

Composite Data Update

For FontCreator 6.5

Although it is a minor version number, FontCreator 5.6 incorporated major changes to the way that composites are composed. The data in CompositeData.xml can now use relative positioning as well as absolute positioning. Positioning of accents is now correct for all typestyles and for fonts with different proportions. Some adjustment will still be needed, but very much less than before. The data can be edited in a plain text editor. Later changes to CompositeData.xml are appended to the end of this document.

New Instructions for CompositeData.xml

<InheritAdvanceWidth>48</InheritAdvanceWidth> Inherits the advance width of glyph 48 (figure zero). Used by currency symbols and mathematical operators.

<InheritLSB>45</InheritLSB> Inherits the left side-bearing from glyph 45 (hyphen).

<InheritRSB>116</InheritRSB> Inherits the right side-bearing from glyph 116 (t).

<InheritSB>65</InheritSB> Inherits the side-bearings from glyph 65 (Capital A).

<LSB>120</LSB> Sets the left side-bearing to 120 funits.

<RSB>44</RSB> Sets the right side-bearing to 44 funits.

<SB>100</SB> Sets both side-bearings to 100 funits.

<AdvanceWidth>1024</AdvanceWidth> Sets the advance width to 1024 funits. Used by accents. Combining diacritics, being non-spacing characters, have zero width.

The above instructions apply to the <Composite> definition,
while the following apply to the composite glyph <Member> definition.

<Member id = "2"> Each glyph member can now have an **id** number for use later in that composite definition.

<Base>2</Base> By default, the first glyph member is assumed to be the base glyph on which the diacritics are positioned. This instruction tells FontCreator to use the second (or another) glyph member as the base glyph. Used for composites like Dž.

<Pos>Auto</Pos> For many composites, this is all that is needed. FontCreator will position most diacritics correctly over or under the base glyph, raising them for uppercase by the difference between CapHeight and x-height. If you're not satisfied with the results, you can manually change the CapHeight on Format, Settings, Ranges, and recompose the uppercase composites. Later you can recalculate the CapHeight.



Design diacritics in the correct vertical position for lowercase.

<Pos>Next</Pos> For most ligatures or diagraphs like IJ, Nj, and Dz, and Alphabetic Presentation Forms like fi, fl, ff, the second glyph is positioned directly after the first, in the position it would normally be without kerning if typed as a separate character. If the designer opts to kern ligatures more tightly, the advance width will need reducing to suit.

<TreatAsUppercase>TRUE</TreatAsUppercase> Tells FontCreator to regard the base glyph as a capital letter, even though it is lowercase. Use by ĥ caron, ě caron, etc.

<TreatAsLowercase>TRUE</TreatAsLowercase> Tells FontCreator to regard the base glyph as lowercase. Used by Capitals with Stacking Diacritics.



Design stacking diacritics in the correct vertical position for uppercase.

<InheritLSB>TRUE</InheritLSB> The composite inherits the left side bearing of the composite glyph member.

<InheritRSB>TRUE</InheritRSB> The composite inherits the right side bearing of the composite glyph member. Used together, these two ensure the correct advance width for fractions and ligatures.

<AlignHorizontal> Aligns the destination glyph horizontally with the source glyph. An offset can be used too. In the illustration, the tilde can be offset from the centre of the "y" by a percentage of the width of the

“y.” Positive values offset the point of alignment to the right. One could also offset the tilde by a percentage of its own width, or by a fixed value.

<Source> Defines the point on the source glyph on which the destination glyph is aligned

<Position> Left, Right, or Center (sic, not Centre) of the base glyph

</Source> Ends the definition of the source glyph

<Dest> Defines the point on the destination glyph used for alignment

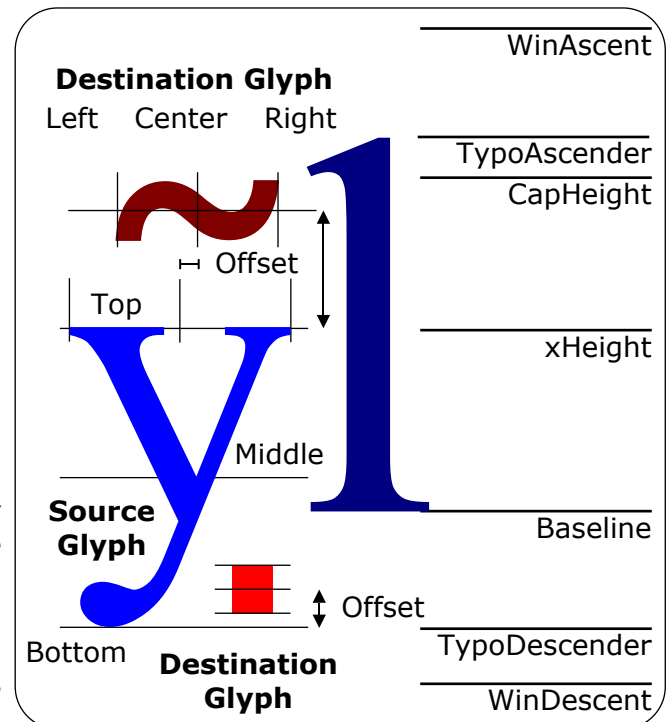
<Position> Left, Right, or Center of the destination glyph that is moved

<Offset>10</Offset> Adjusts the alignment point by a fixed value

Offsets may be a percentage (25%), or defined in units per em (0.80em), and can be negative

</Dest> Ends the definition of the destination glyph

</AlignHorizontal> Ends the definition of one pair of glyphs to be aligned horizontally.



<AlignVertical> Aligns the destination glyph vertically with the source glyph. For vertical alignment, the source may also be any of these font metrics: WinAscent, TypoAscender, CapHeight, xHeight, Baseline, TypoDescender, or WinDescent. Vertical offsets can be a fixed value, a percentage of the glyph's height, a percentage of the font metrics, or expressed in units per em (1em = 2048 funits for many fonts). Positive values offset the point of alignment upwards.

<Source> Defines the point on the source glyph on which the destination glyph is aligned

<Position> Top, Middle, or Bottom of the source or destination glyph; or at the font's CapHeight, xHeight, Baseline, WinAscent, WinDescent, TypoAscender, or TypoDescender

</Source> Ends the definition of the source glyph

<Dest> Defines the point on the destination glyph used for alignment

<Position> Top, Middle, or Bottom of the destination glyph that is moved

<Offset>10</Offset> Adjusts the alignment point by a fixed value

Offsets may be a percentage (e.g. 25%), or negative

</Dest> Ends the definition of the destination glyph

</AlignVertical> Ends the definition of one pair of glyphs to be aligned vertically

Legacy Instructions

<Composite> Start of composite definition **</Composite>** End of composite definition

<!-- Any Text Here --> Comments are used for glyph names

<GlyphMapping> Decimal code-point of composite or glyph member

<Member> Start of composite glyph member definition **</Member>** End of definition

<UseMetrics>TRUE</UseMetrics> Use this glyph's metrics. FALSE is assumed otherwise.

<XPos>100</XPos> Horizontal offset 100 funits to the right (negative values are to the left)

<YPos>500</YPos> Vertical offset 500 funits upwards (negative values are downwards)

<XScale>-1</XScale> Flips the glyph horizontally about the Y-Axis (range ± 1.9999)

<YScale>1.25</YScale> Scales the glyph by 125% vertically (range ± 1.9999)

These instructions were precise, but only for the font used to design the code.

New Instructions

The new instructions make it possible to design composites much more flexibly. We can set bearings for the composite, and align glyph members horizontally or vertically with each other, or vertically with font metrics. We can use offsets from these positions — either a percentage or an absolute value.

Because the CompositeData.xml now uses variables that are set from font metrics, the user can make adjustments by manually changing these font metrics. After the design stage is complete, the correct metrics for the font can be recalculated.

FontCreator can now use substitute composite glyph members. That is, if the first is not available, it will

use another one. For example, if Low Profile accents are available it will use them, but if not, regular accents will be used instead:

The end result is that after using Complete Composites, the user should have much less work to do aligning accents. It will still be necessary to check, but most will be right. If it doesn't work well for your fonts, let us know where the problem lies, and perhaps we can improve the code.

Processing Order for Instructions

- Add glyph members to the composite;
- Set XPos, YPos, XScale and YScale values
- Apply horizontal and vertical alignment
- Apply Next Pos
- Apply Auto Pos
- Apply Inherit Bearings
- Apply UseMetrics

Stacking Diacritics for Vietnamese

The limiting factor is the space available above capital letters. Simply stacking regular accents vertically will increase the line-spacing required to maintain legibility. It is better to reduce the height of the accents and/or stack them side by side. Times New Roman does the former.

In Times New Roman, notice how much bigger the accents are for the lowercase than for capitals. In my view, the latter method, which was used by Victor Gaultney for Gentium, is more attractive. The accents for upper and lowercase are almost the same size, but at a lower angle.

The stacking diacritics are mapped to the Private Use Area from 57344-57364. Use the Insert Characters feature to add these accents, then Complete Composites to compose them. For Vietnamese you will need to add the Combining Hook Above (777) and Combining Horn (795). Complete composites will approximate these accents with the Question Mark and Right Single Quote respectively. Convert them to simple glyphs and modify them to suit the design of the font that you are creating.

To design the combining horn, use the Comparison toolbar to display the lowercase "o" or "u," and display a letter like "t" or "v" to the right. Design the horn to avoid clashes with the letters that follow it.

To design the combining hook, display an "a" before, and an accented vowel afterwards. This will give a guide to the size, weight, and position for the combining hook accent.

Most combining accents will have zero advance width and a negative side-bearing.

Once you have finished designing the accents, run the Vietnamese Transform script to add all of the required characters from the Latin Extended character sets, and the script will also complete the composites. Most of the hard work is done automatically. All you need to do is check that diacritics are in the right position.

Stacking diacritics won't be moved vertically. They are designed to fit between WinAscent and the uppercase vowels.

Superscripts

Super/subscripts should be made heavier to compensate for scaling otherwise they look too light compared to regular numerals. You can use the Glyph Transform Wizard (on the Tools menu) to create them. Click on the folder icon to open the Superscript Transfrom script. This is what it does:

1¹ 2² 3³

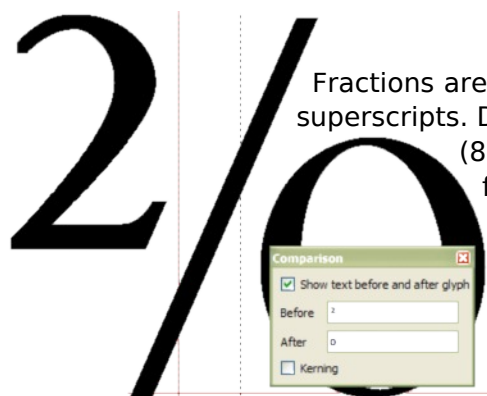
Script: Superscript.xml
 Insert Characters 8304, 185, 178, 179, 8308-8316
 Complete Composites
 Decompose
 Scale (65, 60), LT (about Top Left corner of the glyph).
 Bold (26, 20)
 Move (0, -20)
 Width (fixed 899) Both sides
 Set left side-bearing at x=0

Top alignment should be correct — i.e. at the top of numerals, but the advance width (default 899) must be calculated by scaling the figure width of your font by 65% and adding twice the horizontal weight increase to the overall width.

Subscripts

Subscripts are composed from superscripts. This reduces the workload if you use glyph transform to increase the weight of superscripts as recommended. The subscripts are designed to bisect the baseline.

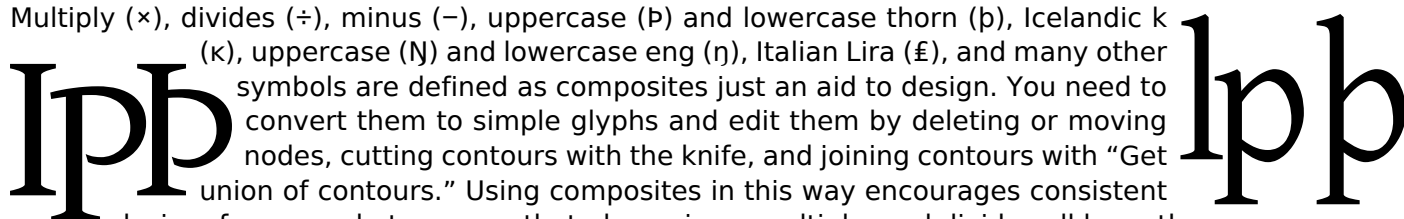
Fractions



Fractions are composed from superscripts. Numerators and denominators are all superscripts. Denominators sit on the baseline or just below it. If the fraction slash (8260) doesn't exist, compose it from forward slash, then skew it forwards. To adjust the spacing of fractions, edit the fraction slash. Display superscript ² and lowercase "o" in the comparison toolbar. Move the glyph and the right side-bearing line until the spacing is just right. Then compose the fractions again. The instruction `<Pos>Next</Pos>` is used for fractions so their width is simply the sum of the widths of each composite glyph member.

Composites Used to Aid Design

Multiply (×), divides (÷), minus (−), uppercase (þ) and lowercase thorn (þ), Icelandic k (κ), uppercase (Ŋ) and lowercase eng (ŋ), Italian Lira (£), and many other symbols are defined as composites just an aid to design. You need to convert them to simple glyphs and edit them by deleting or moving nodes, cutting contours with the knife, and joining contours with "Get union of contours." Using composites in this way encourages consistent design, for example to ensure that plus, minus, multiply, and divides all have the same stroke weights, advance width, and vertical alignment.



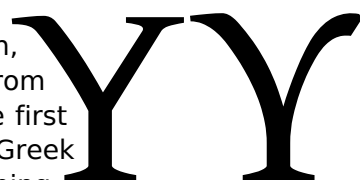
Composites With Stroke

Composites with stroke and currency symbols use stroke overlays (821-823) to ensure a consistent stroke weight. It is often similar to other horizontal strokes in the font. These composites should be made simple before using "Get Union of Contours." If the overlay strokes are missing, the underline glyph (95), which is often too heavy, will be used instead.



Greek and Coptic

Accented characters in this character set are composed from Greek alphabets. Although some fonts use the same letter forms for Greek and Latin, the design may be different. Compare Y and Upsilon, or "v" and "nu" from Gentium (illustrated on the right). Complete composites is no more than the first stage in designing the Greek alphabet. If your font contains the Basic Greek character set, you can generate the full Greek Extended character set by running the Greek Extended Transform script.



Composite data uses accents in the Greek Extended character set. These are composed from Latin accents: tilde, grave, acute, and left/right single quotes. The accents will need to be decomposed and adjusted to be suitable for Greek Extended. The » Gentium « font was used to design the script and composite data. If you look at that, you will see how best to design the Greek diacritics to work well with the Transform script.



Small Capitals and Petite Capitals

These are intended for use with the Small Capitals transform script, which scales them to about 75%, or the Petite Capitals transform script, which scales them to about 60%. Users can adjust the scale in the script to suit the x-height of their font. Small Capitals/Petite Capitals are assigned to the following Private Use code-points:

58033-58126 Basic Latin
58161-58255 Latin-1 Supplement
58256-58383 Latin Extended-A
58384-58591 Latin Extended-B
58912-58974 Basic Greek
59072-59218 Cyrillic
59680-59829 Latin Extended Additional
60531-60542 Number Forms, Nut Fractions or Stacking Fractions
61125-61148 Selected Ligatures from the Medieval Unicode Font Initiative

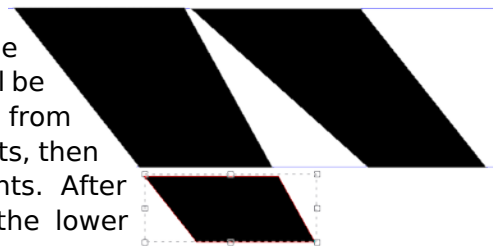
Be mindful that fonts using these mappings will give odd results with scripts that use the same mappings. It is called the “Private” use area because anyone is free to use it as they please. Complete Composites cannot use unmapped glyphs, but composites can.

Old Style Figures

These need to be redesigned for use with lowercase text. The tops should align with the x-height, and the descenders on figures like 3, 5, 7, and 9, should descend below the baseline. They are mapped to the Private Use Area from 58048-58057.

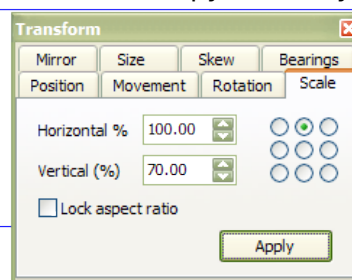
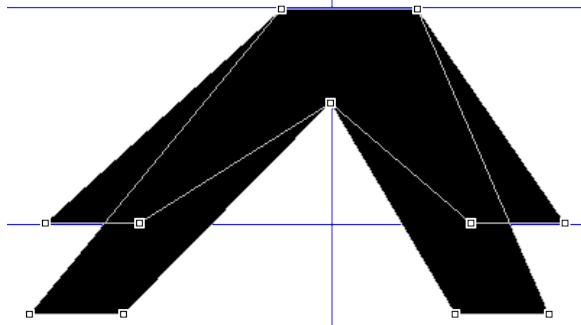
Low Profile Diacritics

If you want to use different sized accents for uppercase, you can define low profile accents. The lowercase composites will use the regular sized accents, but if low profile accents are available they will be used for uppercase. They are mapped to the Private Use Area from 57365-57375. Use the Insert Characters feature to add these accents, then complete composites to compose them from the regular accents. After composing, they must be decomposed and redesigned to suit the lower profile.



Design low profile diacritics in the correct vertical position for lowercase.

Don't just scale the accent vertically. Copy it, scale the copy vertically by about 70%, then cut the original



with the knife to make the low profile version. Some nodes may need to be adjusted to maintain the weight of the accents, as in the low profile circumflex accent. Trimming off the bottom with the knife would make the accent too narrow. Align accents vertically the same as regular lowercase

accents, or bottom aligned instead of middle aligned.

Modifying CompositeData.xml

This is a plain text file that can be opened in any text editor such as notepad or » [Jarte](#) « a free text editor in which one can set the default font for text files, go to line numbers, set bookmarks, etc. After editing CompositeData.xml, you will need to restart Font Creator for the changes to take effect.

The free Garava font from » [Softerviews.org](#) « contains a wide range of Latin Extended characters, Geometric Shapes, and Miscellaneous Symbols. Looking at the font may help you to understand how Complete Composites works as that is the font used while editing and testing the composite data file.



Insert and design all diacritics (accents) first, then use complete composites.

Addendum February 2008

This update makes a few additions and corrections based on my experience of using CompleteComposites while editing my latest fonts. Think of it as a proposal for permanent changes to be incorporated into the next release of FontCreator.

Changes to Existing Definitions

<!-- Low Vertical Line --> (GlyphMapping 716) now uses an offset of -1% from WinDescent. This enables the use of Format, Settings, Metrics, Maximum, Recalculate to have some effect. Several other low glyphs now use the same positioning method. Some composites, e.g. \hat{h} \acute{I} \check{k} \check{h} \check{S} , use a similar offset from WinAscent for the same reason.

<!-- Combining Vertical Line Below --> Now uses vertical line above (GlyphMapping 712) as its source. Only vertical line above needs to use the vertical line as its source. Other vertical line accents now use that as their source. Glyph 712 should be designed to match other accents like acute and caron.

Absolute Positioning (in u diaeresis macron GlyphMapping 470, and elsewhere) has been replaced with relative positioning. `<Pos>Auto</Pos>` cannot be used to position the upper accent vertically. Vertical spacing is now relative to the height of one of the accents.

Solidus Overlays (GlyphMapping 823, 824) now use forward slash.

<!-- Epsilon Tonos --> to **<!-- Omega Tonos -->** Now use `<Pos>Next</Pos>` instead of horizontal positioning. Spacing can be adjusted by changing the right side-bearing of the Tonos accent. Positioning in italic typestyles still needs manual adjustment.

<!-- o tilde acute --> (GlyphMapping 7757) and **<!-- u tilde acute -->** (GlyphMapping 7801) The acute accent is now positioned above the tilde accent by 25% of the acute accent's height instead of 50% of the tilde accent's height.

<!-- Reference Mark --> (GlyphMapping 8251) is now centered vertically on the CapsHeight to give it more space in bold typestyles. The dots are spaced more generously.

<!-- Overline --> (GlyphMapping 8254) Now 20 funits above the TypoAscender instead of 120 funits above the CapHeight.

<!-- Dot plus --> (GlyphMapping 8724) The dot (period centre) is now positioned half a dot height above the plus. This works better for fonts with different weights.

<!-- Parallel --> (GlyphMapping 8741) This no longer uses the metrics of the vertical line, which can be very narrow in some fonts. It now uses horizontal positioning and inherits left and right side-bearings from the vertical line.

<!-- Dot minus --> (GlyphMapping 8760) The dot (period centre) is now positioned one dot height above the minus. This works better for fonts with different weights.

<!-- Tilde operator --> (GlyphMapping 8764) and other maths operators are now middle of xHeight instead of middle of CapHeight.

Corrections

<!-- Combining Horn --> Now uses `<Position>Middle</Position>`, former value `<Position>Center</Position>` was invalid for vertical positioning, and so was just ignored.

<!-- ¶¶ Alphabetic Presentation Forms --> in the Private Use Area have been renamed to **<!-- ¶¶ Miscellaneous Ligatures -->** to avoid confusion with the character set of that name.

<!-- ligature fö --> mapped to code-point 61132 should have been **<!-- ligature fű -->**

<!-- ligature tr --> (GlyphMapping 61145) corrected to (GlyphMapping 61144)

<!-- ligature tt --> (GlyphMapping 61146) corrected to (GlyphMapping 61145)

New Additions

Capital Schwa (GlyphMapping 399): This merely scales lowercase schwa by 1.4. It will need to be decomposed and the weight will need to be reduced to match the font's uppercase.

Greek Capital Theta (GlyphMapping 920), centres Iota on Omicron and scales Omicron vertically.

Composite should be decomposed and the contours combined.

Greek Capital Phi (GlyphMapping 934), centres Iota on Omicron. Composite should be decomposed, the Iota must be rotated and resized by moving some nodes.

Ligatures bb, bg, ck, ey, gg, pp, PP use the mappings proposed by the [Medieval Unicode Font Initiative](#), with the exception of <!-- ligature sp -->, which is mapped to 61143 (proposed by MUFI for “Latin small ligature with flourish.” Ligature sp is far more common. Not knowing where it should go, this seems like the best option. I don’t think many users will make use of these ligatures. I have added ligatures ck and sp to my fonts and to the Miscellaneous Ligatures Glyph Transformation Preset, but I don’t propose to add these other ligatures, as they are only appropriate in Medieval fonts.

<!-- ligature Qu --> (GlyphMapping 61150) and <!-- ligature Th --> (GlyphMapping 61151) These two ligatures are found in Adobe’s fonts. I have used the otherwise unused code-points at the end of the MUFI block for non-structural ligatures. Adobe’s fonts also contain several more ligatures using f (long-s), but I don’t consider them worth adding to my fonts even though some are included in the MUFI proposal. I may add them to CompositeData later. They are: fh, fi, fl, ff, ffi, ffl. Adobe fonts have fb and fk which are not in MUFI. Then fä, fõ, fp, fü, fti, and ftr are in MUFI, but not in Adobe’s fonts.

Not finding any real standards to follow, I have had to improvise. I am happy with this. If you think any of this is worth adding to the next version of FontCreator then at least the groundwork has been done.

Addendum December 2012

Some definitions were added for Unicode versions 6.0 and 6.2, for OpenType feature support, and some improvements were made for Stacking Fractions.

Changes to Existing Definitions

Fallback glyphs were added for many definitions, so that if the preferred glyph doesn’t exist in the font, a second glyph will be used instead if it exists. For example, Petite Capitals will use Low Profile accents if they exist in the font, or regular accents if they do not, or Geometric Shape White Circle will be used to compose Enclosed Alphanumerics, unless Geometric Shape Large Circle exists, which is the preferred glyph.

<InheritSB> is used instead of <InheritAdvanceWidth> for single digit stacking fractions like ¼ to ⅞. For this to work as intended, the divisor should be wider than any single digit, and the divisor for double digit fractions should be included with the denominators for 16ths, 32nds, and 64ths. The double digit fractions still inherit the advance widths from the denominators.

New Additions

German Double S definitions were added in Latin 1 Supplement, and long s with stroke and German Capital S Sharp were added to Latin Extended Additional, and for Petite Capitals in the Private Use Area. A Dieresis Tonos definition was added for Basic Greek.

Definitions were added for the Turkish Lira currency symbol added to Unicode version 6.2, and a few other previously omitted rarely used currency symbols.

Definitions were added for [Titling Capitals](#). These lighter capital letters are designed for use at larger point sizes. If desired, these same code-points can be used for [Small Capitals](#) for fonts that have both [Petite Capitals](#) (x-height) and Small Capitals (larger than x-height) variants of capitals.

Definitions were added for Numerators and Denominators for use with the OpenType [Alternative Fractions](#) feature.

Definitions were added for Historical Ligatures in the Latin Extended D character set.

Heavy quotation marks in the DingBats Character Set can now be composed with Complete Composites.

Definitions for Medieval Ligatures with long s were added to the Private Use Area.

Definitions for miscellaneous Discretionary Ligatures were added to the Private Use Area.