum*	cp	<cPosition:Numerator> Valid values include None (default), Superior, Inferior, Numerator, and Denominator.
OpenType: ordinals	cOTFOrdinal:*Boolean*	cotfo	<cOTFOrdinal:1> 1=on, 0=off (default)
OpenType: titling alternates	cOTFTitlAlt:*Boolean*	cotfttlt	<cOTFTitlAlt:1> 1=on, 0=off (default)
OpenType: swash alternates	cOTFSwaAlt:*Boolean*	cotfsalt	<cOTFSwaAlt:1> 1=on, 0=off (default)
Overprint character fill	cOverprint:*Boolean*	co	<cOverprint:1> 1=on, 0=off (default)
Overprint character stroke	cStrokeOverprint:*Boolean*	cso	<cStrokeOverprint:1> 1=on, 0=off (default)
Prev/Next page number	cPageNumType:*Enum*	cpnt	<cPageNumType:Next> Valid values include Next, Previous, and Current. Current is the default value.
Rotate character	cCharRotate:*Real*	cchr	<cCharRotate:45> Specify a value between -360 and 360 to rotate the character on its center axis. Positive values rotate the character counter-clockwise.
Scaling: horizontal	cHorizontalScale:*Real*	chs	<cHorizontal Scale:2> The value 1.0 equals 100%. You can use values between 1% (.01) and 1000% (10.0).
Scaling: vertical	cVertical Scale:*Real*	cvs	<cVerticalScale:2> The value 1.0 equals 100%. You can use values between 1% (.01) and 1000% (10.0).
Small cap percentage	cSmallCapPercentage:*Real*	cscp	<cSmallCapPercentage:70> Specify a percentage value.
Strikethrough	cStrikethru:*Boolean*	cstrike	<cStrikethru:1> 1=on, 0=off (default)
Strikethrough gap color	cStrikethroughGapColor:*Real*	cugc	<cStrikethroughGapColor:C\=100 M\=0 Y\=0 K\=0>
Strikethrough gap overprint	cStrikethroughGap Overprint:*Boolean*	cugo	<cStrikethroughGapOverprint:1> 1=on, 0=off (default)
Strikethrough gap tint	cStrikethroughGapTint:*Real*	cugt	<cStrikethroughGapTint:30>
Strikethrough offset	cStrikethroughOffset:*Real*	cuoff	<cStrikethroughOffset:3>
Strikethrough weight offset	cStrikethroughWeight Offset:*Real*	cuwoff	<cStrikethroughWeightOffset:3>
Strikethrough type	cStrikethroughType:*Definition*	cutype	<cStrikethroughType:ThickThin>

Attribute	Tag name	Tag abbr.	Examples and notes
Superscript or subscript character position	cPosition:*Enum*	cp	<cPosition:Superscript> Valid values include Superscript, Subscript, or Normal (default).
Tracking	cTracking:*Real*	ct	<cTracking:50> Specify a value between -1000.0 and 10000 (1000ths of an em).
Underline	cUnderline:*Boolean*	cu	<cUnderline:1> 1=on, 0=off (default)
Underline gap color	cUnderlineGapColor:*Real*	cugc	<cUnderlineGapColor:C\=100 M\=0 Y\=0 K\=0>
Underline gap overprint	cUnderlineGap Overprint:*Boolean*	cugo	<cUnderlineGapOverprint:1> 1=on, 0=off (default)
Underline gap tint	cUnderlineGapTint:*Real*	cugt	<cUnderlineGapTint:30>
Underline offset	cUnderlineOffset:*Real*	cuoff	<cUnderlineOffset:3>
Underline weight offset	cUnderlineWeightOffset:*Real*	cuwoff	<cUnderlineWeightOffset:3>
Underline type	cUnderlineType:*Definition*	cutype	<cUnderlineType:ThickThin>
Multi-master optical size	cMultiMasterOpticalSize:*Real*	cmmos	<cMultiMasterOpticalSize:10>

Paragraph-level tags

Attribute	Tag name	Tag abbr.	Examples and notes
Align to baseline grid	pAlignToGrid:*Boolean*	patg	<pAlignToGrid:1> 1=on, 0=off (default)
Alignment oftext in para-graph	pTextAlignment:*Enum*	pta	<pTextAlignment:Center> Valid values include Left, Right, Center, and Justify. Left is the default value.
Alignment of last line in paragraph	pLastLineAlignment:*Enum*	plla	<pLastLineAlignment:Center> Valid values include Left, Right, Center, and Justify. Left is the default value.
Alignment of paragraph body (not including the last line)	pBodyAlignment:*Enum*	pba	<pBodyAlignment:Center> Valid values include Left, Right, Center, and Justify. Left is the default value.
Balance ragged lines	pBalanceLines:*Boolean*	pbl	<pBalanceLines:1> 1=on, 0=off (default)
Drop cap: number of characters	pDropCapCharacters:*Integer*	pdcc	<pDropCapCharacters:1>
Drop cap: number of lines	pDropCapLines:*Integer*	pdcl	<pDropCapLines:3>
First line indent	pFirstLineIndent:*Real*	pfli	<pFirstLineIndent:6>
Glyph scaling: desired	pDesiredGlyphScale:*Real*	pdgs	<pDesiredGlyphScale:1.0> Specify a value between -0.5 (-50%) and 2.0 (200%). The default value is 1.0 (100%).
Glyph scaling: maximum	pMaxGlyphScale:*Real*	pmaxgs	<pMaxGlyphScale:1.2> The value 1.0 equals 100%. Specify a value between -0.5 (-50%) and the desired glyph scaling value. The default value is 0.
Glyph scaling: minimum	pMinGlyphScale:*Real*	pmings	<pMinGlyphScale:0.9> Specify a value between 50% (0.5) and the desired glyph scaling value. The default value is 1.0 (100%).
Grid Alignment: first line	pGridFirstLine:*Boolean*	pgfl	<pGridFirstLine:1> This value determins whether only the first line in the paragraph is aligned to the grid. 1=on, 0=off (default)
Grid Alignment: location	pGridAlign:*Enum*	pga	<pGridAlign:EmCenter> This value sets what metric in the line is aligned to the corresponding metric in the grid. Valid values includes Baseline, EmTop, EmCenter, EmBottom, ICFTop, ICFBottom, and None.
Hyphenate all capital words	pHyphenateCapitals:*Boolean*	phc	<pHyphenateCapitals:1> 1=on, 0=off (default)
Hyphenation: hyphen limit	pHyphenation LadderLimit:*Integer*	phll	<pHyphenationLadderLimit:5> Value indicates maximum number of consecutive hyphens from 1 to 25; zero means unlimited.

Attribute	Tag name	Tag abbr.	Examples and notes
Hyphenation: minimum number of characters after hyphen	pMinCharAfterHyphen:*Integer*	pmcah	\<pMinCharAfterHyphen:3\> Specify a value from 1 to 15. The value 3 is the default.
Hyphenation: minimum number of characters before hyphen	pMinCharBeforeHyphen:*Integer*	pmcbh	\<pMinCharBeforeHyphen:3\> Specify a value from 1 to 15. The value 3 is the default.
Hyphenation: number of characters in shortest word allowed to be hyphenated	pShortestWord Hyphenated:*Integer*	pswh	\<pShortestWordHyphenated:3\> This range of this value is from 3 to 25. The default value is 7.
Hyphenation: on or off	pHyphenation:*Boolean*	ph	\<pHyphenation:1\> 1=on (default), 0=off
Hyphenation: zone	pHyphenationZone:*Real*	phz	\<pHyphenationZone:48\> Value (in points) is valid only if Single-line Composer is turned on. Specify a value between 0 and 8640. The default value is 36 points.
Keep first n lines together (widow control)	pKeepFirstNLines:*Integer*	pkfnl	\<pKeepFirstNLines:4\> Specify a value from 1 to 50 (1 is equivalent to off).
Keep last lines together (orphan control)	pKeepLastNLines:*Integer*	pknl	\<pKeepLastNLines:2\> Specify a value from 1 to 50 (1 is equivalent to off).
Keep lines together	pKeepLines:*Boolean*	pkl	\<pKeepLines:1\> 1=on, 0=off (default)
Keep whole paragraph together	pKeepParaTogether:*Boolean*	pkpt	\<pKeepParaTogether:1\> 1=on, 0=off (default)
Keep with next lines	pKeepWithNext:*Integer*	pkwn	\<pKeepWithNext:4\> Specify a value from 0 to 5 (0 means off).
Leading: autoleading percentage	pAutoLeadPercent:*Real*	calp	\<pAutoLeadPercent:1.2\> The value 1.0 equals 100%. Specify a value between 0 and 500%. The default value is 1.2 (120%).
Left indent	pLeftIndent:*Real*	pli	\<pLeftIndent:6\>
Letterspacing: desired	pDesiredLetterspace:*Real*	pdl	\<pDesiredLetterspace:1.0\> Specify a value between -1.0 (-100%) and 50 (500%). The default value is 1.0 (100%).
Letterspacing: maximum	pMaxletterspace:*Real*	pmaxl	\<pMaxletterspace:1.2\> The value 1.0 equals 100%. Specify a value between the desired letterspacing value and 50 (500%). The default value is 0.
Letterspacing: minimum	pMinLetterspace:*Real*	pminl	\<pMinLetterspace:0.9\> The value 1.0 equals 100%. Specify a value between -1 (-100%) and the desired letterspacing value. The default value is 0%.

Attribute	Tag name	Tag abbr.	Examples and notes
Nested styles	pNestedStyles:*Enum*	pnstyles	<pNestedStyles:Character Style 1\,Words\,1\1;Character Style 2\,Forced Line Break\,1\,0\;>
Page or columns break before paragraph	pBreakBefore:*Enum*	pbb	<pBreakBefore:Column Break> This tag can include Page, Column, or None.
Paragraph rule above color	pRuleAboveColor:*String*	prac	<pRuleAboveColor:Black> This value indicates the color for the rule (line) above the paragraph. This string value must be a color name (not an unnamed color definition).
Paragraph rule above gap color	pRuleAboveGapColor:*String*	pragc	<pRuleAboveGapColor:Blue>
Paragraph rule above tint	pRuleAboveTint:*Real*	prat	<pRuleAboveTint:80> This value indicates the tint for the rule (line) above the paragraph. The value must be between 0 and 100.
Paragraph rule above gap tint	pRuleAboveGapTint:*Real*	pragt	<pRuleAboveGapTint:80>
Paragraph rule above gap overprint	pRuleAboveGap Overprint:*Boolean*	prago	<pRuleAboveGapOverprint:1> 1=on, 0=off (default)
Paragraph rule above stroke type	pRuleAboveStrokeType:*Enum*	prastype	<pRuleAboveStrokeType:ThickThin>
Paragraph rule above is turned on	pRuleAboveOn:*Boolean*	praon	<pRuleAboveOn:1> 1=on, 0=off (default)
Paragraph rule above left indent	pRuleAboveLeftIndent:*Real*	prali	<pRuleAboveLeftIndent:3> This value indicates the left indent from the frame edge or text edge, depending on the paragraph rule mode.
Paragraph rule above mode	pRuleAboveMode:*Enum*	pram	<pRuleAboveMode:Text> This value indicates whether the paragraph rule starts at the width of the column (frame) or width of the first line of text in the paragraph. Valid values include Text and Column. Column is the default.
Paragraph rule above offset	pRuleAboveOffset:*Real*	prao	<pRuleAboveOffset:3> This value indicates the offset from the baseline to the paragraph rule above.
Paragraph rule above right indent	pRuleAboveRightIndent: *Real*	prari	<pRuleAboveRightIndent:3> This value indicates the right indent from the frame edge or text edge, depending on paragraph rule mode.
Paragraph rule above stroke	pRuleAboveStroke:*Real*	pras	<pRuleAboveStroke:2> This value indicates the line weight for the paragraph rule above the paragraph. Specify a value between 0 and 1000.

Attribute	Tag name	Tag abbr.	Examples and notes
Paragraph rule below color	pRuleBelowColor:*String*	prbc	<pRuleBelowColor:Black> This value indicates the color for the rule (line) below the paragraph. This string value must be a color name (not an unnamed color definition).
Paragraph rule below gap color	pRuleBelowGapColor:*String*	prbgc	<pRuleBelowGapColor:Blue>
Paragraph rule below tint	pRuleAboveTint:*Real*	prbt	<pRuleAboveTint:80> This value indicates the tint for the rule (line) below the paragraph. The value must be between 0 and 100.
Paragraph rule below gap tint	pRuleBelowGapTint:*Real*	prbgt	<pRuleBelowGapTint:80>
Paragraph rule below gap overprint	pRuleBelowGap Overprint:*Boolean*	prbgo	<pRuleBelowGapOverprint:1> 1=on, 0=off (default)
Paragraph rule below stroke type	pRuleBelowStrokeType:*Enum*	prbstype	<pRuleBelowStrokeType:ThickThin>
Paragraph rule below is turned on	pRuleBelowOn:*Boolean*	prbon	<pRuleBelowOn:1> 1=on, 0=off (default)
Paragraph rule below left indent	pRuleBelowLeftIndent:*Real*	prbli	<pRuleBelowLeftIndent:3> This value indicates the left indent from the frame edge or text edge, depending on the paragraph rule mode.
Paragraph rule below mode	pRuleBelowMode:*Enum*	prbm	<pRuleBelowMode:Text> This value indicates whether the paragraph rule starts at the width of the column (frame) or width of the first line of text in the paragraph. Valid values include Text and Column. Column is the default.
Paragraph rule below offset	pRuleBelowOffset:*Real*	prbo	<pRuleBelowOffset:3> This value indicates the offset from the baseline to the paragraph rule below.
Paragraph rule below right indent	pRuleBelowRightIndent:*Real*	prbri	<pRuleBelowRightIndent:3> This value indicates the right indent from the frame edge or text edge, depending on paragraph rule mode.
Paragraph rule below stroke	pRuleBelowStroke:*Real*	prbs	<pRuleBelowStroke:2> This value indicates the line weight for the paragraph rule below the paragraph.
Right indent	pRightIndent:*Real*	pri	<pRightIndent:6>
Rotate roman	pRotateRoman:*Boolean*	prr	<pRotateRoman:1> This value determines whether all half-width text in the paragraph is rotated counter-clockwise to stand upright in vertical text. 1=on, 0=off (default)
Space after paragraph	pSpaceAfter:*Real*	psa	<pSpaceAfter:4>
Space before paragraph	pSpaceBefore:*Real*	psb	<pSpaceBefore:4>

Attribute	Tag name	Tag abbr.	Examples and notes
Tab ruler settings	pTabRuler::*x value,tab type,align on,default,leader*;	ptr	<pTabRuler:48,Center,,0,;96,Char,x,0,*;144,Left,,0,;> A semicolon (;) indicates the start of a new tab setting. The *x value* determines the distance from the left edge of the text frame or inset. *Tab type* includes Left, Center, Right, and Char (also called a decimal tab). *Align on* determines the character on which a decimal tab (Char) will align. *Default* is a boolean value (0 or 1) that indicates whether the tab is a default tab setting. *Leader* is a character or string of characters (such as periods or hyphens) you want to use as a tab leader.
Text composition engine	pTextComposer:*String*	ptc	<pTextComposer:Adobe Single-Line Composer> Specify any composer available, including the Paragraph Composer (default) and the Single-line Composer.
Word spacing: align single word	pSingleWordAlignment:*Enum*	pswa	<pSingleWordAlignment:Left>
Word spacing: desired	pDesiredWordSpace:*Real*	pdws	<pDesiredWordSpace:1.2> Specify a value between 0 and 10 (1000%). The default value is 1.0 (100%).
Word spacing: maximum	pMaxWordSpace:*Real*	pmaws	<pMaxWordSpace:1.2> The value 1.0 equals 100%. The default value is 133% (1.3). Specify a value between 0 and 10 (1000%) not less than Desired value.
Word spacing: minimum	pMinWordSpace:*Real*	pmiws	<pMinWordSpace:0.9> The value 1.0 equals 100%. The default value is 0.8 (80%). Specify a value between 0 and 10 (1000%) not greater than Desired value.

Table tags

Attribute	Tag name	Tag abbr.	Examples and notes
Table start	TableStart:*Real,Real, Real, Real*	tStart	<tStart:5,4, 1, 0> <tStart: *rows, columns, header rows, footer rows*>
Table end	TableEnd	tEnd	<tEnd>
Row start	RowStart	rStart	<rStart>
Row end	RowEnd	rEnd	<rEnd>
Cell start	CellStart:*Real,Real*	cStart	<cStart:1,1>
Cell end	CellEnd	cEnd	<cEnd>
Column start	ColStart	coStart	<coStart>
Start headers	tStartHeadersOn:*Real*	tshon	<tStartHeadersOn:2> Possible values are 0, 1, 2 0 = Repeat header on every text column 1 = Repeat header once per frame 2 = Repeat header once per page
Start footers	tStartFootersOn:*Real*	tsfon	<StartFoodersOn:1> Possible values are 0, 1, 2 0 = Repeat footer on every text column 1 = Repeat footer once per frame 2 = Repeat footer once per page
Skip first header	tSkipFirstHeader:*Boolean*	tsfh	<tSkipFirstHeader:1> 1=on, 0=off (default)
Skip first footer	tSkipFirstFooter:*Boolean*	tsff	<tSkipFirstFooter:1> 1=on, 0=off (default)
Cell: bottom inset	tCellAttrBottomInset:*Real*	tcabi	<tCellAttrBottomInset:9>
Cell: bottom stroke color	tCellBottomStrokeColor:*Definition*	tcbsc	<tCellBottomStrokeColor:Blue>
Cell: bottom stroke tint	tCellAttr BottomStrokeTint:*Real*	tcabst	<tCellAttrBottomStrokeTint:70>
Cell: bottom stroke gap tint	tCellBottomStroke GapTint:*Real*	tcbsgt	<<tCellBottomStrokeGapTint:95>
Cell: bottom stroke type	tcBottomStrokeType:*Enum*	tcbst	<tcBottomStrokeType:ThinThick>
Cell: bottom stroke gap color	tCellbottomStroke GapColor:*Real*	tcbsgc	<tCellBottomStrokeGapColor:C\=100 M\=0 Y\=0 K\=0>
Cell: bottom stroke gap overprint	tCellBottomStrokeGap Overprint:*Boolean*	tcbsgo	<tCellBottomStrokeGapOverprint:0> 1=on, 0=off (default)
Cell: default cell type	tCellDefaultCellType:*Enum*	tcdct	<tCellDefaultCellType:1>
Cell: fill color	tCellFillColor:*Definition*	tcfc	<tCellFillColor:Blue>
Cell: fill tint	tCellAttrFillTint:*Real*	tcaft	<tCellAttrFillTint:20>

Attribute	Tag name	Tag abbr.	Examples and notes
Cell: first line offset	tTextCellFirstLineOffset:*Enum*	tcflo	<tTextCellFirstLineOffset:2> Valid values include 1 (Ascent), 2 (Cap Height), 3 (Leading), 4 (x Height), and 5 (Fixed). 1 is the default.
Cell: left inset	tCellAttrLeftInset:*Real*	tcali	<tCellAttrLeftInset:9>
Cell: left stroke color	tCellLeftStrokeColor: *Definition*	tclsc	<tCellLeftStrokeColor:Blue>
Cell: left stroke tint	tCellAttrLeftStrokeTint:*Real*	tcalst	<tCellAttrLeftStrokeTint:70>
Cell: left stroke type	tcLeftStrokeType:*Enum*	tclst	<tcLeftStrokeType:ThinThick>
Cell: left stroke gap tint	tCellleftStrokeGapTint:*Real*	tclsgt	<tCellLeftStrokeGapTint:95>
Cell: left stroke gap color	tCellLeftStrokeGapColor:*Real*	tclsgc	<tCellLeftStrokeGapColor:C\=100 M\=0 Y\=0 K\=0>
Cell: left stroke gap overprint	tCellLeftStrokeGap Overprint:*Boolean*	tclsgo	<tCellLeftStrokeGapOverprint:0> 1=on, 0=off (default)
Cell: outer bottom stroke color	tCellOuterBottom StrokeColor:*Definition*	tcobsc	<tCellOuterBottomStrokeColor:Blue>
Cell: outer bottom stroke type	tOuterBottom StrokeType:*Enum*	tobst	<tOuterBottomStrokeType:ThinThick>
Cell: outer left stroke color	tCellOuterLeft StrokeColor:*Definition*	tcolsc	<tCellOuterLeftStrokeColor:Blue>
Cell: outer left stroke type	tOuterLeftStrokeType:*Enum*	tolst	<tOuterLeftStrokeType:ThinThick>
Cell: outer right stroke color	tCellOuterRight StrokeColor:*Definition*	tcorsc	<tCellOuterRightStrokeColor:Blue>
Cell: outer right stroke type	tOuterRightStrokeType:*Enum*	torst	<tOuterRightStrokeType:ThinThick>
Cell: outer top stroke color	tCellOuterTop StrokeColor:*Definition*	tcotsc	<tCellOuterTopStrokeColor:Blue>
Cell: outer top stroke type	tOuterTopStrokeType:*Enum*	totst	<tOuterTopStrokeType:ThinThick>
Cell: right inset	tCellAttrRightInset:*Real*	tcari	<tCellAttrRightInset:9>
Cell: right stroke color	tCellRightStrokeColor: *Definition*	tcrsc	<tCellRightStrokeColor:Blue>
Cell: right stroke tint	tCellAttrRightStrokeTint:*Real*	tcarst	<tTextCellRightStrokeTint:70>
Cell: right stroke type	tcRightStrokeType:*Enum*	tcrst	<tcRightStrokeType:ThinThick>
Cell: right stroke gap tint	tCellRightStrokeGapTint:*Real*	tcrsgt	<tCellRightStrokeGapTint:95>
Cell: right stroke gap color	tCellRightStroke GapColor:*Real*	tcrsgc	<tCellRightStrokeGapColor:C\=100 M\=0 Y\=0 K\=0>
Cell: right stroke gap over-print	tCellRightStrokeGapOver-print:*Boolean*	tcrsgo	<tCellRightStrokeGapOverprint:0> 1=on, 0=off (default)
Cell: rotation	tCellAttrRotation:*Enum*	tcar	<tCellAttrRotation:180> Valid values include 0 (default), 90, 180, and 270.

Attribute	Tag name	Tag abbr.	Examples and notes
Cell: stroke weight bottom	tCellAttrBottom StrokeWeight:*Real*	tcabsw	<tCellAttrBottomStrokeWeight:3>
Cell: stroke weight left	tCellAttrLeft StrokeWeight:*Real*	tcalsw	<tCellAttrLeftStrokeWeight:3>
Cell: stroke weight right	tCellAttrRight StrokeWeight:*Real*	tcarsw	<tCellAttrRightStrokeWeight:3>
Cell: stroke weight top	tCellAttrTop StrokeWeight:*Real*	tcatsw	<tCellAttrTopStrokeWeight:3>
Cell: top inset	tCellAttrTopInset:*Real*	tcati	<tCellAttrTopInset:9>
Cell: top stroke color	tCellTopStrokeColor: *Definition*	tctsc	<tCellTopStrokeColor:Blue>
Cell: top stroke tint	tCellAttrTopStrokeTint:*Real*	tcatst	<tCellAttrTopStrokeTint:70>
Cell: top stroke type	tcTopStrokeType:*Enum*	tctst	<tcTopStrokeType:ThinThick>
Cell: top stroke gap tint	tCellTopStrokeGapTint:*Real*	tctsgt	<tCellTopStrokeGapTint:95>
Cell: top stroke gap color	tCellTopStrokeGapColor:*Real*	tctsgc	<tCellTopStrokeGapColor:C\=100 M\=0 Y\=0 K\=0>
Cell: top stroke gap overprint	tCellTopStrokeGap Overprint:*Boolean*	tctsgo	<tCellTopStrokeGapOverprint:0> 1=on, 0=off (default)
Cell: type	tCellType:*Enum*	tct	<tCellType:1>
Cell: vertical composition	tTextCellVertical Composition:*Enum*	tcvc	<tTextCellVerticalComposition:1>
Cell: vertical justification	tTextCellVertical Justification:*Enum*	tcvj	<tTextCellVerticalJustification:2> Valid values include 0 (Top), 1 (Bottom), 2 (Center), and 3 (Justify)
Cell: vertical justification paragraph spacing limit	tTextCellAttrMaxVJ InterParaSpace:*Real*	ttcamvjips	<tTextCellAttrMaxVJInterParaSpace:40>
Column width	tColAttrWidth:*Real*	tcaw	<tColAttrWidth:120>
Column: fill pattern (first color)	tColFillPatFirstColor:*Enum*	tcfpfcl	<tColFillPatFirstColor:ThinThick>
Column: fill pattern (second color)	tColFillPatSecondColor:*Enum*	tcfpscl	<tColFillPatSecondColor:ThinThick>
Column: stroke pattern (first color)	tColStrokePatFirstColor:*Enum*	tcspfcl	<tColStrokePatFirstColor:ThinThick>
Column: stroke pattern (first type)	tColStrokePattern FirstType:*Enum*	tcspft	<tColStrokePatternFirstType:ThinThick>
Column: stroke pattern (second color)	tColStrokePat SecondColor:*Enum*	tcrspscl	<tColStrokePatSecondColor:ThinThick>
Column: stroke pattern (second type)	tColStrokePattern SecondType:*Enum*	tcspst	<tColStrokePatternSecondType:ThinThick>
Column: stroke pattern first weight	tColStrokePattern FirstWeight:*Real*	tcspfw	<tColStrokePatternFirstWeight:9>

Attribute	Tag name	Tag abbr.	Examples and notes
Column: stroke pattern second weight	tColStrokePattern SecondWeight:*Real*	tcspsw	<tColStrokePatternSecondWeight:9>
Column: stroke type	tColStrokeType:*Enum*	tcst	<tColStrokeType:ThickThin>
Row height	tRowAttrHeight:*Real*	trah	<tRowAttrHeight:20>
Row: AutoGrow	tRowAutoGrow:*Enum*	trag	<tRowAutoGrow:1>
Row: fill pattern (first color)	tRowFillPatFirstColor:*Enum*	trfpfcl	<tRowFillPatFirstColor:ThinThick>
Row: fill pattern (second color)	tRowFillPat SecondColor:*Enum*	trfpscl	<tRowFillPatSecondColor:ThinThick>
Row: row size maximum	tRowAttrMaxRowSize:*Real*	tramars	<tRowAttrMaxRowSize:40>
Row: row size minimum	tRowAttrMinRowSize:*Real*	tramirs	<tRowAttrMinRowSize:9>
Row: stroke pattern (first color)	tRowStrokePat FirstColor:*Enum*	trspfcl	<tRowStrokePatFirstColor:ThinThick>
Row: stroke pattern (first type)	tRowStrokePattern FirstType:*Enum*	trspft	<tRowStrokePatternFirstType:ThinThick>
Row: stroke pattern (second color)	tRowStrokePat SecondColor:*Enum*	trspscl	<tRowStrokePatSecondColor:ThinThick>
Row: stroke pattern (second type)	tRowStrokePattern SecondType:*Enum*	trspst	<tRowStrokePatternSecondType:ThinThick>
Row: stroke pattern first weight	tRowStrokePattern FirstWeight:*Real*	trspfw	<tRowStrokePatternFirstWeight:9>
Row: stroke pattern second weight	tRowStrokePattern SecondWeight:*Real*	trspsw	<tRowStrokePatternSecondWeight:9>
Row: stroke type	tRowStrokeType:*Enum*	trst	<tRowStrokeType:ThickThin>
Table: column end pattern (start value)	tTablerColFillPattern EndValue:*Real*	trcfpev	<tTablerColFillPatternEndValue:1>
Table: column fill pattern (first count)	tTableColFillPattern FirstCount:*Real*	tcfpfc	<tTableColFillPatternFirstCount:2>
Table: column fill pattern (second count)	tTableColFillPattern SecondCount:*Real*	tcfpsc	<tTableColFillPatternSecondCount:2>
Table: column fill pattern (start value)	tTableColFill PatternStartValue:*Real*	tcfpsv	<tTableColFillPatternStartValue:1>
Table: column stroke color	tColStrokeColor:*Definition*	tcsc	<tColStrokeColor:Blue>
Table: column stroke pattern (first count)	tTableColStrokePattern FirstCount:*Real*	tcspfc	<tTableColStrokePatternFirstCount:2>
Table: column stroke pattern (second count)	tTableColStrokePattern SecondCount:*Real*	tcspsc	<tTableColStrokePatternSecondCount:2>
Table: column stroke weight	tColStrokeWeight:*Real*	tcsw	<tColStrokeWeight:3>
Table: fill pattern priority	tTableFillPatternPriority:*Real*	tfpp	<tTableFillPatternPriority:1>
Table: outer bottom stroke tint	tOuterBottomStrokeTint:*Real*	tobst	<tOuterBottomStrokeTint:70>

Attribute	Tag name	Tag abbr.	Examples and notes
Table: outer bottom stroke weight	tOuterBottom StrokeWeight:*Real*	tobsw	<tOuterBottomStrokeWeight:3>
Table: outer left stroke tint	tOuterLeftStrokeTint:*Real*	tolst	<tOuterLeftStrokeTint:70>
Table: outer left stroke weight	tOuterLeftStrokeWeight:*Real*	tolsw	<tOuterLeftStrokeWeight:3>
Table: outer right stroke tint	tOuterRightStrokeTint:*Real*	torst	<tOuterRightStrokeTint:70>
Table: outer right stroke weight	tOuterRight StrokeWeight:*Real*	torsw	<tOuterRightStrokeWeight:3>
Table: outer top stroke tint	tOuterTopStrokeTint:*Real*	totst	<tOuterTopStrokeTint:70>
Table: outer top stroke weight	tOuterTopStrokeWeight:*Real*	totsw	<tOuterTopStrokeWeight:3>
Table: row end pattern (start value)	tTableRowFillPattern EndValue:*Real*	trfpev	<tTableRowFillPatternEndValue:1>
Table: row fill pattern (first count)	tTableRowFillPattern FirstCount:*Real*	trfpfc	<tTableRowFillPatternFirstCount:2>
Table: row fill pattern (second count)	tTableRowFillPattern SecondCount:*Real*	trfpsc	<tTableRowFillPatternSecondCount:2>
Table: row fill pattern (start value)	tTableRowFill PatternStartValue:*Real*	trfpsv	<tTableRowFillPatternStartValue:1>
Table: row keeps	tRowKeeps:*Real*	trk	<tRowKeeps:3>
Table: row stroke color	tRowStrokeColor:*Definition*	trsc	<tRowStrokeColor:Blue>
Table: row stroke pattern (first count)	tTableRowStrokePattern FirstCount:*Real*	trspfc	<tTableRowStrokePatternFirstCount:2>
Table: row stroke pattern (first count)	tTableRowStrokePattern SecondCount:*Real*	trspsc	<tTableRowStrokePatternSecondCount:2>
Table: row stroke weight	tRowStrokeWeight:*Real*	trsw	<tRowStrokeWeight:3>
Table: stroke drawing order	tStrokeOrder:*Real*	tso	<tStrokeOrder:3> Possible values are 0-3: 0 - Best Join 1 - Row Strokes in Front 2 - Column Strokes in Front 3 - InDesign 2.0 Compatability
Table: spacing after	tAfterSpace:*Real*	tas	<tAfterSpace:9>
Table: spacing before	tBeforeSpace:*Real*	tbs	<tBeforeSpace:9>

Hyperlink and index tags

Attribute	Tag name	Tag abbr.	Examples and notes
Hyperlink	Hyperlink:*Definition*	hpl	<Hyperlink:=<HyperlinkName:Page 3>>
Hyperlink destination definition	HyperlinkDestDefn:*Definition*	hpldestdfn	<HyperlinkDestDefn:=<HyperlinkName:Page 3>>
Hyperlink name	HyperlinkName:*Real*	hplname	<HyperlinkName:Page 3>
Hyperlink destination	HyperlinkDest:*Real*	hpldest	<HyperlinkDest
Hyperlink length	HyperlinkLength:*Real*	hpllen	<HyperlinkLength:6>

Specify the number of characters in the hyperlink. |
Hyperlink destination name	HyperlinkDestName:*Real*	hpldest-name	<HyperlinkDestName:Page 3>
Hyperlink destination URL	HyperlinkDestUrl:*Real*	hpldesturl	<HyperlinkDestUrl:http\:\/\/www.adobe.com>
Hyperlink destination index	HyperlinkDestIndex:*Real*	hpldestidx	<HyperlinkDestIndex:1>
Hyperlink destination page	HyperlinkDestPage:*Real*	hpldestpage	<HyperlinkDestPage:3>
Hyperlink destination page zoom type	HyperlinkDestPage ZoomType:*Real*	hpldestpag-ezoomtype	<HyperlinkDestPageZoomType:Fixed>

Valid values include Fixed, Fit View, Fit in Window, Fit Width, Fit Height, Fit Visible, and Inherit Zoom. |
| Hyperlink destination page zoom factor | HyperlinkDestPageZoom Factor:*Real* | hpldestpag-ezoomfactor | <HyperlinkDestPageZoomFactor:1.7>

1=100% magnification |
Hyperlink hidden	Hidden:*Boolean*	hid	<Hidden:0>
Hyperlink border width	BrdrWidth:*Real*	brdrw	<BrdrWidth:2>
Hyperlink border visible	BrdrVisible:*Boolean*	brdrv	<BrdrVisible:1>
Hyperlink border highlight	BrdrHilight:*Boolean*	brdrh	<BrdrHilight:1>
Hyperlink border style	BrdrStyle:*Enum*	brdrs	<BrdrStyle:1>
Hyperlink border color	BrdrColor:*Real*	brdrc	<BrdrColor:Blue>
Index entry	IndexEntry	Idx	<IndexEntryType:IndexPageEntry> <IndexEntryStyleUid:0><IndexEntryRangeType: kCurrentPage><IndexEntryDisplayString:simple page index>>
Index entry type	IndexEntryType	IdxEnType	<IndexEntryType:IndexPageEntry>
Index entry range type	IdexEntryRangeType	IdxEnRng-Type	<IndexEntryRangeType:kForNextNParagraphs>
Index entry style	IndexEntryStyle	IdxEnStyle	<IndexEntryStyle:MyStyle>
Index entry next style	InexEntryNextStyle	IdxEnNxt-Style	<IndexEntryNextStyle:MyStyle>
Index entry number of paragraph or page	IndexEntryNumParOrPage	IdexEn-NumPgOr-Par	<IndexEntryNumParOrPage:30>
Index entry display string	IndexEntryDisplayString	IdxEnDispStr	<IndexEntryDisplayString:simple page index>

Attribute	Tag name	Tag abbr.	Examples and notes
Index entry sorting string	IndexEntrySortString	IdxEnSortStr	<IndexEntrySortString:An index>
Index cross-reference entry type	IndexXRefEntryType	IdxXrEnType	<IndexXRefEntryType:kSeeHerein>
Index cross-reference entry custom string	IndexXRefEntryCustomString	IdxXrEn-CusStr	<IndexXRefEntryCustomString:abc>

InDesign tags for special characters

If you want to add special characters into a tagged text file, type the following values into the tagged text file.

Special character	Value
Auto page numbering	<0xE0018>
Section marker	<0xE0019>
Bullet character	<0x2022>
Copyright symbol	<0x00A9>
Degree symbol	<0x00B0>
Ellipsis	<0x2026>
Paragraph symbol	<0x00B6>
Registered trademark symbol	<0x00AE>
Section symbol	<0x00A7>
Trademark symbol	<0x2122>
Em dash	<0x2014>
Em space	<0x2003>
En dash	<0x2013>
En space	<0x2002>
Flush space	<0x2001>
Hair space	<0x200A>
Nonbreaking space	<0x00A0>
Thin space	<0x2009>
Discretionary hyphen	<0x00AD>
Nonbreaking hyphen	<0x2011>
Double left quotation mark	<0x201C>
Double right quotation mark	<0x201D>
Single left quotation mark	<0x2018>
Single right quotation mark	<0x2019>