
**Information technology — Coding of
audio-visual objects —**

**Part 28:
Composite font representation**

*Technologies de l'information — Codage des objets audiovisuels —
Partie 28: Représentation de la police de caractères composite*



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2012

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
1 Scope	1
2 Normative references	1
3 Abbreviated terms	1
4 The Composite Font Representation format	2
4.1 Description	2
4.2 Filenames	2
4.3 Syntax and data types.....	2
5 The Composite Font Representation structure.....	2
5.1 High-level overview	2
5.2 The 'PosingFont' element	2
5.3 The 'Name' element	3
5.4 The 'FontMetrics' element	3
5.5 The 'Components' element.....	6
5.6 The 'LanguagePreferredList' element	6
5.7 The 'LanguagePreferredComponentDef' element.....	6
5.8 The 'Language' element.....	6
5.9 The 'ComponentDef' element.....	7
5.10 The 'Tracking' element.....	7
5.11 The 'Matrix' element	7
5.12 The 'UnicodeCharSet' element.....	8
5.13 The 'cmapOverride' element.....	8
5.14 The 'map' element	8
5.15 The 'ToUnicode' element	8
5.16 The 'mapChar' element	9
Annex A (normative) Composite Font Representation Document Type Definition	10
Annex B (informative) Composite Font Representation Examples	13
Bibliography.....	22

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

ISO/IEC 14496-28 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

ISO/IEC 14496 consists of the following parts, under the general title *Information technology — Coding of audio-visual objects*:

- *Part 1: Systems*
- *Part 2: Visual*
- *Part 3: Audio*
- *Part 4: Conformance testing*
- *Part 5: Reference software*
- *Part 6: Delivery Multimedia Integration Framework (DMIF)*
- *Part 7: Optimized reference software for coding of audio-visual objects*
- *Part 8: Carriage of ISO/IEC 14496 contents over IP networks*
- *Part 9: Reference hardware description*
- *Part 10: Advanced Video Coding*
- *Part 11: Scene description and application engine*
- *Part 12: ISO base media file format*
- *Part 13: Intellectual Property Management and Protection (IPMP) extensions*
- *Part 14: MP4 file format*
- *Part 15: Advanced Video Coding (AVC) file format*
- *Part 16: Animation Framework eXtension (AFX)*

- *Part 17: Streaming text format*
- *Part 18: Font compression and streaming*
- *Part 19: Synthesized texture stream*
- *Part 20: Lightweight Application Scene Representation (LASeR) and Simple Aggregation Format (SAF)*
- *Part 21: MPEG-J Graphics Framework eXtensions (GFX)*
- *Part 22: Open Font Format*
- *Part 23: Symbolic Music Representation*
- *Part 24: Audio and systems interaction*
- *Part 25: 3D Graphics Compression Model*
- *Part 26: Audio conformance*
- *Part 27: 3D Graphics conformance*
- *Part 28: Composite font representation*

Information technology — Coding of audio-visual objects —

Part 28: Composite font representation

1 Scope

This part of ISO/IEC 14496 specifies the Composite Font Representation, an XML-based document format that allows combining individual component font resources into a single virtual font. Recent advances in developing the Unicode Standard and the addition of new characters that represent most of the world's languages and writing systems have resulted in a significant increase of the Unicode character repertoire, and this process is likely to continue in the future. Therefore, the ability to combine a number of individual fonts supporting different languages and Unicode ranges in a single virtual font description provides the opportunity for various users and creators of multimedia, graphics and textual content to support all of the world's languages and utilize the existing font rendering solutions that are implemented in current computing platforms and deployed in many existing devices.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

BCP 47 — Tags for Identifying Languages, <<http://tools.ietf.org/html/bcp47>>

Unicode Version 6.1, <<http://www.unicode.org/versions/Unicode6.1.0/>>

3 Abbreviated terms

For the purposes of this document, the following abbreviated terms apply.

CFF	Compact Font Format
CFR	Composite Font Representation
CID	Character ID
GID	Glyph ID
ICU	International Components for Unicode
OFF	Open Font Format
OTF	OpenType Font
UCS	Universal Character Set
UPM	Units Per Em
UTF	Unicode Transformation Format

4 The Composite Font Representation format

4.1 Description

A Composite Font Representation (CFR) resource is an XML representation describing the following essential components:

- Font name – the name(s) used to identify the CFR resource
- Basic metrics for the font – the metrics derived from all of the component fonts. Typically the largest values for a specific metric will be used
- Component list

The component list can specify two kinds of components:

- Single component definitions
- Lists of component definitions based on the order of language preference

A component definition in turn is defined by a font name (a PostScript name that uniquely identifies the component font), along with an optional transformational matrix, optional ICU (International Components for Unicode) UnicodeSet pattern strings defining the Unicode coverage for the component, and an optional character mapping (cmap) that defines how Unicode characters map to glyphs in the component font. The component font must be installed or downloaded for temporary use, and/or registered with the consumer's font system in a way that it can be found via its PostScript name. Note that an optional "LocationHint" attribute may be used as part of the component font definition to specify the location of a component font resource, whether embedded, local, or remote. The ordering of the components is significant, because for elements or attributes that are optional but not specified, their values are derived from the font resource that is referenced by the first instance of a component. Also, the first component containing a particular character mapping is used when resolving overlapping or conflicting character mappings, or when multiple languages are specified.

This format provides the basic framework to develop a composite font resource that is based purely on character ranges and language preferences. It can be used to implement a simple virtual font or a fallback font resource.

4.2 Filenames

CFR resources, when instantiated as a file, may have the extension .cfr.

4.3 Syntax and data types

Refer to the DTD that is provided in Annex A.

5 The Composite Font Representation structure

5.1 High-level overview

A CFR resource consists of a single 'PosingFont' element, and its 'Components' element specifies an arbitrary number of 'ComponentDef' or 'LanguagePreferredList' elements. A CFR resource can be instantiated as a standalone file or resource, and it can be referenced by another CFR resource. Examples of Composite Font Representation are presented in Annex B.

5.2 The 'PosingFont' element

Required. The 'PosingFont' element may contain zero or more 'Name' elements, an optional 'FontMetrics' element, and a single 'Components' element. It also contains two required attributes, 'name' and 'version'.

Attribute definitions:

name = "string"

Required. A string that uniquely identifies the CFR resource, and that follows PostScript font naming conventions.

`version = "string"`

Required. A string that represents the version of the CFR resource.

5.3 The 'Name' element

Optional. Each instance of the 'Name' element specifies an attribute that is equivalent to a specific 'name' table string of an 'sfnt' resource.

Attribute definitions:

`type = "string"`

Required. An sfnt 'name' table string identifier expressed as an integer.

`string = "string"`

Required. The actual content of the 'Name' element.

`language = "string"`

Optional. The BCP 47 tag that corresponds to the language of the 'string' attribute. The script and region can also be specified, if necessary.

5.4 The 'FontMetrics' element

Optional. The 'FontMetrics' element specifies various line-layout attributes that apply globally to the CFR resource.

Attribute definitions:

`unitsPerEm = "string"`

Optional. A virtual UPM (Units Per Em) of the CFR resource expressed as an integer. Its purpose is to facilitate the correct interpretation of the additional attributes. Unless otherwise specified, the values of the additional attributes are based on this UPM value.

`xMin = "string"`

Optional. The minimal X-axis value of the glyph bounding boxes expressed as an integer.

`yMin = "string"`

Optional. The minimal Y-axis value of the glyph bounding boxes expressed as an integer.

`xMax = "string"`

Optional. The maximum X-axis value of the glyph bounding boxes expressed as an integer.

`yMax = "string"`

Optional. The maximum Y-axis value of the glyph bounding boxes expressed as an integer.

`macStyle = "string"`

Optional. The head.macStyle value expressed as a 16-digit bit array.

`lowestRecPPEM = "string"`

Optional. The smallest readable size expressed in number of pixels.

`ascender = "string"`

Optional. The horizontal typographic ascent expressed as an integer.

`descender = "string"`

Optional. The horizontal typographic descent expressed as an integer.

`lineGap = "string"`

Optional. The horizontal typographic line gap expressed as an integer.

`vertTypoAscender = "string"`

Optional. The vertical typographic ascent expressed as an integer.

`vertTypoDescender = "string"`

Optional. The vertical typographic descent expressed as an integer.

`vertTypoLineGap = "string"`

Optional. The vertical typographic line gap expressed as an integer.

`advanceWidthMax = "string"`

Optional. The maximum horizontal advance width expressed as an integer.

`advanceHeightMax = "string"`

Optional. The maximum vertical advance height expressed as an integer.

`minLeftSideBearing = "string"`

Optional. The minimum left sidebearing value expressed as an integer.

`minRightSideBearing = "string"`

Optional. The minimum right sidebearing value, calculated as $(\text{advanceWidthMax} - \text{minLeftSideBearing} - (\text{xMax} - \text{xMin}))$, and expressed as an integer.

`minTopSideBearing = "string"`

Optional. The minimum top sidebearing value expressed as an integer.

`minBottomSideBearing = "string"`

Optional. The minimum bottom sidebearing value expressed as an integer.

`xMaxExtent = "string"`

Optional. The value of the formula $(\text{minLeftSideBearing} + (\text{xMax} - \text{xMin}))$ expressed as an integer.

`yMaxExtent = "string"`

Optional. The value of the formula $(\text{minTopSideBearing} + (\text{yMax} - \text{yMin}))$ expressed as an integer.

`caretSlopeRise = "string"`

Optional. Used to calculate the slope of the cursor according to the formula ($\text{caretSlopeRise} \div \text{caretSlopeRun}$). Use the value 1 for a vertical cursor, which is best for horizontal writing mode.

`caretSlopeRun = "string"`

Optional. Used to calculate the slope of the cursor according to the formula ($\text{caretSlopeRise} \div \text{caretSlopeRun}$). Use the value 0 for a vertical cursor, which is best for horizontal writing mode.

`caretOffset = "string"`

Optional. The amount by which a slanted highlight on a glyph needs to be shifted to produce the best appearance. Use the value 0 for non-slanted fonts.

`vertCaretSlopeRise = "string"`

Optional. Used to calculate the slope of the cursor according to the formula ($\text{vertCaretSlopeRise} \div \text{vertCaretSlopeRun}$). Use the value 0 for a horizontal cursor, which is best for vertical writing mode.

`vertCaretSlopeRun = "string"`

Optional. Used to calculate the slope of the cursor according to the formula ($\text{vertCaretSlopeRise} \div \text{vertCaretSlopeRun}$). Use the value 1 for a horizontal cursor, which is best for vertical writing mode.

`vertCaretOffset = "string"`

Optional. The amount by which a slanted highlight on a glyph needs to be shifted to produce the best appearance. Use the value 0 for non-slanted fonts.

`italicAngle = "string"`

Optional. The italic angle in counter-clockwise degrees from the vertical expressed as an integer. Upright text should be set to 0, and text that leans to the right should be set to a negative value.

`underlinePosition = "string"`

Optional. The top of the underline expressed as an integer. If it is below the baseline, a negative value should be used.

`underlineThickness = "string"`

Optional. The suggested value for the underline thickness expressed as an integer.

`isFixedPitch = "string"`

Optional. A value that specifies whether the glyphs in the CFR resource are monospaced (non-zero) or proportional (zero).

`numGlyphs = "string"`

Optional. The implied maximum number of glyphs in the CFR resource, which is intended to be used for the purpose of optimization.

`weightClass = "string"`

Optional. The visual weight of the glyphs in the CFR resource expressed as an integer.

`widthClass = "string"`

Optional. The relative aspect ratio of the glyphs in the CFR resource expressed as an integer.

familyClass = "string"

Optional. The IBM Font Class and the IBM Font Subclass parameter values of the CFR resource expressed as an integer.

5.5 The 'Components' element

Required. The 'Components' element contains one or more of either the 'ComponentDef' or 'LanguagePreferredList' elements.

5.6 The 'LanguagePreferredList' element

The 'LanguagePreferredList' element contains one or more 'LanguagePreferredComponentDef' elements.

5.7 The 'LanguagePreferredComponentDef' element

Required. The 'LanguagePreferredComponentDef' element contains a single 'Language' element and one or more 'ComponentDef' elements. If the same 'ComponentDef' element is meant to serve more than one language, multiple instances of the 'LanguagePreferredComponentDef' element may be specified.

The 'LanguagePreferredList' makes most sense if there is more than one 'LanguagePreferredComponentDef' element specified, though it is not an error if it specifies only one 'LanguagePreferredComponentDef' element.

The user's language setting determines which 'LanguagePreferredComponentDef' element to use first, and when a character that is not supported by the font specified by that 'LanguagePreferredComponentDef' element is encountered, the ordering of the 'LanguagePreferredComponentDef' elements is significant, and the first 'LanguagePreferredComponentDef' element that supports the character is used.

Furthermore, the 'LanguagePreferredComponentDef' elements in the 'LanguagePreferredList' element can be reordered according to the language preference settings of the user's system or environment. For example, a CFR resource can specify a 'LanguagePreferredList' element such as the following:

```
<LanguagePreferredList>
<LanguagePreferredComponentDef <ComponentDef name="Tokyo">, language="ja">
<LanguagePreferredComponentDef <ComponentDef name="Taipei">, language="zh-Hant">
<LanguagePreferredComponentDef <ComponentDef name="Beijing">, language="zh-Hans">
</LanguagePreferredList>
```

By default, the order of the component fonts is "Tokyo," "Taipei," then "Beijing." This may be suitable for a typical Japanese environment, but for a user in Shanghai, the preferred language would be Simplified Chinese, and the preferred order of the component fonts should be "Beijing," "Taipei," then "Tokyo," because the user's language preference is likely to be Simplified Chinese first, Traditional Chinese second, and perhaps followed by no other language preference. This reordering within the 'LanguagePreferredList' element takes place in applications or environment for which users' language settings are applicable, otherwise the order of the 'LanguagePreferredComponentDef' elements should be respected.

As described above, 'LanguagePreferredComponentDef' elements can be reordered by the consumer, but consumers cannot remove nor add them.

5.8 The 'Language' element

Required. The 'Language' element specifies the language that is to be applied to the 'ComponentDef' elements of the 'LanguagePreferredComponentDef' element.

Attribute definitions:

string = "string"

Required. The BCP 47 tag that corresponds to the language of the 'Language' element. In some cases, such as to distinguish Simplified and Traditional Chinese, script and/or region codes may be appended, following a hyphen, such as zh-Hans for Simplified Chinese or zh-Hant for Traditional Chinese.

5.9 The 'ComponentDef' element

Required. The 'ComponentDef' element may contain one 'Tracking' element, one 'Matrix' element, any number of 'UnicodeCharSet' elements, one 'cmapOverride' element and one 'ToUnicode' element.

Attribute definitions:

name = "string"

Required. The PostScript name of the font resource specified by the 'ComponentDef' element. For CIDFont resources, the CIDFontName string should be used. For recursive use of CFR resources, the 'PosingFont' element's 'name' attribute should be used.

locationHint = "string"

Optional. The location of the font resource described in various ways, such as a URL or local system path. The value of this attribute is not limited to being a URL or local system path. If the value cannot be resolved, the consumer is to default to its standard font resource locating mechanism.

5.10 The 'Tracking' element

Optional. The 'Tracking' element is used to specify tracking that is applied to the 'ComponentDef' element.

Attribute definitions:

trackingValue = "string"

Required. A tracking value expressed as a negative or positive percentage.

5.11 The 'Matrix' element

Optional. The 'Matrix' element specifies a six-element transformational matrix that is applied to the 'ComponentDef' element. This is the same 2×3 transformation matrix as used by the PostScript imaging model.

Attribute definitions:

xx = "string"

Required. The first element of the six-element transformational matrix.

xy = "string"

Required. The second element of the six-element transformational matrix.

yx = "string"

Required. The third element of the six-element transformational matrix.

yy = "string"

Required. The fourth element of the six-element transformational matrix.

tx = "string"

Required. The fifth element of the six-element transformational matrix.

ty = "string"

Required. The sixth element of the six-element transformational matrix.

5.12 The 'UnicodeCharSet' element

Optional. The 'UnicodeCharSet' element specifies the Unicode code points or code point ranges for which the 'ComponentDef' element is to be used.

Attribute definitions:

uset = "string"

Required. A series of Unicode code points or code point ranges that are specified according to ICU's UnicodeSet pattern syntax. As an example, the ranges U+0020 through U+007E and U+4E00 through U+9FCC can thus be expressed as [[\u0020-\u007E] [\u4E00-\u9FCC]].

5.13 The 'cmapOverride' element

Optional. The 'cmapOverride' element contains one or more 'map' elements.

The 'cmapOverride' element provides a mechanism to override the 'cmap' table of the font resource specified by the 'ComponentDef' element. The resulting font will have a character set that is the intersection of the 'cmapOverride' element and the actual font resource. The consumer must honor this element in order to determine whether the component font contains a glyph for a referenced character. It is considered an exception if the font fails to supply a glyph for its advertised character set, but this is true for any font resource.

5.14 The 'map' element

Required. The 'map' element maps between a glyph and a Unicode code point or Unicode code point sequence.

Attribute definitions:

charValue = "string"

Required. A Unicode code point or Unicode code point sequence that is specified with a notation that uses Unicode scalar values, substituting the "U+" prefix with a "u" prefix, such as "u4E00" for U+4E00.

charName = "string"

Optional. The Unicode name of the Unicode code point. If there is no corresponding Unicode name, an arbitrary name can be specified.

glyphRefID = "string"

Required. The GID (Glyph ID) of the glyph. For purposes of efficiency, even for CID-keyed fonts, this still refers to the GID.

glyphName = "string"

Optional. A descriptive name of the glyph. For most purposes, naming glyphs according to the AGL Specification [1] is recommended.

5.15 The 'ToUnicode' element

Optional. The 'ToUnicode' element contains one or more 'mapChar' elements.

Attribute definitions:

fromEncoding = "string"

Optional. The name of the encoding from which the 'ToUnicode' element transcodes to Unicode. Do not use if the encoding is Unicode.

5.16 The 'mapChar' element

Required. The 'mapChar' element maps from arbitrary character codes or arbitrary character code sequences, including Unicode, to Unicode.

Attribute definitions:

`fromCharValue = "string"`

Required. An arbitrary character code or character code sequence that is specified by the '0x' prefix followed by multiples of two hexadecimal digits whereby each two hexadecimal digits represent a single byte. If a Unicode code point or Unicode code point sequence is used, it should be specified according to ICU's UnicodeSet pattern syntax.

`toCharValue = "string"`

Required. A Unicode code point or Unicode code point sequence that is specified with a notation that uses Unicode scalar values, substituting the "U+" prefix with a "u" prefix, such as "u4E00" for U+4E00.

Annex A (normative)

Composite Font Representation Document Type Definition

A.1 Introduction

This Annex defines the DTD for the Composite Font Representation.

A.2 Composite Font Representation DTD

```
<!-- Posing Font - name:PostScript name -->
<!ELEMENT PosingFont (Name*, FontMetrics?, Components)>
<!ATTLIST PosingFont
    name CDATA #REQUIRED
    version CDATA #REQUIRED
>

<!-- Name - language:BCP 47 language tag -->
<!ELEMENT Name EMPTY >
<!ATTLIST Name
    type CDATA #REQUIRED
    string CDATA #REQUIRED
    language CDATA #IMPLIED
>

<!-- Font Metrics -->
<!ELEMENT FontMetrics EMPTY >
<!ATTLIST FontMetrics
    unitsPerEm CDATA #IMPLIED
    xMin CDATA #IMPLIED
    yMin CDATA #IMPLIED
    xMax CDATA #IMPLIED
    yMax CDATA #IMPLIED
    macStyle CDATA #IMPLIED
    lowestRecPPEM CDATA #IMPLIED
    ascender CDATA #IMPLIED
    descender CDATA #IMPLIED
    lineGap CDATA #IMPLIED
    vertTypoAscender CDATA #IMPLIED
    vertTypoDescender CDATA #IMPLIED
    vertTypoLineGap CDATA #IMPLIED
    advanceWidthMax CDATA #IMPLIED
    advanceHeightMax CDATA #IMPLIED
    minLeftSideBearing CDATA #IMPLIED
    minRightSideBearing CDATA #IMPLIED
    minTopSideBearing CDATA #IMPLIED
    minBottomSideBearing CDATA #IMPLIED
    xMaxExtent CDATA #IMPLIED
    yMaxExtent CDATA #IMPLIED
    caretSlopeRise CDATA #IMPLIED
    caretSlopeRun CDATA #IMPLIED
    caretOffset CDATA #IMPLIED
    vertCaretSlopeRise CDATA #IMPLIED
    vertCaretSlopeRun CDATA #IMPLIED
    vertCaretOffset CDATA #IMPLIED
    italicAngle CDATA #IMPLIED
    underlinePosition CDATA #IMPLIED
    underlineThickness CDATA #IMPLIED
    isFixedPitch CDATA #IMPLIED
```



```

        numGlyphs CDATA #IMPLIED
        weightClass CDATA #IMPLIED
        widthClass CDATA #IMPLIED
        familyClass CDATA #IMPLIED
    >

<!-- Components List -->
<!ELEMENT Components (ComponentDef | LanguagePreferredList)+>

<!-- Component Font Definition - name:PostScript name -->
<!ELEMENT ComponentDef (Tracking?, Matrix?, UnicodeCharSet*, cmapOverride?,
ToUnicode?)+>
<!-- ATTLIST ComponentDef
    name CDATA #REQUIRED
    locationHint CDATA #IMPLIED
-->

<!-- List of components by Language Preferred -->
<!ELEMENT LanguagePreferredList (LanguagePreferredComponentDef)+>

<!-- Component font definition for a specific language -->
<!ELEMENT LanguagePreferredComponentDef (Language, ComponentDef)+>

<!-- Language - string:BCP 47 language tag -->
<!ELEMENT Language EMPTY >
<!-- ATTLIST Language
    string CDATA #REQUIRED
-->

<!-- Tracking -->
<!ELEMENT Tracking EMPTY >
<!-- ATTLIST Tracking
    trackingValue CDATA #REQUIRED
-->

<!-- Matrix -->
<!ELEMENT Matrix EMPTY >
<!-- ATTLIST Matrix
    xx CDATA #REQUIRED
    xy CDATA #REQUIRED
    yx CDATA #REQUIRED
    yy CDATA #REQUIRED
    tx CDATA #REQUIRED
    ty CDATA #REQUIRED
-->

<!-- CharSet - uset:ICU UnicodeSet pattern string -->
<!ELEMENT UnicodeCharSet EMPTY >
<!-- ATTLIST UnicodeCharSet
    uset CDATA #REQUIRED
-->

<!-- cmap for components glyph ranges -->
<!ELEMENT cmapOverride (map+)>

<!-- ELEMENT map EMPTY -->
<!-- ATTLIST map
    charValue CDATA #REQUIRED
    charName CDATA #IMPLIED
    glyphRefID CDATA #REQUIRED
    glyphName CDATA #IMPLIED
-->

<!-- Transcoding to Unicode -->
<!-- ELEMENT ToUnicode (mapChar+)>
<!-- ATTLIST ToUnicode
    fromEncoding CDATA #IMPLIED
-->

```

```
<!ELEMENT mapChar EMPTY>
<!--ATTLIST mapChar
      fromCharValue CDATA #REQUIRED
      toCharValue CDATA #REQUIRED
-->
```

Annex B (informative)

Composite Font Representation Examples

B.1 Example 1

Assume we have a Japanese font called TokyoSans. This font's Latin glyphs are not ideal, and we would like to create a new version of this font using the Latin glyphs from another font called MonteCarloSans. We can do so by defining a CFR resource called TokyoCarloSans with 2 components. The first component is MonteCarloSans. We take the Latin glyphs from MonteCarloSans by defining a UnicodeCharSet for the Latin characters. We also define a transformational matrix that we use to match the baseline of TokyoSans.

The second component font is TokyoSans. We do not specify a character set or anything else because we wish to use all of the glyphs present in TokyoSans.

Note that the Latin glyphs will come from MonteCarloSans because it was the first component font specified.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PosingFont SYSTEM
"file://localhost/System/Library/DTDs/SplicedFont.dtd">
<PosingFont name="TokyoCarloSans" version="1.0">
  <Name type="1" string="Tokyo Sans Monospaced" language="en"/>
  <Name type="2" string="Regular" language="en"/>
  <Name type="3" string="FakeTokyoTypeworks Ltd." language="en"/>
  <Name type="4" string="Tokyo Sans Monospaced Regular"/>
  <FontMetrics
    unitsPerEm="1000"
    ascender="561.1"
    descender="214.3"
    lineGap="21.0"
    italicAngle="0.0"
    isFixedPitch="1"
  />
  <Components>
    <ComponentDef name="MonteCarloSans">
      <Matrix
        xx="1.0"
        xy="0.0"
        yx="0.0"
        yy="1.0"
        tx="0.0"
        ty="-6.4"
      />
      <UnicodeCharSet
        uset="[[\u0020-\u007E][\u00A0-\u00FC][\u02BB-\u0336][\u2010-
\u2044][\u2212-\u223C]]"
      />
    </ComponentDef>
    <ComponentDef name="TokyoSans">
    </ComponentDef>
  </Components>
</PosingFont>
```

B.2 Example 2

Assume we have a Japanese font called TokyoSans. This font has monospaced and proportional Latin glyphs. We want to create a CFR resource based on this font that overrides the default proportional Latin glyphs with the monospaced ones. Note that the two component fonts use the same base font, TokyoSans. The first component font specifies a cmap override for the Latin characters to use to the monospaced glyphs in the font.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PosingFont SYSTEM
"file://localhost/System/Library/DTDs/SplicedFont.dtd">
<PosingFont name="TokyoSansMono" version="1.0">
  <Name type="1" string="Tokyo Sans Monospaced" language="en"/>
  <Name type="2" string="Regular" language="en"/>
  <Name type="3" string="FakeTokyoTypeworks Ltd." language="en"/>
  <Name type="4" string="Tokyo Sans Monospaced Regular"/>
  <FontMetrics
    unitsPerEm="1000"
    ascender="561.1"
    descender="214.3"
    lineGap="21.0"
    italicAngle="0.0"
    isFixedPitch="1"
  />
  <Components>
    <ComponentDef name="TokyoSans">
      <UnicodeCharSet
        uset="[[\u0020-\u007E][\u00A0-\u00FC][\u02BB-\u0336][\u2010-
\u2044][\u2212-\u223C]]"
      />
      <cmapOverride>
        <map charValue="u0020" charName="SPACE" glyphRefID="231"/>
        <map charValue="u0021" charName="EXCLAMATION MARK"
glyphRefID="232"/>
        <map charValue="u0022" charName="QUOTATION MARK"
glyphRefID="12087"/>
        <map charValue="u0023" charName="NUMBER SIGN" glyphRefID="234"/>
        <map charValue="u0024" charName="DOLLAR SIGN" glyphRefID="235"/>
        <map charValue="u0025" charName="PERCENT SIGN" glyphRefID="236"/>
        <map charValue="u0026" charName="AMPERSAND" glyphRefID="237"/>
        <map charValue="u0027" charName="APOSTROPHE" glyphRefID="12086"/>
        <map charValue="u0028" charName="LEFT PARENTHESIS"
glyphRefID="239"/>
        <map charValue="u0029" charName="RIGHT PARENTHESIS"
glyphRefID="240"/>
        <map charValue="u002A" charName="ASTERISK" glyphRefID="241"/>
        <map charValue="u002B" charName="PLUS SIGN" glyphRefID="242"/>
        <map charValue="u002C" charName="COMMA" glyphRefID="243"/>
        <map charValue="u002D" charName="HYPHEN-MINUS" glyphRefID="244"/>
        <map charValue="u002E" charName="FULL STOP" glyphRefID="245"/>
        <map charValue="u002F" charName="SOLIDUS" glyphRefID="246"/>
        <map charValue="u0030" charName="DIGIT ZERO" glyphRefID="247"/>
        <map charValue="u0031" charName="DIGIT ONE" glyphRefID="248"/>
        <map charValue="u0032" charName="DIGIT TWO" glyphRefID="249"/>
        <map charValue="u0033" charName="DIGIT THREE" glyphRefID="250"/>
        <map charValue="u0034" charName="DIGIT FOUR" glyphRefID="251"/>
        <map charValue="u0035" charName="DIGIT FIVE" glyphRefID="252"/>
        <map charValue="u0036" charName="DIGIT SIX" glyphRefID="253"/>
        <map charValue="u0037" charName="DIGIT SEVEN" glyphRefID="254"/>
        <map charValue="u0038" charName="DIGIT EIGHT" glyphRefID="255"/>
        <map charValue="u0039" charName="DIGIT NINE" glyphRefID="256"/>
      </cmapOverride>
    </ComponentDef>
  </Components>
</PosingFont>
```

```

    <map charValue="u003A" charName="COLON" glyphRefID="257"/>
    <map charValue="u003B" charName="SEMICOLON" glyphRefID="258"/>
    <map charValue="u003C" charName="LESS-THAN SIGN" glyphRefID="259"/>
    <map charValue="u003D" charName="EQUALS SIGN" glyphRefID="260"/>
    <map charValue="u003E" charName="GREATER-THAN SIGN"
glyphRefID="261"/>
    <map charValue="u003F" charName="QUESTION MARK" glyphRefID="262"/>
    <map charValue="u0040" charName="COMMERCIAL AT" glyphRefID="263"/>
    <map charValue="u0041" charName="LATIN CAPITAL LETTER A"
glyphRefID="264"/>
    <map charValue="u0042" charName="LATIN CAPITAL LETTER B"
glyphRefID="265"/>
    <map charValue="u0043" charName="LATIN CAPITAL LETTER C"
glyphRefID="266"/>
    <map charValue="u0044" charName="LATIN CAPITAL LETTER D"
glyphRefID="267"/>
    <map charValue="u0045" charName="LATIN CAPITAL LETTER E"
glyphRefID="268"/>
    <map charValue="u0046" charName="LATIN CAPITAL LETTER F"
glyphRefID="269"/>
    <map charValue="u0047" charName="LATIN CAPITAL LETTER G"
glyphRefID="270"/>
    <map charValue="u0048" charName="LATIN CAPITAL LETTER H"
glyphRefID="271"/>
    <map charValue="u0049" charName="LATIN CAPITAL LETTER I"
glyphRefID="272"/>
    <map charValue="u004A" charName="LATIN CAPITAL LETTER J"
glyphRefID="273"/>
    <map charValue="u004B" charName="LATIN CAPITAL LETTER K"
glyphRefID="274"/>
    <map charValue="u004C" charName="LATIN CAPITAL LETTER L"
glyphRefID="275"/>
    <map charValue="u004D" charName="LATIN CAPITAL LETTER M"
glyphRefID="276"/>
    <map charValue="u004E" charName="LATIN CAPITAL LETTER N"
glyphRefID="277"/>
    <map charValue="u004F" charName="LATIN CAPITAL LETTER O"
glyphRefID="278"/>
    <map charValue="u0050" charName="LATIN CAPITAL LETTER P"
glyphRefID="279"/>
    <map charValue="u0051" charName="LATIN CAPITAL LETTER Q"
glyphRefID="280"/>
    <map charValue="u0052" charName="LATIN CAPITAL LETTER R"
glyphRefID="281"/>
    <map charValue="u0053" charName="LATIN CAPITAL LETTER S"
glyphRefID="282"/>
    <map charValue="u0054" charName="LATIN CAPITAL LETTER T"
glyphRefID="283"/>
    <map charValue="u0055" charName="LATIN CAPITAL LETTER U"
glyphRefID="284"/>
    <map charValue="u0056" charName="LATIN CAPITAL LETTER V"
glyphRefID="285"/>
    <map charValue="u0057" charName="LATIN CAPITAL LETTER W"
glyphRefID="286"/>
    <map charValue="u0058" charName="LATIN CAPITAL LETTER X"
glyphRefID="287"/>
    <map charValue="u0059" charName="LATIN CAPITAL LETTER Y"
glyphRefID="288"/>
    <map charValue="u005A" charName="LATIN CAPITAL LETTER Z"
glyphRefID="289"/>

```

```

        <map charValue="u005B" charName="LEFT SQUARE BRACKET"
glyphRefID="290"/>
        <map charValue="u005C" charName="REVERSE SOLIDUS"
glyphRefID="8719"/>
        <map charValue="u005D" charName="RIGHT SQUARE BRACKET"
glyphRefID="292"/>
        <map charValue="u005E" charName="CIRCUMFLEX ACCENT"
glyphRefID="293"/>
        <map charValue="u005F" charName="LOW LINE" glyphRefID="294"/>
        <map charValue="u0060" charName="GRAVE ACCENT" glyphRefID="390"/>
        <map charValue="u0061" charName="LATIN SMALL LETTER A"
glyphRefID="296"/>
        <map charValue="u0062" charName="LATIN SMALL LETTER B"
glyphRefID="297"/>
        <map charValue="u0063" charName="LATIN SMALL LETTER C"
glyphRefID="298"/>
        <map charValue="u0064" charName="LATIN SMALL LETTER D"
glyphRefID="299"/>
        <map charValue="u0065" charName="LATIN SMALL LETTER E"
glyphRefID="300"/>
        <map charValue="u0066" charName="LATIN SMALL LETTER F"
glyphRefID="301"/>
        <map charValue="u0067" charName="LATIN SMALL LETTER G"
glyphRefID="302"/>
        <map charValue="u0068" charName="LATIN SMALL LETTER H"
glyphRefID="303"/>
        <map charValue="u0069" charName="LATIN SMALL LETTER I"
glyphRefID="304"/>
        <map charValue="u006A" charName="LATIN SMALL LETTER J"
glyphRefID="305"/>
        <map charValue="u006B" charName="LATIN SMALL LETTER K"
glyphRefID="306"/>
        <map charValue="u006C" charName="LATIN SMALL LETTER L"
glyphRefID="307"/>
        <map charValue="u006D" charName="LATIN SMALL LETTER M"
glyphRefID="308"/>
        <map charValue="u006E" charName="LATIN SMALL LETTER N"
glyphRefID="309"/>
        <map charValue="u006F" charName="LATIN SMALL LETTER O"
glyphRefID="310"/>
        <map charValue="u0070" charName="LATIN SMALL LETTER P"
glyphRefID="311"/>
        <map charValue="u0071" charName="LATIN SMALL LETTER Q"
glyphRefID="312"/>
        <map charValue="u0072" charName="LATIN SMALL LETTER R"
glyphRefID="313"/>
        <map charValue="u0073" charName="LATIN SMALL LETTER S"
glyphRefID="314"/>
        <map charValue="u0074" charName="LATIN SMALL LETTER T"
glyphRefID="315"/>
        <map charValue="u0075" charName="LATIN SMALL LETTER U"
glyphRefID="316"/>
        <map charValue="u0076" charName="LATIN SMALL LETTER V"
glyphRefID="317"/>
        <map charValue="u0077" charName="LATIN SMALL LETTER W"
glyphRefID="318"/>
        <map charValue="u0078" charName="LATIN SMALL LETTER X"
glyphRefID="319"/>
        <map charValue="u0079" charName="LATIN SMALL LETTER Y"
glyphRefID="320"/>

```

```

        <map charValue="u007A" charName="LATIN SMALL LETTER Z"
glyphRefID="321"/>
        <map charValue="u007B" charName="LEFT CURLY BRACKET"
glyphRefID="322"/>
        <map charValue="u007C" charName="VERTICAL LINE" glyphRefID="323"/>
        <map charValue="u007D" charName="RIGHT CURLY BRACKET"
glyphRefID="324"/>
        <map charValue="u007E" charName="TILDE" glyphRefID="631"/>
        <map charValue="u00A0" charName="NO-BREAK SPACE" glyphRefID="231"/>
        <map charValue="u00A1" charName="INVERTED EXCLAMATION MARK"
glyphRefID="612"/>
        <map charValue="u00A2" charName="CENT SIGN" glyphRefID="608"/>
        <map charValue="u00A3" charName="POUND SIGN" glyphRefID="609"/>
        <map charValue="u00A5" charName="YEN SIGN" glyphRefID="291"/>
        <map charValue="u00A6" charName="BROKEN BAR" glyphRefID="323"/>
        <map charValue="u00A8" charName="DIAERESIS" glyphRefID="502"/>
        <map charValue="u00AC" charName="NOT SIGN" glyphRefID="8718"/>
        <map charValue="u00AD" charName="SOFT HYPHEN" glyphRefID="514"/>
        <map charValue="u00AF" charName="MACRON" glyphRefID="325"/>
        <map charValue="u00B4" charName="ACUTE ACCENT" glyphRefID="501"/>
        <map charValue="u00B7" charName="MIDDLE DOT" glyphRefID="331"/>
        <map charValue="u00BD" charName="VULGAR FRACTION ONE HALF"
glyphRefID="614"/>
        <map charValue="u00BF" charName="INVERTED QUESTION MARK"
glyphRefID="613"/>
        <map charValue="u00C4" charName="LATIN CAPITAL LETTER A WITH
DIAERESIS" glyphRefID="599"/>
        <map charValue="u00C7" charName="LATIN CAPITAL LETTER C WITH
CEDILLA" glyphRefID="605"/>
        <map charValue="u00D1" charName="LATIN CAPITAL LETTER N WITH TILDE"
glyphRefID="606"/>
        <map charValue="u00D6" charName="LATIN CAPITAL LETTER O WITH
DIAERESIS" glyphRefID="615"/>
        <map charValue="u00DC" charName="LATIN CAPITAL LETTER U WITH
DIAERESIS" glyphRefID="616"/>
        <map charValue="u00DF" charName="LATIN SMALL LETTER SHARP S"
glyphRefID="603"/>
        <map charValue="u00E0" charName="LATIN SMALL LETTER A WITH GRAVE"
glyphRefID="627"/>
        <map charValue="u00E1" charName="LATIN SMALL LETTER A WITH ACUTE"
glyphRefID="630"/>
        <map charValue="u00E2" charName="LATIN SMALL LETTER A WITH
CIRCUMFLEX" glyphRefID="622"/>
        <map charValue="u00E4" charName="LATIN SMALL LETTER A WITH
DIAERESIS" glyphRefID="617"/>
        <map charValue="u00E7" charName="LATIN SMALL LETTER C WITH CEDILLA"
glyphRefID="604"/>
        <map charValue="u00E8" charName="LATIN SMALL LETTER E WITH GRAVE"
glyphRefID="628"/>
        <map charValue="u00E9" charName="LATIN SMALL LETTER E WITH ACUTE"
glyphRefID="601"/>
        <map charValue="u00EA" charName="LATIN SMALL LETTER E WITH
CIRCUMFLEX" glyphRefID="623"/>
        <map charValue="u00EB" charName="LATIN SMALL LETTER E WITH
DIAERESIS" glyphRefID="618"/>
        <map charValue="u00ED" charName="LATIN SMALL LETTER I WITH ACUTE"
glyphRefID="602"/>
        <map charValue="u00EE" charName="LATIN SMALL LETTER I WITH
CIRCUMFLEX" glyphRefID="624"/>
        <map charValue="u00EF" charName="LATIN SMALL LETTER I WITH

```

```

DIAERESIS" glyphRefID="619"/>
    <map charValue="u00F1" charName="LATIN SMALL LETTER N WITH TILDE"
glyphRefID="607"/>
    <map charValue="u00F3" charName="LATIN SMALL LETTER O WITH ACUTE"
glyphRefID="610"/>
    <map charValue="u00F4" charName="LATIN SMALL LETTER O WITH
CIRCUMFLEX" glyphRefID="625"/>
    <map charValue="u00F6" charName="LATIN SMALL LETTER O WITH
DIAERESIS" glyphRefID="620"/>
    <map charValue="u00F9" charName="LATIN SMALL LETTER U WITH GRAVE"
glyphRefID="600"/>
    <map charValue="u00FA" charName="LATIN SMALL LETTER U WITH ACUTE"
glyphRefID="611"/>
    <map charValue="u00FB" charName="LATIN SMALL LETTER U WITH
CIRCUMFLEX" glyphRefID="626"/>
    <map charValue="u00FC" charName="LATIN SMALL LETTER U WITH
DIAERESIS" glyphRefID="621"/>
    <map charValue="u02BB" charName="MODIFIER LETTER TURNED COMMA"
glyphRefID="295"/>
    <map charValue="u02BC" charName="MODIFIER LETTER APOSTROPHE"
glyphRefID="238"/>
    <map charValue="u02DC" charName="SMALL TILDE" glyphRefID="631"/>
    <map charValue="u0300" charName="COMBINING GRAVE ACCENT"
glyphRefID="390"/>
    <map charValue="u0301" charName="COMBINING ACUTE ACCENT"
glyphRefID="501"/>
    <map charValue="u0302" charName="COMBINING CIRCUMFLEX ACCENT"
glyphRefID="293"/>
    <map charValue="u0303" charName="COMBINING TILDE"
glyphRefID="631"/>
    <map charValue="u0304" charName="COMBINING MACRON"
glyphRefID="325"/>
    <map charValue="u0305" charName="COMBINING OVERLINE"
glyphRefID="325"/>
    <map charValue="u0308" charName="COMBINING DIAERESIS"
glyphRefID="502"/>
    <map charValue="u0332" charName="COMBINING LOW LINE"
glyphRefID="294"/>
    <map charValue="u0336" charName="COMBINING LONG STROKE OVERLAY"
glyphRefID="514"/>
    <map charValue="u2010" charName="HYPHEN" glyphRefID="244"/>
    <map charValue="u2011" charName="NON-BREAKING HYPHEN"
glyphRefID="244"/>
    <map charValue="u2012" charName="FIGURE DASH" glyphRefID="244"/>
    <map charValue="u2013" charName="EN DASH" glyphRefID="514"/>
    <map charValue="u2014" charName="EM DASH" glyphRefID="514"/>
    <map charValue="u2018" charName="LEFT SINGLE QUOTATION MARK"
glyphRefID="295"/>
    <map charValue="u2019" charName="RIGHT SINGLE QUOTATION MARK"
glyphRefID="238"/>
    <map charValue="u201C" charName="LEFT DOUBLE QUOTATION MARK"
glyphRefID="503"/>
    <map charValue="u201D" charName="RIGHT DOUBLE QUOTATION MARK"
glyphRefID="233"/>
    <map charValue="u2039" charName="SINGLE LEFT-POINTING ANGLE
QUOTATION MARK" glyphRefID="259"/>
    <map charValue="u203A" charName="SINGLE RIGHT-POINTING ANGLE
QUOTATION MARK" glyphRefID="261"/>
    <map charValue="u203E" charName="OVERLINE" glyphRefID="325"/>
    <map charValue="u2044" charName="FRACTION SLASH" glyphRefID="246"/>

```



```

        <map charValue="u2212" charName="MINUS SIGN" glyphRefID="514"/>
        <map charValue="u2219" charName="BULETTE OPERATOR"
glyphRefID="331"/>
        <map charValue="u223C" charName="TILDE OPERATOR" glyphRefID="631"/>
    </cmapOverride>
</ComponentDef>
<ComponentDef name="TokyoSans">
</ComponentDef>
</Components>
</PosingFont>

```

B.3 Example 3

MonteCarloSans is a very good Latin font. We want to enhance this font to support CJK languages. We can do so by creating a CFR resource called ColumboSans with 2 components. The first component is MonteCarloSans and the second component is a language preferred list. This list defines a font to use when the user's system is running in Japanese (TokyoSans), Traditional Chinese (TaipeiSans), Simplified Chinese (BeijingSans), and Korean (SeoulSans). If the user is running in one of these languages, and the font chosen for that language cannot map the specified character, the language order appearing in the list is significant. For example, if the user's language is set to Simplified Chinese, BeijingSans would be used first, followed by TokyoSans, TaipeiSans, and SeoulSans.

Note that the Latin glyphs will come from MonteCarloSans because it was the first component font defined.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PosingFont SYSTEM
"file://localhost/System/Library/DTDs/SplicedFont.dtd">

<PosingFont name="ColumboSans" version="1.0">
  <Name type="3" string="Orient Press AG. 1998(c)"/>
  <FontMetrics
    unitsPerEm="1000"
    ascender="561.1"
    descender="214.3"
    lineGap="21.0"
  />
  <Components>
    <ComponentDef name="MonteCarloSans">
      <Matrix
        xx="1.0"
        xy="0.0"
        yx="0.0"
        yy="1.0"
        tx="0.0"
        ty="-6.4"
      />
    </ComponentDef>
    <LanguagePreferredList>
      <LanguagePreferredComponentDef>
        <Language string="ja"/>
        <ComponentDef name="TokyoSans"/>
      </LanguagePreferredComponentDef>
      <LanguagePreferredComponentDef>
        <Language string="zh-Hant"/>
        <ComponentDef name="TaipeiSans"/>
      </LanguagePreferredComponentDef>
      <LanguagePreferredComponentDef>
        <Language string="zh-Hans"/>
        <ComponentDef name="BeijingSans"/>
      </LanguagePreferredComponentDef>
    </LanguagePreferredList>
  </Components>
</PosingFont>

```

```

        </LanguagePreferredComponentDef>
        <LanguagePreferredComponentDef>
            <Language string="ko"/>
            <ComponentDef name="SeoulSans">
                <Tracking trackingValue = "-5.1"/>
            </ComponentDef>
        </LanguagePreferredComponentDef>
    </LanguagePreferredList>
</Components>
</PosingFont>

```

B.4 Example 4

Expanding on Example 3, if we wanted to have full control of what fonts are to be used for a user's CJK language setting, we can use a list of components for each language. For example, if the user's language is set to Simplified Chinese, BeijingSans will be used first, followed by TaipeiSans, TokyoSans, and SeoulSans.

```

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE PosingFont SYSTEM
"file://localhost/System/Library/DTDs/SplicedFont.dtd">

<PosingFont name="ColumboSans" version="1.0">
    <Name type="3" string="Orient Press AG. 1998 (c)"/>
    <FontMetrics
        unitsPerEm="1000"
        ascender="561.1"
        descender="214.3"
        lineGap="21.0"
    />
    <Components>
        <ComponentDef name="MonteCarloSans">
            <Matrix
                xx="1.0"
                xy="0.0"
                yx="0.0"
                yy="1.0"
                tx="0.0"
                ty="-6.4"
            />
        </ComponentDef>
        <LanguagePreferredList>
            <LanguagePreferredComponentDef>
                <Language string="ja"/>
                <ComponentDef name="TokyoSans"/>
                <ComponentDef name="SeoulSans">
                    <Tracking trackingValue = "-5.1"/>
                </ComponentDef>
                <ComponentDef name="BeijingSans"/>
                <ComponentDef name="TaipeiSans"/>
            </LanguagePreferredComponentDef>
            <LanguagePreferredComponentDef>
                <Language string="zh-Hant"/>
                <ComponentDef name="TaipeiSans"/>
                <ComponentDef name="BeijingSans"/>
                <ComponentDef name="TokyoSans"/>
                <ComponentDef name="SeoulSans">
                    <Tracking trackingValue = "-5.1"/>
                </ComponentDef>
            </LanguagePreferredComponentDef>
        </LanguagePreferredList>
    </Components>
</PosingFont>

```

```

    <Language string="zh-Hans"/>
    <ComponentDef name="BeijingSans"/>
    <ComponentDef name="TaipeiSans"/>
    <ComponentDef name="TokyoSans"/>
    <ComponentDef name="SeoulSans">
      <Tracking trackingValue = "-5.1"/>
    </ComponentDef>
  </LanguagePreferredComponentDef>
<LanguagePreferredComponentDef>
  <Language string="ko"/>
  <ComponentDef name="SeoulSans">
    <Tracking trackingValue = "-5.1"/>
  </ComponentDef>
  <ComponentDef name="BeijingSans"/>
  <ComponentDef name="TaipeiSans"/>
  <ComponentDef name="TokyoSans"/>
</LanguagePreferredComponentDef>
</LanguagePreferredList>
</Components>
</PosingFont>

```

Bibliography

- [1] AGL Specification, <<http://sourceforge.net/adobe/aglfn/aglspec/>>
- [2] ICU UnicodeSet Class Reference, <http://ssl.icu-project.org/apiref/icu4c/classUnicodeSet.html>
- [3] ISO/IEC 10646, *Information technology — Universal Coded Character Set (UCS)*
- [4] ISO/IEC 14496-22, *Information technology — Coding of audio-visual objects — Open Font Format*

