INTERNATIONAL STANDARD ISO/IEC 14496-28:2012
TECHNICAL CORRIGENDUM 2

Published 2014-07-01

Information technology — Coding of audio-visual objects —

Part 28:
Composite font representation

TECHNICAL CORRIGENDUM 2: Changes and clarifications of CFR element descriptions

Technologies de l'information — Codage des objets audiovisuels —

Partie 28: Représentation de la police de caractères composite

RECTIFICATIF TECHNIQUE 2: Modifications et précisions de descriptions d'éléments CFR

Technical Corrigendum 2 to ISO/IEC 14496-28:2012 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

This corrigendum aims to clarify the meaning and use of certain Composite Font elements and explain their relationships within the components of the document.

In clause 2 "Normative references" – add the new normative reference to the ISO/IEC 14496-22 "Open Font Format"

In subclause 5.1 "High-level overview" – replace the text description with the following:

A CFR resource consists of a single 'PosingFont' element, and its 'Components' element specifies an arbitrary number of 'ComponentDef' or 'LanguagePreferredList' elements. It may contain a number of optional elements (such as 'Name' and 'FontMetrics' elements) which provide additional information about the CFR resource and its metrics. Whenever an optional element is missing from the CFR, or if there are optional attributes that are missing from any of the elements – the corresponding information expected to be provided by the missing optional elements and attributes shall be inherited from the first component font specified by either 'ComponentDef' or 'LanguagePreferredList' element. Please note that the choice of the first component font specified by a 'LanguagePreferredList' element may be dependent on user's language preferences and, therefore, font metrics and other values inherited from the first component font may also change depending on user language preferences.

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.

In subclause 5.3 "The 'Name' element" – replace the first paragraph of the element description with the following:

Optional. Each instance of the 'Name' element specifies an attribute that is equivalent to a specific 'name' table string of an 'sfnt' resource. The original set of 'name' table entries shall be inherited from the first component font; however, whenever an attribute is present, it will effectively override a particular 'name' table entry with the string defined by the particular instance of the element.

In subclause 5.4 "The 'FontMetrics' element" – replace the first paragraph of the element description with the following:

Optional. The 'FontMetrics' element specifies various line-layout attributes that apply globally to the CFR resource. All attributes of the element are optional, if the element or any of its attributes are not present in the CFR resource definitions but the missing font metrics data is deemed essential for proper layout – the corresponding font metrics entry from the first component font shall be used. Since the choice of the first component font may vary based on the user language preferences, it is important to make sure that either language-specific components have matching font metrics or that the attributes and their values provided by the 'FontMetrics' element define a complete overriding set of parameters to avoid the unwanted layout effects.

In subclause 5.4 "The 'FontMetrics' element" – replace the attribute definitions of the element with the following:

`unitsPerEm = "string"`

Optional. A virtual UPM (Units Per Em) of the CFR resource expressed as an integer. Its only purpose is to facilitate the correct interpretation of the additional attributes defined by this element, the actual UPM values of the component fonts defined by the CFR may differ both between the component fonts and from the CFR FontMetrics value. The values of the additional attributes are based on this UPM value.

`macStyle = "string"`

Optional. The head.macStyle value expressed as a 16-digit bit array, see ISO/IEC 14496-22 for detailed descriptionof the attribute value.

`ascender = "string"`

Optional. When provided, this value of `ascender` will be used by the host rendering system implementations and will override the default value of each component font – it is the CFR author's responsibility to make sure that the value is suitable for all component fonts on various host rendering systems.

If the value of `ascender` is not defined, the host rendering system will use the first, highest priority component font to determine the `ascender` value according to the algorithm defined in the ISO/IEC 14496-22 (in the "Baseline to Baseline Distances" section of the "Recommendations" clause.).

`descender = "string"`

Optional. When provided, this value of `descender` will be used by the host rendering system implementations and will override the default value of each component font – it is the CFR author's responsibility to make sure that the value is suitable for all component fonts on various host rendering systems.

If the value of `descender` is not defined, the host rendering system will use the first, highest priority component font to determine the `descender` value according to the algorithm defined in the ISO/IEC 14496-22 (in the "Baseline to Baseline Distances" section of the "Recommendations" clause.).

`linegap = "string"`

Optional. When provided, this value of `linegap` will be used by the host rendering system implementations and will override the default value of each component font – it is the CFR author's responsibility to make sure that the value is suitable for all component fonts on various host rendering systems.

If the value of `linegap` is not defined, the host rendering system will use the first, highest priority component font to determine the `linegap` value according to the algorithm defined in the ISO/IEC 14496-22 (in the "Baseline to Baseline Distances" section of the "Recommendations" clause.).

`xMaxExtent = "string"`

Optional. The value of the formula MAX (minLeftSideBearing + (xMax - xMin)) expressed as an integer.

`yMaxExtent = "string"`

Optional. The value of the formula MAX (minTopSideBearing + (yMax - yMin)) expressed as an integer.

`caretSlopeRise = "string"`

Optional. The value, expressed as an integer, is used to calculate the slope of the cursor according to the formula (caretSlopeRise ÷ caretSlopeRun). Use the value 1 for a vertical cursor, which is best for horizontal writing mode.

`caretSlopeRun = "string"`

Optional. The value, expressed as an integer, is used to calculate the slope of the cursor according to the formula (caretSlopeRise ÷ 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, expressed as an integer. Use the value 0 for non-slanted fonts.

`vertCaretSlopeRise = "string"`

Optional. The value, expressed as an integer, is used to calculate the slope of the cursor according to the formula (vertCaretSlopeRise ÷ vertCaretSlopeRun). Use the value 0 for a horizontal cursor, which is best for vertical writing mode.

`vertCaretSlopeRun = "string"`

Optional. The value, expressed as an integer, is used to calculate the slope of the cursor according to the formula (vertCaretSlopeRise ÷ vertCaretSlopeRun). Use the value 1 for a horizontal cursor, which is best for vertical writing mode.