FontMap

Introduction and Overview

The FontMap module provides two functions:

- A means by which font names from other systems can be converted to RISC OS font names used by FontManager
- A means by which a font's properties can be returned and manipulated

The FontMap module allows conversion of the names os fonts from Windows (which have generally been converged upon). These font mappings are intended to be extended and configurable on a given system.

The font properties which the FontMap module manages are those that can be derived from the font name - the weight, obliqueness, and condensedness features.

Technical details

Font naming systems

The FontMap module handles two naming systems, but has scope to be expanded with more in the future. The module specifically handles the RISC OS naming system, and the Windows naming system. Although the system is called 'Windows', it is actually a more widely adopted naming scheme due to the specification of fonts on websites and the availability of fonts.

There is another naming system which is not supported by FontMap at present - the PostScript font naming system. This is partially catered for by the Printer driver system, so is not explicitly handled in the FontMap module.

The font naming systems use the following values:

Value	Meaning
0	RISC OS
1	Windows
2-7	Reserved

Font naming components

Fonts consist of four major features which can be recognised by the FontMap module:

- The font family name, for example 'Homerton' or 'Helvetica'.
- The weight of the font, for example 'Bold', or 'Light'.
- The obliqueness of the font, which is commonly given as 'Oblique' or 'Italic'.
- The stretchedness of the font, which is commonly given as 'Condensed' or 'Expanded'.

The properties recognised here take their inspiration from the *SVG 'font-face' element*, which is itself based on the CSS 2 '@font-face' facility.

Font family

The font family generally groups fonts which have the stylistic design. How characters are formed, and how they relate to one another is what distinguishes the families of fonts. The FontMap module can perform a mapping between these font families where they are sufficiently similar. This is necessary because RISC OS does not use the same standardised names for its fonts as are used elsewhere.

For example, the RISC OS font family has similar stylistic design to the font used elsewhere as 'Helvetica'. The FontMap module allows a translation between these font names to produce an approximation of the original intent. This translation is (commonly) bidirectional, so that as well as importing content which uses the foreign naming schemes to RISC OS, the RISC OS font names can be translated to the equivalent foreign font names.

Font weight

The weight of a font indicates the relative darkness or heaviness of the text. There is no standardised naming scheme for the weights of gonts, but there are commonly used weight names that are used in a self-consistent continuum within a family.

The FontMap module assigns these weights values which can be compared and used to understand the relationship between different fonts within the family.

Value	Meaning
0	Don't care - a value that can be requested when any suitable weight is accepted
100	Thin
200	Extra light
300	Light
400	Regular
500	Medium
600	Semi-bold
700	Bold
800	Extra bold
900	Black

Font obliqueness

The obliqueness of a font indicates whether the font is set at an angle (which is usually used for emphasis), or not. Within the FontMap module this is a binary choice - either the font is at an angle or it is not.

Stylistically, the font may differentiate the manner in which it is angled.

Generally, 'Oblique' fonts are the same as the non-oblique font, but with a simple transformation applied to make the top of the font lean rightwards. 'Italic' fonts are those that have been specially crafted to look different when set at an angle, sometimes with different ornamentation or flourishes, or with different emphasis on components of the characters.

The FontMap module does not differentiate between 'Italic' and 'Oblique' variants of a font, but treats both as the same state in its values. The FontMap module has no support for the degree to which the font is angled, and has no support for reverse leaning fonts. The values used by the module for obliqueness are:

Value	Meaning
0	Don't care - a value that can be requested when any suitable obliqueness is accepted
1	Plain, not at any angle
2	Oblique or italic

Font stretch

The stretch of a font gives a relative measure of how much wider or thinner a font is than others within the same family. As with the font weight names, there is no standardised naming for these features, but a continuum within a given font is usually expected.

The values used by FontMap are:

Value	Meaning
0	Don't care - a value that can be requested when any suitable stretch is accepted
100	Ultra-condensed
200	Extra-condensed
300	Condensed
400	Semi-condensed
500	Normal
600	Semi-expanded
700	Expanded
800	Extra-expanded
900	Ultra-expanded

FontMap mappings file

The Font Mapping file describes the translation between formats. It is configured through the *FontMap$Mappings (on page 7)* variable.

The translation file is a text file containing lines for mappings between the formats. The file has a simple line oriented format, which uses '#' character prefix for comments. Other lines consist of:

- The source system format, either an 'R' or a 'W' for RISC OS or Windows format of font family.
- The conversion type, either a '-' or a '>'. '-' indicates a bidirectional conversion between the source and destination system formats. '>' indicates a unidirectional conversion.
- The destination system format, either an 'R' or a 'W' for RISC OS or Windows format of font family.
- A colon (':') separator.
- The source font family name.
- A colon (':') separator.
- The destination font family name.

Described in ABNF the format is:

```
mapping-file = *mapping-line
mapping-line = [ comment / rule ] LF
comment      = "#" *( SP / VCHAR )
rule         = src-system conversion dst-system ":" src-family ":" dst-family
src-system   = system
dst-system   = system
conversion   = "-" / ">"
src-family   = family
dst-family   = family
family       = 1*( SP / ALPHA / DIGIT / "-" / "_" )
```

Syntax for FontMap mappings files

By default the FontMap module will use a font mapping file that looks similar to the following:

```
# Foreign font mappings
#
# Cut down form as fallback for ROM.

# Standard mappings for RISC OS and Windows fonts
R-W:Trinity:Times New Roman
```

```
R-W:Homerton:Arial
W>R:Helvetica:Homerton
R-W:Corpus:Courier New
```

Translating font names

The *SWI FontMap_Translate (on page 8)* SWI provides a means of applying these properties and mappings to font names. The FontMap module will try to satisfy requests using the information available to it from the FontManager and the font mapping file.

The translation process goes through simple steps to translate the fonts:

1. Identify the input font and its properties in the input font system. This means extracting the font properties from the name, and identifying the font family.
2. Replace any of the properties from the input font with those supplied by the user.
3. Map the font family name from the input system to the output system.
4. Search the output font system for a suitable font name for the closest match to the font required.
5. Return the result to the caller.

This process is expected to be inexact when translating between font systems. Different property inputs may match multiple fonts - for example, if an Extra Bold weight font is requested, but only Bold is available, the Bold will be returned, just as it would if Bold weight had been requested.

When translating RISC OS fonts, the FontMap module is able to synthesise oblique fonts by using a font transformation matrix. This is only performed when there is no suitable oblique or italic font.

System variables

FontMap$Mappings

Filename of the FontMap mappings definition

Use

The `FontMap$Mappings` variable contains a filename which should be used for the mappings to and from RISC OS font names. The file's format is described in the *FontMap mappings file (on page 5)*.

Related SWIs

SWI FontMap_Translate (on page 8)

SWI calls

FontMap_Translate (SWI &57F00)

Perform a translation of the supplied font name

On entry

R0 = Flags:

Bit(s)	Meaning
0-2	Input *font name system (on page 2)*
3-5	Output *font name system (on page 2)*

R1 = Pointer to a zero-terminated font name
R2 = Requested *font oblique state (on page 3)*, or 0 for a default
R3 = Requested *font weight (on page 3)*, or 0 for a default
R4 = Requested *font stretch state (on page 4)*, or 0 for a default
R5 = Pointer to output buffer for font name, or 0 to just request the length required
R6 = Length of the font buffer

On exit

R0 preserved
R1 preserved
R2 = Returned font oblique state
R3 = Returned font weight
R4 = Returned font stretch state
R5 preserved
R6 = Number of bytes written to the output buffer, including the terminator

Interrupts

Interrupts are undefined
Fast interrupts are enabled

Processor mode

Processor is in SVC mode

Re-entrancy

SWI is not re-entrant

Use

The Font_Translate SWI is used to either convert between RISC OS and foreign font naming systems, or to apply styling to a font name. The requested properties will be matched as closely as possible for a given font, and the actual matched font's properties will be returned. Because the closest match for a given font may not be suitable, callers may wish to check the returned properties.

If no translation is required, and only the font properties are desired, R5 may be set to 0. This will perform the translation and update the properties in the return registers, but no font name will be returned.

If the font name will not fit into the output buffer, the value of R6 will be updated to the size required, and the Buffer overflow (error number &1E5) error will be returned.

Related APIs

None

Error messages

Error_BadFlags (Error &81AE40)

Invalid flags were supplied to FontMap

Use

This error is returned by the FontMap module when the flags supplied in R0 are not recognised.

Related SWIs

SWI FontMap_Translate (on page 8)

Error_BadMappingType (Error &81AE41)

Invalid font format mapping was requested of FontMap

Use

This error is returned by the FontMap module when the input or output font system supplied in R0 was not recognised.

Related SWIs

SWI FontMap_Translate (on page 8)

Error_NoMappingAvailable (Error &81AE42)

No equivalent font name available

Use

This error is returned by the FontMap module when the font name, or the font family name, was nor recognised, or when the equivalent font is not available in the output font system.

Related SWIs

SWI FontMap_Translate (on page 8)

Examples

Reading the properties of a font:

```
>DIM buf% 256
>SYS "FontMap_Translate",(0<<0) OR (0<<0), "Homerton.Medium", 0,0,0 TO
,,oblique,weight,stretch
>PRINT oblique, weight, stretch
         1       500       500
```

Making a font bold:

```
>DIM buf% 256
>SYS "FontMap_Translate",(0<<0) OR (0<<0), "Homerton.Medium",
0,900,0,buf%,256 TO ,,oblique,weight,stretch,font$
PRINT oblique, weight, stretch; " - ";font$
         1       900       500 - Homerton.Bold
```

Document information

Maintainer(s): Charles Ferguson <gerph@gerph.org>

History:

Revision	Date	Author	Changes
1	17 Oct 2020	Gerph	Initial version for 0.05 ● Initial documentation created for the old module.

Disclaimer: © Gerph.