RenderWare Graphics

White Paper

Using Unicode

Contact Us

Criterion Software Ltd.

For general information about RenderWare Graphics e-mail info@csl.com.

Developer Relations

For information regarding Support please email devrels@csl.com.

Sales

For sales information contact: rw-sales@csl.com

Acknowledgements

With thanks to RenderWare Graphics development and documentation teams.

1. Introduction

Unicode is a character code set, much like ASCII, except it encodes beyond the 127 ASCII characters. It has been designed to provide a unique code to all characters used in major languages around the globe.

ASCII characters are supported in Unicode, with the first 127 characters matching that of the ASCII character set.

For more information on Unicode, see http://www.unicode.org

UTF-8 is a method of encoding Unicode character codes while remaining compatible with ASCII. Unicode can use either 2 or 4 bytes to encode a single character. Obviously, this can cause confusion when mixed with single byte ASCII characters. UTF-8 provides a way of encoding multi-byte characters with single byte characters.

For more information on UTF-8 and how it encodes Unicode characters, see http://www.cl.cam.ac.uk/~mgk25/unicode.html#unicode

RenderWare uses UTF-8 for its font metric files.

Metrics 1

If you look at the specification for the metrics1 type font (described in the API Reference) you'll understand that this file does not need to be encoded using UTF-8. Since it lists Unicode code points in a numerical form (as unsigned decimal), you can use regular text files. Thus, the following is the start of a metrics type 1 font:

```
METRICS1
unicode-met1.png
5
19968   0     0     32    40
19969   33    0     65    40
19970   66    0     98    40
19971   99    0     131   40
19972   132   0     164   40
19973   165   0     197   40
19974   198   0     230   40
19975   231   0     263   40
19976   264   0     296   40
19977   297   0     329   40
19978   330   0     362   40
```

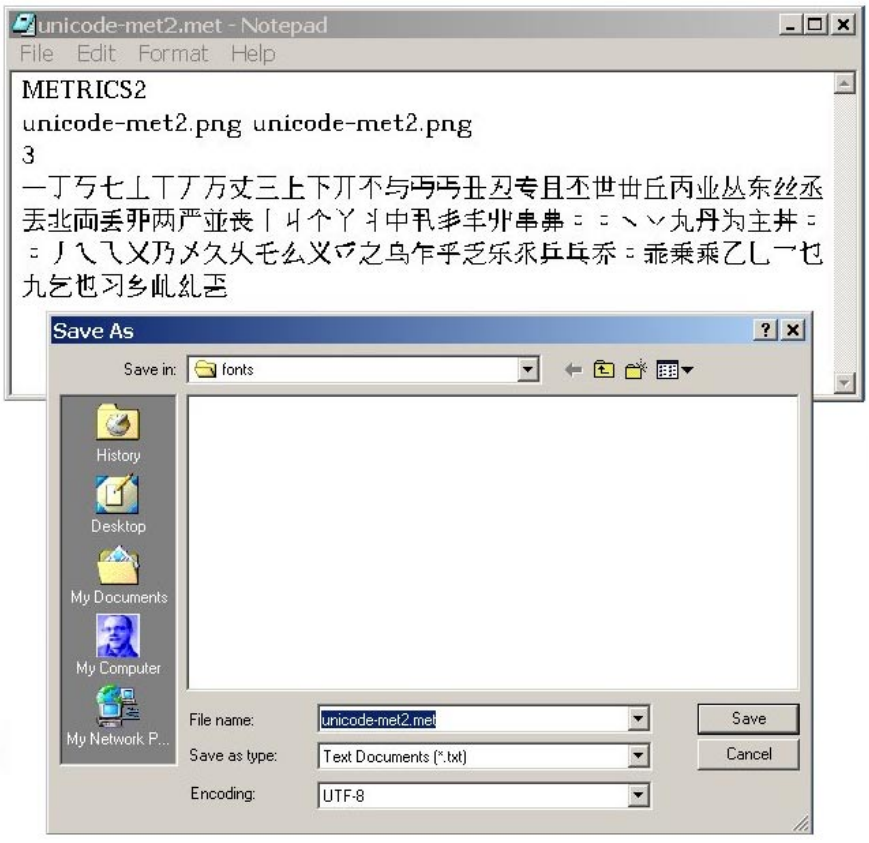
The numbers in the first column are the first characters in the CJK Unified Ideographs section of the Unicode specification. ("CJK" stands for Chinese, Japanese and Korean.)

Note that the `makefont` utility that is shipped with the SDK cannot, currently, generate characters outside of the range 32..127. Future versions might have this restriction removed. This utility seems to place a comment at the end of each line. My experience is that this causes font reading to fail sometimes. If you have problems, then one thing to try is to remove these comments.

So, metrics type 1 files should be relatively easy to create. One point that must be considered, however, is that the `makefont` utility cannot be used to generate artwork that your title can ship with, unless you own the copyright on the font file you are rasterizing. For example, rasterizing a Windows font and using this in your game will break your license agreement with Microsoft. We provide makefont only to help developers generate stand-in artwork and to assist with debugging.

Metrics 2

Metrics type 2 fonts are more tricky. You know already that these have to be UTF-8 encoded. The reason for this is that the characters that are used in the texture are written into the metrics file. As it happens, in western version of Windows, notepad can create UTF-8 encoded files. When you save your file select the UTF-8 encoding type. See the picture below:



Unfortunately, this process creates a file on disk that contains 3 header bytes (that presumably are used by Microsoft to store the fact that the file is UTF-8 encoded.) Before you can use your UTF-8 encoded file with the `Rt2d` library you will have to strip off these bytes. One approach is to load the file into DevStudio as a binary file, and then delete the first 3 characters.

There is no easy way to enter the Unicode characters into notepad using a regular keyboard. One approach is to use something like Microsoft Word and insert characters using the *Insert→Symbol* menu. In testing, a small macro was written to enter characters:

```
Sub InsertUnicode()
'
' InsertUnicodeMacro
' Macro created 21/03/03
'
   For i = 19968 To 20068
        Selection.InsertSymbol CharacterNumber:=i, Unicode:=True
   Next
End Sub
```

This just takes the first 101 characters in the CJK range and inserts them into a Word document. If you select a font that has these characters, for example, the shareware Code2000 font (see http://home.att.net/~jameskass) then you can cut and paste these characters into notepad. Naturally, you need to tell Notepad to use the same font (under *Format→Fonts*...). Following this procedure allows you to create the `.met` font description.

Examples of metrics1 and metrics2 files are displayed below. If you want to experiment using them, you'll need to know that the strings passed to `Rt2dFontShow` are regular 8-bit character strings. RenderWare knows whether you have loaded Unicode fonts or not. If you are rendering using a Unicode font, then the strings passed to `Rt2dFontShow` are treated as UCS-2, or, put simply as 16-bit integers. So, for illustration, the following can be used to generate a suitable string:

```
RwChar s[256];
RwUInt16 *p = (RwUInt16 *)s;
RwUInt16 i;

for (i = 19968; i < 20000; i++)
  *p++ = i;
*p++ = 0;
// now we can do: Rt2dFontShow(f, s, h, &a, b);
```

The `.met` and `.png` files shown below can be downloaded from the RenderWare Support site (FMSS):

https://support.renderware.com/kb/upload/unicode-met1.met

https://support.renderware.com/kb/upload/unicode-met1.png

https://support.renderware.com/kb/upload/unicode-met2.met

https://support.renderware.com/kb/upload/unicode-met2.png

2. Unicode .met File Examples

The `.met` and `.png` files shown below can be downloaded from the RenderWare Support site (FMSS):

- https://support.renderware.com/kb/upload/unicode-met1.met
- https://support.renderware.com/kb/upload/unicode-met1.png
- https://support.renderware.com/kb/upload/unicode-met2.met
- https://support.renderware.com/kb/upload/unicode-met2.png

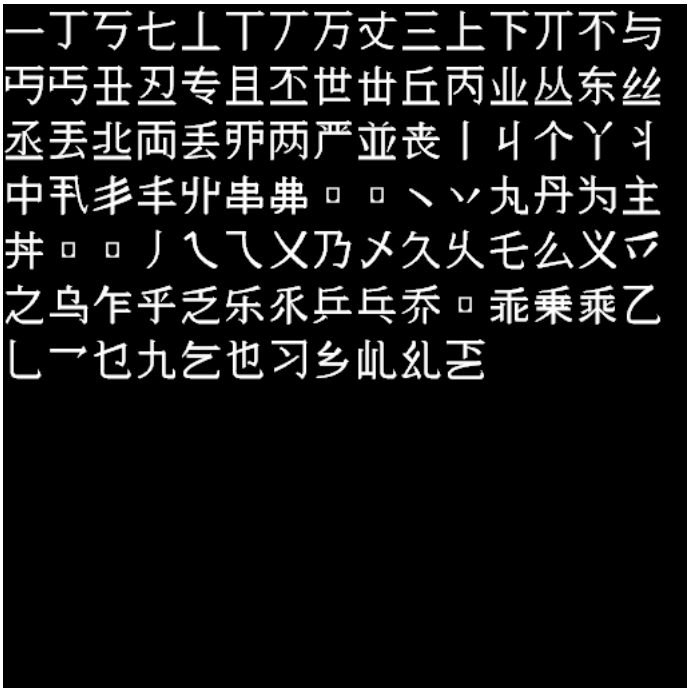
unicode-met1.met

```
METRICS1
unicode-met1.png
5
19968   0   0  32  40 # ' '
19969  33   0  65  40 # ' '
19970  66   0  98  40 # '1'
19971  99   0 131  40 # ' '
19972 132   0 164  40 # ' '
19973 165   0 197  40 # '[0]'
19974 198   0 230  40 # ' '
19975 231   0 263  40 # ' '
19976 264   0 296  40 # ' '
19977 297   0 329  40 # ''
19978 330   0 362  40 # '
'
19979 363   0 395  40 # '
'
19980 396   0 428  40 # ' '
19981 429   0 461  40 # '
'
19982 462   0 494  40 # ' '
19983   0  41  32  81 # ' '
19984  33  41  65  81 # ' '
19985  66  41  98  81 # ' '
19986  99  41 131  81 # '_'
19987 132  41 164  81 # ' '
19988 165  41 197  81 # ' '
19989 198  41 230  81 # '_'
19990 231  41 263  81 # ' '
19991 264  41 296  81 # ' '
19992 297  41 329  81 # ' '
19993 330  41 362  81
19994 363  41 395  81
19995 396  41 428  81 # ' '
19996 429  41 461  81 # ' '
19997 462  41 494  81 # ' '
```

```
19998   0  82  32 122 # '-'
19999  33  82  65 122 # ''
20000  66  82  98 122 # ' '
20001  99  82 131 122 # '!'
20002 132  82 164 122 # '"'
20003 165  82 197 122 # '#'
20004 198  82 230 122 # '$'
20005 231  82 263 122 # '%'
20006 264  82 296 122 # '&'
20007 297  82 329 122 # '''
20008 330  82 362 122 # '('
20009 363  82 395 122 # ')'
20010 396  82 428 122 # '*'
20011 429  82 461 122 # '+'
20012 462  82 494 122 # ','
20013   0 123  32 163 # '-'
20014  33 123  65 163 # '.'
20015  66 123  98 163 # '/'
20016  99 123 131 163 # '0'
20017 132 123 164 163 # '1'
20018 165 123 197 163 # '2'
20019 198 123 230 163 # '3'
20020 231 123 263 163 # '4'
20021 264 123 296 163 # '5'
20022 297 123 329 163 # '6'
20023 330 123 362 163 # '7'
20024 363 123 395 163 # '8'
20025 396 123 428 163 # '9'
20026 429 123 461 163 # ':'
20027 462 123 494 163 # ';'
20028   0 164  32 204 # '<'
20029  33 164  65 204 # '='
20030  66 164  98 204 # '>'
20031  99 164 131 204 # '?'
20032 132 164 164 204 # '@'
20033 165 164 197 204 # 'A'
20034 198 164 230 204 # 'B'
20035 231 164 263 204 # 'C'
20036 264 164 296 204 # 'D'
20037 297 164 329 204 # 'E'
20038 330 164 362 204 # 'F'
20039 363 164 395 204 # 'G'
20040 396 164 428 204 # 'H'
20041 429 164 461 204 # 'I'
20042 462 164 494 204 # 'J'
20043   0 205  32 245 # 'K'
20044  33 205  65 245 # 'L'
20045  66 205  98 245 # 'M'
20046  99 205 131 245 # 'N'
20047 132 205 164 245 # 'O'
20048 165 205 197 245 # 'P'
20049 198 205 230 245 # 'Q'
20050 231 205 263 245 # 'R'
20051 264 205 296 245 # 'S'
```

```
20052 297 205 329 245 # 'T'
20053 330 205 362 245 # 'U'
20054 363 205 395 245 # 'V'
20055 396 205 428 245 # 'W'
20056 429 205 461 245 # 'X'
20057 462 205 494 245 # 'Y'
20058   0 246  32 286 # 'Z'
20059  33 246  65 286 # '['
20060  66 246  98 286 # '\'
20061  99 246 131 286 # ']'
20062 132 246 164 286 # '^'
20063 165 246 197 286 # ' '
20064 198 246 230 286 # '`'
20065 231 246 263 286 # 'a'
20066 264 246 296 286 # 'b'
20067 297 246 329 286 # 'c'
20068 330 246 362 286 # 'd'
```

unicode-met1.png



Unicode-Met2

unicode-met2.met

This example contains displays a list of fonts. To be able to display these fonts you need to have the relevant fonts installed.

unicode-met2.png