A Guide to XPress Tags 2021

November 2021

Brilliant content that works

quark.com

Contents

Understanding XPress Tags

With XPress Tags Filter XTensions software, you can use plain text files to import and export text into QuarkXPress layouts with paragraph and character attributes already applied. Because the plain text file format doesn't actually support attributes, the formatting is indicated with codes that are translated by the XPress Tags filter. As the new character and paragraph formatting options are added to QuarkXPress, new XPress Tags codes are added to support the new attributes.

Importing tagged text

To import a plain text file that contains embedded XPress Tags codes:

1. Choose **File > Import**.
2. Select the target XPress tags file.
3. The application attempts to detect the encoding of the XPress Tags file. To specify a different encoding, choose an option from the **Encoding** drop-down menu.
4. To convert XPress Tags codes into actual character attributes and paragraph formats, check **Interpret XPress Tags** in the **Import Text** dialog box. If you do not check **Interpret XPress Tags** in the **Import Text** dialog box before importing tagged text, QuarkXPress does not convert XPress Tags codes. Instead, the program imports the codes as text characters.

Generating tagged text in another application

To include XPress Tags information in a text file that you created in another application, precede the text with the codes you want. Begin the text file with a version code and encoding indication (for example: <v17.00><e8>). For more information about encoding codes, see "*Encoding*".

Save the text you generate in the plain text file format using the appropriate encoding.

General information on XPress tags

When entering XPress Tags codes, keep the following in mind:

- XPress Tags codes are case-sensitive.
- XPress Tags codes for character and paragraph attributes must begin and end with angle brackets (< >). For example, the XPress Tags code for boldface text is <B>.
- To combine character attribute codes, begin with a left angle bracket, then enter the codes you want to specify, and complete the code with a right angle bracket. For example, the code for bold italic text is <BI>.

- XPress Tags codes for character attributes must be placed immediately preceding the characters to which you want to apply the attributes. When you apply a character attribute using an XPress Tags code, the attribute remains in effect until you cancel it or until you enter codes that change the style sheet. You can cancel an attribute by re-specifying its code following the last character to which you want it applied (or for type styles, you can enter the code for plain text, <P>).
- For XPress Tags code commands that let you specify more than one value (such as paragraph attributes), you can enter a $ in place of an actual value. When QuarkXPress encounters a $ code, the program substitutes the value specified in the currently applied style sheet. (If no style sheet is currently applied, the value of the Normal style sheet is used.) For example, you might want a paragraph to contain all the formats specified in the applied style sheet, but you want to apply 18 points of leading instead of the value specified in the style sheet. The code for this would be: <*p($,$,$,$,18,$,$,$,$,$,$($,$))>.
- Names you specify as XPress Tags codes must be preceded and terminated by an inch mark " character. For example, if you want to specify the font AvenirLTStd-Roman, use the <f"AvenirLTStd-Roman"> code.
- Some features (such as rubi text, group characters, and emphasis marks) are only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can open a project that uses such features in any edition of QuarkXPress, and you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Character and Paragraph Formatting

When you specify type styles using XPress Tags codes, <P> always sets the type style to **Plain**. When you specify any other type style (for example, <B> for **Bold**), that style is applied if it isn't already specified, and is removed if it has been specified. For example, the first time you specify <B>, the **Bold** type style is applied to the text that follows. If you enter <B> again, the **Bold** type style is not applied to the text that follows. If you enter <$>, the type style is set to the one specified in the current style sheet. If a style sheet is not currently applied, the **Normal** style sheet is used.

Font Styling

Font name can be specified in XPress Tags either by using the Postscript name of the font or by using the Full Font name.

Example for Font PS Name:

@Times New Roman Italic=<PIs100p100t0Y1h100z12k0b0cKf"TimesNewRomanPSMT"n0o("Calt","liga","locl")L0G0>

@Times New Roman Plain=<Ps100p100t0Y1h100z12k0b0cKf"TimesNewRomanPSMT"n0o("Calt","liga","locl")L0G0>

where TimesNewRomanPSMT is the Font PS Name and Italic Font Styling is indicated by I

Example for Full Font Name:

@Times New Roman Italic=<PIs100p100t0Y1h100z12k0b0cKf"Times New Roman"n0o("Calt","liga","locl")L0G0>

@Times New Roman Plain=<Ps100p100t0Y1h100z12k0b0cKf"Times New Roman"n0o("Calt","liga","locl")L0G0>

where Times New Roman is the Full Font Name and Italic Font Styling is indicated by I

Character attributes

- Plain: <P>
- Bold: <B>
- Italic: <I>
- Underline: <U>
- Word Underline: <W>

- Strikethrough: </>
- Double Strikethrough: <R>
- Outline: <O>
- Shadow: <S>
- All Caps: <K>
- Small Caps: <H>
- Superscript: <+>
- Subscript: <-> (hyphen)
- Superior: <V>
- Change Font ✦: <f"font name">
- Change Font Size ✦: <z###.##> in points
- Change Color*: <c"color name"> or <cC, cM, cY, cK, and cW>
- Change Shade ✦: <s###.#> in percentage of shade
- Horizontal Scale ✦: <h###.#> in percentage of scale
- Vertical Scale ✦: <y###.#> in percentage of scale
- Kern ✦: <k###.##> in 1/200 em space
- Track ✦: <t###.##> in 1/200 em space
- Sending ❗: Same tag as tracking but with a value in points added to the end
- Baseline Shift ✦: <b###.##> in points
- Ligatures: <G1> to turn on, or <G0> to turn off
- Opacity ✦: <p###.#> in percentage of opacity
- Emphasis Mark ❗: <Mn>, where n is the emphasis mark
- OpenType: <o("xxxx")> where “xxxx” = the OpenType feature
- Language: <n##> (see “*Languages*“)
- Keep Half Width Characters Upright ❗ (a character attribute used only in vertical stories): <Ln>, where <L0> = sideways, <L1> = upright, and <L$> = revert to style sheet
- Emphasis Mark ❗: <Mn>, where n is the emphasis mark

- Apply Sending to Non-CJK characters ●: <Y1> to turn on, <Y0> to turn it off, or <Y$> to use the current style sheet setting

When any of these commands is followed by a$ (for example, <f$>), the attributes are set to the values specified in the current character style sheet. If a character style sheet is not currently applied, the Normal style sheet is used. (A # character in this list indicates a numeric value.)

The ● icon indicates text features that are considered East Asian features and are only available when you have the East Asian preference enabled (QuarkXPress/Edit (> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Languages

You can apply the following languages with the <n##> character tag.

- None: <n254>
- Bulgarian: <n72>
- Catalan: <n73>
- Chinese (Simplified): <n52>
- Chinese (Traditional): <n53>
- Croatian: <n68>
- Czech: <n56>
- Danish: <n9>
- Dutch: <n5>
- English (International): <n2>
- English (US): <n0>
- Estonian: <n44>
- Finnish: <n17>
- French: <n1>
- German: <n3>
- German (Reformed): <n70>
- German (Swiss Reformed): <n69>
- German (Swiss): <n19>
- Greek: <n20>

- Hungarian: <n43>
- Icelandic: <n21>
- Italian: <n4>
- Japanese: <n14>
- Korean: <n51>
- Latvian: <n45>
- Lithuanian: <n41>
- Norwegian (Bokmal): <n12>
- Norwegian (Nynorsk): <n80>
- Polish: <n42>
- Portuguese (Brazilian): <n71>
- Portuguese (European): <n10>
- Romanian: <n39>
- Russian: <n49>
- Slovak: <n57>
- Slovenian: <n66>
- Spanish: <n8>
- Swedish: <n7>
- Turkish: <n24>
- Ukrainian: <n62>
- Hindi: <n33>
- Bengali: <n60>
- Punjabi: <n95>
- Tamil: <n90>
- Malayalam: <n89>
- Odia: <n63>
- Telugu: <n87>
- Kannada: <n93>
- Gujarati: <n94>
- Marathi: <n99>

- Assamese: <n38>

Rubi text

Rubi text lets you annotate base characters with smaller rubi characters. ❗ This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Rubi text tags use the following form:

<A(\#68Base text\#36<@><Character attributes><\#9Rubi text\#132,50,C,A,0,2,B,T)>

The contents of this tag are as follows:

- <A()> encloses the tag.
- /#68 and /#36 mark the beginning and end of the base text.
- <@><Character attributes> lets you style the rubi text (see "*Character attributes*").
- \#n marks the beginning of the rubi text, where n = the number of rubi characters (in the above example, 9).
- \#132 marks the end of the rubi text and the beginning of the rubi positioning codes.
- 50 is a percentage of the base font size.
- C is alignment, where L = left, C = center, R = right, J = justified, F = forced, O = 1–2–1 (JIS) rule, and E = equal space.
- A is base alignment, where A = none, L = left, C = center, R = right, J = justified, F = forced, O = 1–2–1 (JIS) rule, and E = equal space.
- 0 is horizontal offset of the rubi in points.
- 2 is overhang, where 0 = none, 1 = up to one rubi character, 2 = up to 1/2 rubi character, 3 = up to one base character, 4 = up to 1/2 base character, and 5 = unrestricted.
- B is placement relative to base text, where A = Above and B = below.
- T is auto-alignment, where T = true and F = false.

Group characters

Character grouping lets you group horizontal characters in a vertical text story. ❗ This feature is an East Asian feature and is only available when you have the

East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Group character tags use the following form:

```
<A(\#72ABC\#40<t–10h80>)>
```

The contents of this tag are as follows:

- <A()> encloses the tag.
- /#72 and /#40 mark the beginning and end of the grouped characters.
- <t–10h80> are applications of tracking and horizontal scaling (see “*Character attributes*“).

Font sets

Font sets let you specify a group of fonts to be applied to different types of text in a text run. Like style sheet definitions, font set definitions are stored in XPress Tags files. For example, if you export text that uses a font set named “Font set 1,” the definition of Font set 1 is exported in the form of a tag like the following, at the beginning of the XPress Tag file:

```
@Font set 1=[F]<"HiraMinPro-W3",100,0,h100,"GrecoStd-DB",100,0,h100,
"Corbel",100,0,h100,"HiraMinPro-W3",89.999,-5,h105,"Times-Roman",100,0,h100>
```

```
<h105z10.89b-0.6f"HiraMinPro-W3",("Contest 1",12)>
```

The contents of this tag are as follows:

- h105 = horizontal or vertical scaling (see “*Character attributes*“).
- z10.8 = percentage of current font size
- b-0.6 = baseline shift, in points
- f"HiraMinPro-W3" = font
- ("Font set 1", 12) = name of the font set to be applied and current point size of the font

❗ This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit (> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

OpenType attributes

Turn on/off styling is similar to the current method used for type styles. Re-specifying a tag toggles the current state of the attribute.

An asterisk ✦ in the topics below indicates a glyph attribute. The format for applying glyph attributes is: <DoO,F"zero",I0,f"ACaslonPro-Bold">0<oC>, where DoO = opening tag, F"zero" = feature, I0 = glyph index (where I0 = the first glyph), f"AcaslonPro-Bold" = font, 0 = base character, and oC = close tag.

Basic set

- All Small Caps: a2sc
- Alternative Annotation Forms ✦: nalt
- Contextual Alternates: calt
- Discretionary Ligatures: dlig
- Denominators: dnom
- Fractions: frac
- Hangul ✦: hngl
- Historical Forms: hist
- Horizontal Kana Alternates ❗: hkna
- Italics ❗: ital
- Standard Ligatures: liga
- Numerators: numr
- Ordinals: ordn
- Ornaments: ornm
- Lining Figures: lnum
- Oldstyle Figures: onum
- Ruby Notation Forms ❗: ruby
- Scientific Inferiors: sinf
- Stylistic Alternates: salt
- Small Caps: smcp
- Subscript: subs
- Superscript: sups
- Swash: swsh
- Titling Alternates: titl

- Tabular Figures: tnum
- Proportional Figures: pnum
- Slashed Zero: zero
- Vertical Kana Alternates ❗: vkna
- Small Capitals from Capitals: c2sc
- Case Sensitive Forms: case
- Capital Spacing: cpsp
- Historical Ligatures: hlig
- Petite Capitals: pcap
- Alternative Fractions : afrc
- Stylistic Set 0: ss00
- Stylistic Set 1: ss01
- Stylistic Set 2: ss02
- Stylistic Set 3: ss03
- Stylistic Set 4: ss04
- Stylistic Set 5: ss05
- Stylistic Set 6: ss06
- Stylistic Set 7: ss07
- Stylistic Set 8: ss08
- Stylistic Set 9: ss09
- Stylistic Set 10: ss10
- Stylistic Set 11: ss11
- Stylistic Set 12: ss12
- Stylistic Set 13: ss13
- Stylistic Set 14: ss14
- Stylistic Set 15: ss15
- Stylistic Set 16: ss16
- Stylistic Set 17: ss17
- Stylistic Set 18: ss18
- Stylistic Set 19: ss19

- Stylistic Set 20: ss20
- Stylistic Set 21: ss21
- Stylistic Set 22: ss22
- Stylistic Set 23: ss23
- Stylistic Set 24: ss24
- Stylistic Set 25: ss25
- Stylistic Set 26: ss26
- Stylistic Set 27: ss27
- Stylistic Set 28: ss28
- Stylistic Set 29: ss29
- Stylistic Set 30: ss30
- Stylistic Set 31: ss31
- Stylistic Set 32: ss32
- Stylistic Set 33: ss33

✦ This is a glyph attribute; see above.

❗ This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Alternate forms

- Expert Forms ✦: expt
- Hojo Kanji Forms ✦: hojo
- JIS2004 Forms ❗: jp04
- JIS78 Forms ❗: jp78
- JIS83 Forms ❗: jp83
- JIS90 Forms ❗: jp90
- NLC Kanji Forms ✦: nick
- Simplified Forms ❗: smpl

- Traditional Forms ❗: trad
- Traditional Name Forms ✦: tnam

✦ This is a glyph attribute; see above.

❗ This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Alternate metrics

- Full Widths ❗: fwid
- Half Widths ❗: hwid
- Third Widths ❗: twid
- Quarter Widths ❗: qwid
- Proportional Widths ❗: pwid
- Alternate Vertical Metrics ❗: valt
- Proportional Alternate Widths ❗: palt
- Proportional Alternate Vertical Metrics ❗: vpal
- Alternate Half Widths ❗: halt
- Alternate Vertical Half Metrics ❗: vhal

✦ This is a glyph attribute; see above.

❗ This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Paragraph attributes

XPress Tags codes for paragraph attributes must be placed at the beginning of a paragraph. Formats specified by XPress Tags codes remain applied until you specify other values at the beginning of a subsequent paragraph or until you enter codes that change the style sheet.

- Left-align paragraph: <*L>
- Center-align paragraph: <*C>
- Right-align paragraph : <*R>

- Justify paragraph: <*J>
- Force Justify paragraph: <*F>
- Towards Spine paragraph: <*T>
- Away from Spine paragraph: <*W>
- Character Alignment !: <*An>, where n indicates the type of alignment; <AT> = Em box Top/Right, <AO> = ICF Top/Rright, <AC> = Center, <AM> = ICF Bottom/Left, <AB> = Em box Bottom/Left, <AL> = Baseline
- Set Tab stops ✦: <*t(##.#,#,"1 or 2 characters")>. Values in parentheses represent **Position** (in points), **Alignment**, and **Fill Characters**. For no fill characters, enter a 1 followed by two spaces. For 1 fill character, enter a 1 followed by the repeat character (enter the character twice). For 2 fill characters, enter a 2 followed by the alternating characters. Alignment options are 0 for **Left**, 1 for **Center**, 2 for **Right**, 4 for **Decimal**, 5 for **Comma**, and any printing character (in quotation marks) for **Align On**. Note: Reverting text to the Normal style sheet does not revert the alignment to the alignment specified in the Normal style sheet. Instead, you must specify the new alignment in addition to reapplying the Normal style sheet. All the tabs for a paragraph are included, separated by commas, within the parentheses.
- Set Paragraph attributes ✦: <*p(##.#,##.#,##.#,##.#,##.#,##.#,##.#,##,##,G or g(P,S))>. Values in parentheses represent **Left Indent**, **First Line Indent**, **Right Indent**, **Last Line Indent**, **Leading**, **Space Before**, **Space After**, **Space Between, Use Space Before** and **Lock to Grid**. For example, <*p(16,36,16,-8,14,9,12,0,2,g(P,S))> would format text that has a 16 pt **Left Indent**, 36 pt **First Line Indent**, 16 pt **Right Indent**, -8 pt **Last Line Indent,** 14 pt **Leading**, 9 pt **Space Before**, 12 pt **Space After**, 0 pt Space Between, 2 **Space Before applied both at column top and between paragraphs** and g does not **Lock to Grid**. You can specify the first line of a paragraph as a percentage of an Em space by using a decimal percentage value in the **First Line Indent** position (for example, use 0.6 for 60%). In **Space Between**, -1 will Ignore Space Between, 0 or any numerical value will apply Space Between. In **Use Space Before**, 0 will **apply Space Before only between paragraphs**, 1 will **apply Space Before only at column top**, 2 will **apply Space Before both at column top and between paragraphs**. If you specify the leading of a paragraph as a percentage, it is termed as **Proportional Leading**. The Proportional Leading value can be both positive and negative. In the **Lock to Grid** position, G = **lock to grid**; g = **do not lock to grid**, and the two values in parentheses indicate whether to **lock to the page grid** (P) or text **box grid** (B) and how to align the characters (T = **top/right**, C = **center**, S = **baseline**, B = **bottom/left**).
- Hyphenation and Justification specification: <*h"specification name">. **Hyphenation and Justification specifications** must already exist within the QuarkXPress project before you import tagged text that specifies them. If

you specify a hyphenation and justification specification in XPress Tags and QuarkXPress cannot locate the specification in the project's list of hyphenation and justification specifications, the Standard hyphenation and justification specification is substituted. The maximum length for the names of style sheets and of hyphenation and justification specifications is 63 characters.

- Paragraph Rule Above ✦✦: <*ra(##,#,"color name",#,##,##,## or ##%)>. Values in parentheses represent **Width**, **Style**, "color name", **Shade**, **From Left**, **From Right**, and **Offset**. Enter rule widths and indents in points. You can enter "T" in front of the left indent value to base rule length on the first line of text. **Rules Style** options are 0 for **Solid**, 1 for **Dotted**, 2 for **Dotted 2**, 3 for **Dash Dot**, 4 for **All Dots**, 5 for **Double**, 6 for **Thin-Thick**, 7 for **Thick-Thin**, 8 for **Thin-Thick-Thin**, 9 for **Thick-Thin-Thick**, and 10 for **Triple**. **Shade** is a percentage. You can specify a rule's offset value in points or as a percentage (#%). For example, <*ra(4,5,"Blue",100,T12,12,50%)> would format a **Rule Above** 4 points wide in the **Double** style, 100% Blue, indented 12 points **From Left** and **From Right** (based on the first line of text), and **Offset** 50%.

- Paragraph Rule Below ✦✦: <*ra(##,#,"color name",#,##,##,## or ##%)>. Values in parentheses represent **Width**, **Style**, "color name", **Shade**, **From Left**, **From Right**, and **Offset**. Enter rule widths and indents in points. You can enter "T" in front of the left indent value to base rule length on the first line of text. **Rules Style** options are 0 for **Solid**, 1 for **Dotted**, 2 for **Dotted 2**, 3 for **Dash Dot**, 4 for **All Dots**, 5 for **Double**, 6 for **Thin-Thick**, 7 for **Thick-Thin**, 8 for **Thin-Thick-Thin**, 9 for **Thick-Thin-Thick**, and 10 for **Triple**. **Shade** is a percentage. You can specify a rule's offset value in points or as a percentage (#%). For example, <*rb(2,0,C,50,6,18,30%)> would format a **Rule Below** 2 points wide in the **Solid** style, 50% Cyan, indented 6 points **From Left** and 18 points **From Right**, and **Offset** 30%.

- Drop Caps ✦✦: <*d(#,#)> where d = **Drop Caps**, #,# = **Character Count**, **Line Count**.

- Keep with Next Paragraph ✦✦: <*kn1> or <*kn0> where kn = **Keep with Next Paragraph**; 1 = **allow keep with next paragraph**; 0 = **do not keep with next paragraph**

- Keep Lines Together ✦✦: <*ktA> or <*kt(#,#)> where kt = **Keep Lines Together**; A = All; #,# = **Numbers of lines at Start of Paragraph**, **Number of Lines at End of Paragraph**. To return to the setting used in the **Normal** style sheet, enter <*kt($)>

- Start Paragraph ✦✦ : <*ks1> where ks = **Start Paragraph**, 0 = **Anywhere**, 1 = **In Next Column**, 2 = **In Next Box**, 3 = **On Next Page**, 4 = **On Next Odd Page**, 5 = **On Next Even Page**. It is associated with Keep options.

- Hanging Character Set: <*s"Hanging Character Set Name">. If the layout does not contain a **Hanging Character Set** by this name, no hanging character set is applied. However, if you subsequently add a hanging character set with this name, that hanging character set is applied to the text.
- Mojigumi Set ❗ : <*m"Mojigumi Set Name">. If the layout does not contain a **Mojigumi Set** by this name, no Mojigumi set is applied. However, if you subsequently add a Mojigumi set with this name, that Mojigumi set is applied to the text.
- Bullet style: <*bb"Bullet Style Name"(7.2,10,0)>. If the layout does not contain a **Bullet Style** by this name, no bullet style is applied. In the parenthesized list, the first number is the **Minimum Bullet Distance from Text**, the second number is the **Start At** value, and the third number is for **Restart Numbering** locally (0 = false, 1 = true). Note that the last value is irrelevant for bullet styles.
- Numbering Style: <*bb"Numbering Style Name"(7.2,5,0)>. If the layout does not contain a **Numbering Style** by this name, no numbering style is applied. In the parenthesized list, the first number is the **Minimum Number Distance from Text**, the second number is the **Start At** value, and the third number is for **Restart Numbering** locally (0 = false, 1 = true).
- Outline Style: <*bo"Outline Style Name"(20,10,0,4,1,0,3)>. If the layout does not contain an **Outline Style** by this name, no outline style is applied. In the parenthesized list, the first number is the **Minimum Number Distance from Text**, the second number is the **Start At** value, the third number is for **Restart Numbering** locally (0 = false, 1 = true), 4 is **Level**(values range from 0-8; 0= Level 1, 1= Level 2, ...), 1 is **Restart Numbering After** is True (values are 0 and 1), 0 is **Restart numbering After Level** (0= Level, 1= Style), 3 is **Level** for **Restart numbering After** (values range from 0-9; 0= After Any Previous Level, 1= Level 1, 2= Level 2, ...)
- Outline Style: <*bo"Outline Style Name"(20,10,0,4,1,1,4)>. If the layout does not contain an **Outline Style** by this name, no outline style is applied. In the parenthesized list, the first number is the **Minimum Number Distance from Text**, the second number is the **Start At** value, and the third number is for **Restart Numbering** locally (0 = false, 1 = true), 4 is **Level**(values range from 0-8; 0= Level 1, 1= Level 2, ...), 1 is **Restart Numbering After** is True (values are 0 and 1), 1 is **Restart numbering after Style** (0= Level, 1= Style), 4 is Row index of Paragraph Style Sheet in **Style** pop up.

✦ If a $ replaces any or all format codes (for example, <*t$>), the current paragraph style sheet values are used. If a style sheet is not currently applied, the **Normal** style sheet is used. All numeric values in these two commands are measured in points.

Any or all of the format codes can be replaced by a $ to use the current style sheet's definition, or by a 0 to specify no rule (for example, <*ra$> and <*ra0>).

This feature is an East Asian feature and is only available when you have the East Asian preference enabled (QuarkXPress/Edit(> Preferences > East Asian). However, you can import and export text that uses these features in XPress Tags format using any language edition of QuarkXPress.

Other attributes

This section lists attributes that do not fit into the other categories.

- Glyph: <DoO,F"zero",I0,f"ACaslonPro-Bold">0<oC>, where DoO = opening tag, F"zero" = Feature, I0 = index of the alternate glyph, f"AcaslonPro-Bold" = Font, 0 = base character, and oC = close tag.
- Hyperlink: <A(3,"HYPB",\#002\#000\#000\#000)[2]>Linked text<A(3,"HYP\"",\#nnn\#xxx)[n]>, where <A(3,"HYPB",\#002\#000\#000\#000)[2]> is the opening tag (this never changes), \#nnn is the type of link (\#000 = URL, \#004 = anchor, \#008 = page), and \#xxx indicates the creation order of this hyperlink in the project (\#001 = first created, \#002 = second created, and so forth).
- Transcoding sequences: <EX>Unicode value, language, legacy code value<EX>. For language codes, see "*Languages*."
- Unencoded glyphs: <DO, gxxxx, f"Font"P> <DC>, where DO = opening tag, g = glyph, f = font, P = style (P, B, I, or BI), and DC = close tag
- Redline:

Insertion:

<A(3,"QDDA",)[0]>what<A(3,"QDD!",\#027\#000\#000\#000\#001\#000\#001\#000\#000\#000\#001\#206\#176\#222\#220\#006\#000\#000\#000s\#000i\#000n\#000a\#000i\#000r\#000)[1506]> where what is the inserted text.

Deletion:

<A(3,"QDDa",\#041\#000\#000\#000\#000\#000\#005\#000\#000\#000\#001\#227\#176\#222\#220\#006\#000\#000\#000s\#000i\#000n\#000a\#000i\#000r\#000\#005\#000\#000\#000s\#000o\#000m\#000e\#000 \#000)[2017]> where some is the deleted text #000s\#000o\#000m\#000e

- Note: <A(3,"QCDP",'\#000\#000\#000\#002\#000\#000\#000\#008\#000\#000\#000\#001\#181q\#229\#220\#000\#000\#000\#000\#000\#000\#000\#000\#006\#000\#006\#000\#000\#000j\#000a\#000m\#000e\#000s\#000\#000N\#000o\#000t\#000e\#000 \#000t\#000e\#000x\#000t)[2342]><A(3,"QCD0",\#000)[0]> where james is user name who inserted the note; Note text is the text within the note.

Turn on/off styling

- Turn on feature “a”: <o("aaaa")>
- Turn on features “x”, “y”, and “z”: <o("xxxx", "yyyy", "zzzz")>
- Turn on features “a” and “b”, then turn on feature “c” and turn off feature “b”: <o("aaaa", "bbbb")>some<o("cccc", "bbbb")> text
- Turn on features “a” and “y”: <o("xxxx", "aaaa", "xxxx", "yyyy")>

Revert styling to style sheets

- Revert features “x” and “y” to the character attributes in the currently applied paragraph style sheet: <o($"xxxx","yyyy")>
- Revert features “x” and “a” to the character attributes in the currently applied character style sheet: <o($$"xxxx","aaaa")>
- Revert all OpenType features to the character attributes in the currently applied paragraph style sheet: <o($)>
- Revert all OpenType features to the character attributes in the currently applied character style sheet: <o($$)>

Style Sheets

You can use XPress Tags to apply character style sheets and establish a relationship between paragraph and character style sheets.

Defining style sheets

Style sheet definitions may include paragraph attributes only, character attributes only, or both paragraph and character attributes.

- Define paragraph style sheet with default character attributes: @paragraph stylesheetname=[Sp"",""] <paragraph and character attributes> (Hard Return) For example, @Paragraph1=[Sp"","11"]<*L*AL*h"Standard"*s"None"*m"None"*bn(7.2,1,0)*kn0*kt0*ks0*ra0*rb0*d0*p(0,0,0,0,0,0,0,-1,0,g(P,S))Ps100p100t0Y1h100z12k0b0cKf"Helvetica"n0o("Calt","liga","locl")L0G0>
- Define character style sheet: @character stylesheetname=<character attributes>(Hard Return) For example, @Character1=Ps100p100t0Y1h100z12k0b0cKf"Helvetica"n0o("Calt","liga","locl")L0G0>
- Define paragraph style sheet with character style sheet: @paragraph stylesheetname=[Sp"","paragraph stylesheetname","character stylesheetname"]<paragraph attributes>(Hard Return) For example, @Paragraph1=[Sp"","Paragraph1","Character1"]<*L*AL*h"Standard"*s"None"*m"None"*bn(7.2,1,0)*kn0*kt0*ks0*ra0*rb0*d0*p(0,0,0,0,0,0,0,-1,0,g(P,S))>
- Base one paragraph style sheet on another paragraph style sheet, and apply Next Style paragraph style sheet: @paragraph stylesheetname=[Sp"based on paragraph stylesheetname","next paragraph stylesheetname","character stylesheetname"]<paragraph attributes>(Hard Return) For example, @Paragraph2=[Sp"Paragraph1","Paragraph3","Character1"]<>
- Base one character style sheet on another character style sheet: @stylesheetname=[St"","","","based on character stylesheetname"] <character attributes>(Hard Return) For example, @Character2=[St"","","","Character1"]<>

Applying style sheets

The @ character is used to apply a style sheet. When applying a character style sheet, you can set all character attributes to the character style sheet's default attributes by preceding @ with an "x". This clears any existing character attribute overrides. For example, <x@$> applies the Normal character style sheet, erasing any existing character attributes.

Applying a paragraph style sheet

- Apply Normal paragraph style sheet: @$:paragraph text
- Apply No Style paragraph style sheet: @:paragraph text
- Apply defined paragraph style sheet: @stylesheetname:paragraph text

To apply the **Normal** style sheet (which had attributes defined in the QuarkXPress project) to a paragraph, begin the paragraph with the @$: code.

To specify that a specific style sheet be applied to a paragraph, begin the paragraph with the @stylesheetname: code.

To specify that **No Style** be applied to paragraphs, begin the first paragraph you want to disassociate from any style sheet with the @: code.

When you apply a style sheet to a paragraph using XPress Tags, the style sheet remains applied to subsequent paragraphs until another style sheet is applied or until No Style is applied using the @: code.

You can apply attributes to characters (local formatting) within a paragraph to which you have applied a style sheet. These attributes remain applied until you cancel them or until you apply a different style sheet.

If you import text tagged with style sheet names that the project already contains, QuarkXPress automatically applies the character and paragraph attributes specified in the project's existing style sheets. If you import text tagged with style sheet names that do not already exist in the project, QuarkXPress adds each style sheet name to the **Style Sheets** palette. If a new style sheet is not defined in the tagged text, QuarkXPress applies the **Normal** paragraph and character style sheet attributes to the new style sheet and adds the paragraph style sheet to the **Style Sheets** palette.

The following characters cannot be used in style sheet names: " : = @

Applying a character style sheet

- Apply Normal character style sheet: <@$>
- Apply the paragraph's character style sheet: <@$p>
- Apply No Style character style sheet: <@>
- Apply defined character style sheet: <@stylesheetname>

Additional XPress Tags for style sheets

The possibility of a relationship between a character style sheet and a paragraph style sheet creates the need for additional XPress Tags.

- Set type style according to character attributes in the applied paragraph style sheet: <$>

- Set type style according to character attributes in the currently applied character style sheet: <$$>
- Set all character attributes according to character attributes in the applied paragraph style sheet.: <a$>

This command does not apply a character style sheet

- Set all character attributes to character attributes in the currently applied character style sheet: <a$$>

Colors

A non-process color needs to be accurately defined in order to be accurately interpreted upon import. For these colors, a definition is placed at the top of the XPress Tag file similar to style sheet definitions.

Defining a custom color

@colorname=[C]<"colorclass",colorspec>

Where:

colorname = name of the color

[C] = denotes a custom color

colorspec = "libraryname",S or P,#,"colorsubclass" #.##,#.##,#.##,#.##,#.##,#.##

S or P = “S“ indicates a spot color, “P“ indicates a process color

If it is a spot color (“S“), then a number follows to indicate the halftone screen value used # = {1 = Cyan, 2 = Magenta, 3 = Yellow, 4 = Black}

For example,

libraryname = The swatchbook or library (short) name of the color

colorsubclass = similar to colorclass. Values include: “CMYK“, “LAB“, “Hexachrome“, and “RGB.” This backup specification is used if a library is missing.

#.##,#.##,#.##,#.##,#.##,#.## = Numeric color specifications

Detailed Custom Color Definition

colorclass = {"CMYK", "RGB", "HSB", "LAB", "DIC", "MULTI-IN,", "FOCALTON,", "PANTONE®…" (there are 14 Pantone options, such as "PANTONE® solid coated"), "TOYO", “TRUMATCH”, "Web Safe Colors", "Web Named Colors"}

if colorclass = "RGB", "HSB", "LAB", "Web Safe Color," or "Web Named Colors" then colorspec = S or P,#, #.##,#.##,#.##

Example: @Red=[C]<"RGB",P,1,0,0>

if colorclass = "CMYK", then colorspec = S or P,#,#.##,#.##,#.##,#.##

Example: @CMYK-M50Y100spotY=[C]<"CMYK",S,3,0,0.5,1,0>

if colorclass = "DIC", "FOCALTONE", "PANTONE®…", "TOYO", "TRUMATCH" then colorspec = "libraryname",S or P,#,"colorsubclass",#.##,#.##,#.##,#.##,#.##,#.##

Example1: @DIC 399p spotB=[C]<"DIC","DIC 399p",S,4,"CMYK",0.55,0.14,0.47,0>

Example2: @PANTONE 259 HexC=[C]<"PANTONE® solid in hexachrome® coated","PANTONE 259 HC",S,4,"Hexachrome",0.4,1,0,0.25,0,0>

if colorclass = "MULTI-INK", then colorspec = "colorname", "colorclass",I or C,#.##,#

Where:

I or C = the color is Ink alias or custom color.

I = Ink-alias

C = Custom Ink

#.## = the multi-ink percentage

# = the ink index. This value is written only for ink-alias components

Example: @Mink-M70Lab20=[C]<"MULTI-INK","Process Magenta","CMYK",I,0.7,1"LAB-L50A45B–75spotB","LAB",C,0.2>

Applying a custom color

Once the colors are defined, they can be applied with the normal color tag. For example, <c"Red">, or <c"Mink-M70Lab20">

Special Characters

XPress Tags lets you specify character encoding and insert and manipulate special characters such as discretionary hyphens and non-breaking spaces.

Encoding

One of the following three extended character set indicators is automatically placed at the top of an XPress Tags file you create using the Save Text command (File menu).

- Unicode (UTF-16): <e8>
- Unicode (UTF-8): <e9>
- Mac Roman (x-mac-roman): <e0>
- Windows Latin (windows–1252): <e1>
- Western (iso–8859–1): <e2>
- Japanese Win (windows–932–2000): <e3>
- Japanese Mac (x-mac-japanese): <e21>
- Korean Windows (MS CodePage 949): <e19>
- Korean Mac (KSC5601): <e20>
- Traditional Chinese (BIG5): <e6>
- Simplified Chinese (GB2312): <e7>

Escape characters

To use as text certain characters that XPress Tags would otherwise consider as part of specific codes, use these special characters.

- @: <\@>
- <: <\<>
- \: <\\>

Other special characters

Some special characters have their own XPress Tags codes. Use the following codes to define these characters.

- Tab: <\t>
- Standard space ✦: <\s>

- Em space ✦: <\m>
- En space ✦: <\e>
- 3-per-Em space ✦: <\#>
- 4-per-Em space ✦: <\$>
- 6-per-Em space ✦: <\^>
- Thin space ✦: <\[>
- Hair space ✦: <\{>
- Zero width space ✦: <\z>
- Flexible space ✦: <\f>
- Figure space ✦: <\8>
- Punctuation space ✦: <\p>
- Word joiner: <\j>
- Ideographic space ✦: <\o>
- Indent Here: <\i>
- Em dash ✦: <_>
- En dash ✦: <\a>
- Hyphen ✦: <\->
- Discretionary hyphen: <\h>
- Discretionary new line: <\d>
- New line: <\n>
- New column: <\c>
- New box: <\b>
- Previous box page number: <\2>
- Current box page number: <\3>
- Next box page number: <\4>
- Conditional Style marker: <\r>

- Zero Width Joiner: <k0.6>
- Zero Width Non-Joiner: <k-0.8>

✦ To define any space or hyphen as non-breaking, use the format: <\!><\!>. Example: <\!><\!-> or <\!><\!m>.

➡ Right Indent tab is defined as <\!><\t>

ASCII codes

Some word processing applications may require you to use the decimal ASCII codes to create XPress Tags for special characters. The XPress Tags code for these characters is <\#decimal value>. The # symbol is part of the code. The XPress Tags code must be three digits for the ASCII code; when entering a four-digit ASCII code, do not enter the leading zero. The following list defines the ASCII decimal codes for some common characters.

- Decimal ASCII code for a character ✦: <\#decimal value>
- New paragraph (Hard return): <\#13>
- Tab: <\#9>
- En dash: <\#208>
- Em dash: <\#209>
- Open double quotation marks: <\#210>
- Close double quotation marks: <\#211>
- Open single quotation mark: <\#212>
- Close single quotation mark (apostrophe): <\#213>

✦ To define any space or hyphen as non-breaking, use the format: <\!><\!>.

Indexing

You can import and export QuarkXPress index tags in XPress Tags format. The text might be tagged by writers or indexers using a word processing application, and then imported into QuarkXPress for layout purposes. Alternatively, the might be exported from QuarkXPress with index tags for editing.

Marking text for an index

The XPress Tags for indexing let you insert index markers at the text insertion point or specify ranges of text to be indexed.

- Insert an index marker at the insertion point: <XI,Tag Info>
- Indicate the start of an indexed range of text: <XO>
- Indicate the end of an indexed range of text: <XC,Tag Info>

Specifying index tag information

Index tags include information about an entry's level, style, and scope.

- First-level entry: "First Level entry","",Sort As,Style Info,Scope,Extra Info,"Cross-Reference String". For example, <XO>20th Century<XC,"20th Century","","Twentieth Century","Index Text 1",6,1,"Modern Age">
- Second-level entry: "First Level entry","Second Level entry","",Sort As,Style Info,Scope,Extra Info,"Cross-Reference String". For example, <XO>Humanities<XC,"20th Century","Humanities","","","Index Text 2",6,1,"Arts">
- Third-level entry: "First Level entry","Second Level entry","Third Level entry","",Sort As,Style Info,Scope,Extra Info,"Cross-Reference String". For example, <XO>Literature<XC,"20th Century","Humanities","Literature","","","Index Text 3",6,1,"Books">
- Fourth-level entry: "First Level entry","Second Level entry","Third Level entry","Fourth Level entry",Sort As,Style Info,Scope,Extra Info,"Cross-Reference String". For example, <XO>English<XC,"20th Century","Humanities","Literature","English","","Index Text 4",6,1,"Great Britain">
- Style information: "style sheet name" or "" for **Entry's Style**
- Scope: 0,1,2,3,4,5 (**Selection Start** = 0, **Selection Text** = 1, **To Style** = 2, **Specified number of paragraphs** = 3, To End Of = 4, **Suppress Page Number** = 5, **Cross-Reference** = code)
- Sort As: “sort as text“
- Selection Start: 0
- Selection Text: 1

- To Style: style sheet name
- Specified # of ¶s: Number of paragraphs
- Suppress Page #: 0
- To End Of: 0,1 (Story = 0, Layout = 1)
- Suppress Page #: 0
- Cross-Reference: 0 1,2 (See = 0, See also = 1, See herein = 2)