TEST

Test
Test

TEST

Test
Test

QTF

QTF is the native format of Ultimate++ rich texts (formatted texts).

It is byte oriented format. Bytes with values 2-31 are ignored. Other values are interpreted as characters or formatting commands.

Letters (a-zA-Z), numbers (0-9), space (32) and characters

. , ; ! ? % ( ) / < > #

and bytes greater than 127 are guaranteed to be never used as command characters (not even in future versions of QTF). Other characters should be prefixed with escape character ` (reverse apostrophe). Group of characters can be escaped using byte 1. Example:

```
"\1a[x]\1[* bold]"
```

Byte 0 represents the end of input sequence.

Dimension units of QTF are dots - one dot is defined as 1/600 of inch.

Colors are described as either number 0-9, with meaning

0	1	2	3	4	5	6	7	8	9
Black	LtGray	White	Red	Green	Blue	LtRed	WhiteGray	LtCyan	Yellow

or letters

b	c	g	k	l	m	o	r	y
Blue	Cyan	Green	Black	LtGray	Magenta	Brown	Red	Yellow

B	C	G	K	L	M	W	R	Y
LtBlue	LtCyan	LtGreen	Gray	WhiteGray	LtMagenta	White	LtRed	LtYellow

or as the RGB value in form

(*number.number.number*)

where *number* is 0-255.

Form with single *number*

(*number*)

specifies grays.

Letter

N

used in place of color designates transparent color.

Basic QTF codes

_	Hard-space - space that cannot be divided at the end of line.
&	New paragraph.
-|	Tabulator
@$**hex**;	Unicode character as hexadecimal number.

Character and paragraph formatting

Character and paragraph formatting starts with

[

character followed by ***character/paragraph formating*** *sequence* ended with single space character and applies to the text until matching

]

is encountered.

Character/paragraph formating sequence	
/	Italic.

*	Bold.
_	Underline.
-	Strikeout.
c	Capitals.
`	Superscript.
,	Subscript.
d	Dashed underline.
T	Non anti aliased font.
^*text*^	Hyperlink.
I*text*;	Index entry.
+*number*	Font height in dots.
0	Font height 50 dots (6 points).
1	Font height 67 dots (8 points).
2	Font height 84 dots (10 points).
3	Font height 100 dots (12 points).
4	Font height 134 dots (16 points).
5	Font height 167 dots (20 points).
6	Font height 200 dots (24 points).
7	Font height 234 dots (28 points).
8	Font height 300 dots (36 points).
9	Font height 400 dots (48 points).
@*color*	Text color.
$*color*	Text background color.
A	Arial font.
R	Times New Roman font.
C	Courier font.
G	Standard GUI font.
S	Symbol font.
.*number*	Font with specified font *number*.
!*text*!	Font with face name equal to *text*. If such font does not exist on system, Arial is used.
{*charset*}	Character set. It can be defined as either single character _ utf-8 0 windows-1250 1 windows-1251 2 windows-1252 3 windows-1253 4 windows-1254 5 windows-1255

	<table><tr><td>_</td><td>utf-8</td></tr><tr><td>6</td><td>windows-1256</td></tr><tr><td>7</td><td>windows-1257</td></tr><tr><td>A</td><td>iso-8859-1</td></tr><tr><td>B</td><td>iso-8859-2</td></tr><tr><td>C</td><td>iso-8859-3</td></tr><tr><td>D</td><td>iso-8859-4</td></tr><tr><td>E</td><td>iso-8859-5</td></tr><tr><td>F</td><td>iso-8859-6</td></tr><tr><td>G</td><td>iso-8859-7</td></tr><tr><td>H</td><td>iso-8859-8</td></tr><tr><td>I</td><td>iso-8859-9</td></tr><tr><td>J</td><td>iso-8859-10</td></tr></table> or as the string designating character set (example: **"[{_} ...]"**, **"[{windows-1250} ...]"**).
%*lang*	Language. It is defined in form XX-YY, according to ISO 639 and ISO 3166 standards. (example: **"[%EN-US ...]"**). **"[%00-00 "** represents "none" language. As special optimization, **%-** is equivalent to %00-00 (no language) and %% is equivalent to %EN-US.
:*text*:	Paragraph label.
<	Align paragraph left.
=	Center paragraph.
>	Align paragraph right.
#	Justify paragraph.
l*number*	Left margin in dots.
r*number*	Right margin in dots.
i*number*	Indent in dots.
pn	Line spacing 1.0.
ph	Line spacing 1.5.
pd	Line spacing 2.0.
H*number*	Horizontal ruler height (if zero, there is no ruler).
h*color*	Color of horizontal ruler (default is black).
L*number*	Style of horizontal ruler line, 0 - solid color, 1 - dots, 2 - dashes
b*number*	Space before paragraph in dots.
a*number*	Space after paragraph in dots.
P	Page break before paragraph.
k	Keep paragraph on single page.
K	Keep paragraph on same page as next one.
Q	Orphan control.

<table>
<tr><td>ntext;</td><td>Text to insert before paragraph number.</td></tr>
<tr><td>mtext;</td><td>Text to insert after paragraph number.</td></tr>
<tr><td>N</td><td>Numbering. It is followed by up to 8 characters defining numbering style of each level

<table><tr><td>-</td><td>Level is not used.</td></tr><tr><td>1</td><td>Numbers, starting with 1.</td></tr><tr><td>0</td><td>Numbers, starting with 0.</td></tr><tr><td>a</td><td>Lowercase letters, starting with a.</td></tr><tr><td>A</td><td>Uppercase letters, starting with A.</td></tr><tr><td>i</td><td>Lowercase roman numbers, starting with i.</td></tr><tr><td>I</td><td>Uppercase roman numbers, starting with I</td></tr></table>
If after 8 numbering style characters there is a '!' suffix, the reset is performed for those numbering levels that are used. If all levels are not used, reset is performed for the top-level.</td></tr>
<tr><td>o</td><td>Bullet style.</td></tr>
<tr><td>O_</td><td>No bullet.</td></tr>
<tr><td>O0</td><td>Bullet style.</td></tr>
<tr><td>O1</td><td>Bullet style.</td></tr>
<tr><td>O2</td><td>Bullet style.</td></tr>
<tr><td>O3</td><td>Bullet style.</td></tr>
<tr><td>O9</td><td>Text bullet style.</td></tr>
<tr><td>tnumber</td><td>Default tab size.</td></tr>
<tr><td>~</td><td>Tabulator setting. Can be followed by character designating type of tabulation

<table><tr><td>></td><td>Normal tabulation.</td></tr><tr><td><</td><td>Left tabulation.</td></tr><tr><td>=</td><td>Centered tabulation.</td></tr></table>
(default is normal) and by filler character

<table><tr><td>.</td><td>-</td><td>_</td></tr></table>
and is followed by number specifying tabulator position in dots. The number can be preceded by > - in that case, position is relative to the right side.

When followed by ~ ("~~"), clears all current tab settings (including those inherited from paragraph style).</td></tr>
</table>

;	NOP separator. In some cases it is needed to separate command code. Example: `"[l200;4 "`
s*number* s"*text*"	Paragraph style, either defined by style number, or style name.

Styles

Paragraph styles are defined using normal character/paragraph formatting sequence with

$$*number*,*nnumber*#*uuid*:*name*

instead of text, where

number	Number of style - can be used with s paragraph format command code.
nnumber	Number of style of next paragraph - used by RichText editor when inserting paragraphs.
uuid	32 digit unique hexadecimal identifier of style.
name	Name of style, displayed by editors. Can also be used with s paragraph format command code.

Style with number = 0 and uuid = 00000000000000000000000000000000 is ***default*** style.

Example:

```
"[*/+117 $$2,0#07143242482611002448121871408047:title]"
```

Objects

Object plays the role of the single character and is displayed according to its type. It is started with a header in the form

@@*format*:*cx*&*cy*

or

@@*format*:*cx***cy*

where

format	Format of objects. This format must be recognized by the application. By default, RichText recognizes the PNG format and ***iml*** format (see bellow).
cx	Width of object in dots.

cy	Height of object in dots.

First form with '&' activates "keep aspect ratio" for the object, second form with '*' leaves this option inactive.

Optionally, there can be 3rd number, separated by '/' character. This number represent vertical placement of object, default value 0 means that bottom border of object is aligned with baseline of text.

If header is followed by '`' character, object data are in text format. In that case, object data are terminated with another '`' character. If there needs to be '`' in the text, '``' can be used as escape sequence.

If header is followed by '(', it countains BASE64 encoded binary data, ending with ')'. **This is default format for binary data**.

If there is no '`' nor '(', header is in binary 7 bit format. Bit 7 of data bytes is always 1, so that actual data bytes are in range 128-255. First byte in range 32-127 ends data sequence. Data are encoded in 7 byte groups, which corresponds to 8 bytes of encoded format. First byte of this 8 bytes block always contains eight bits of following bytes, LSB (that is bit 0) being the eight bit for first byte in block. *This format is deprecated.*

***iml* format**

iml format is text format of rich object where text data reference existing .iml based Image as pair iml_class_name:image_name. Example of full object definition in ***iml*** format:

```
"@@iml:400*400`CtrlImg:exclamation`"
```

Fields

Fields are special elements of text that are evaluated by client code into rich text. QTF format for fields is

{:*field_type_id*:*parameter*:}

field_type_id	Type of field. Field types are represented by RichPara::FieldType derived instances and registred using RichPara::Register function.
parameter	Additional string parameter that gets passed to FieldType::Evaluate method

Tables

Table definition starts with

{{

pair, followed by set of numbers separated with

:

Numbers represent ratios of column widths; count of numbers is equivalent to count of columns. Next there is *table/cell formatting sequence* ended with single space character. Cells are separated with

::

characters and another table/cell formating sequence (to setup format for each individual cell). Formating of cells is inherited from previous cells. Table ends with

}}

pair.

Table/cell formating sequence	
<*number*	Left margin of table in dots.
>*number*	Right margin of table in dots.
B*number*	Space before table in dots.
A*number*	Space after table in dots.
f*number*	Frame thickness in dots. Frame is outer border of table. Default value is 10.
F*color*	Color of the frame.
g*number*	Grid thickness in dots. Grid are lines dividing cells inside table. Default value is 4.
G*color*	Color of the grid.
k	Keep the cell on single page.
K	Keep the table on single page.
~	Sets grid and frame thickness to zero. Useful when using tables to organize text.
h*number*	Number of header rows. Header rows are repeated at the beginning of every page.
^	Cell aligns to top.
=	Cell aligns to center (vertical). Default.
v	Cell aligns to bottom.
l*number/number* l*number* l*/number*	Sets left cell border (first *number*) and margin in dots. If any of numbers is missing, sets only the one present. Default is border: 0, margin: 25.
r*number/number* r*number* r*/number*	Sets right cell border (first *number*) and margin in dots. If any of numbers is missing, sets only the one present. Default is border: 0, margin: 25.
t*number/number* t*number*	Sets top cell border (first *number*) and margin in dots. If any of

t/*number*	numbers is missing, sets only the one present. Default is border: 0, margin: 15.
b*number*/*number* b*number* b/*number*	Sets bottom cell border (first *number*) and margin in dots. If any of numbers is missing, sets only the one present. Default is border: 0, margin: 15.
a*number*/*number* a*number* a/*number*	Sets all cell borders (first *number*) and margins in dots. If any of numbers is missing, sets only the one present.
@*color*	Cell background color. Default is White.
R*color*	Cell border color. Default is Black.
!	Resets cell formatting to default values.
H*number*	Sets the minimal height of cell (and therefore also of row) in dots.
-*number*	Horizontal cell span.
|*number*	Vertical cell span.
;	NOP separator. In some cases it helps to separate command code.

Note: There is also legacy support for old table format (from previous QTF version) that is based on ++ pair as table start/stop and || -- to divide cells/lines.

Header and Footer

Global text header is defined using ^H*qtf_text*^^, footer ^F*qtf_text*^^, where H*qtf_text* is complete embeded QTF representing header/footer. This QTF can contain field {:VALUE:PAGENUMBER:} to represent page number and {:VALUE:PAGECOUNT:} to represent total number of pages.

Examples

```
"Normal [* bold] [/ italic] [_ underline] [` superscript] [, subscript]"
```

Normal **bold** *italic* underline ^{superscript} $_{subscript}$

```
"`[ `] \1[escaped]\1 [* bold]"
```

[ ] [escaped] **bold**

```
"[A Arial (Sans-Serif)] [R Times New Roman (Serif)] [C Courier (Monospace)]"
```

Arial (Sans-Serif) Times New Roman (Serif) Courier (Monospace)

```
"[0 6pt ][1 8pt ][2 10pt ][3 12pt ][4 16pt ][5 20pt ][6 24pt ][7 28pt ][8 36pt ][9 48pt ]"
```

6pt 8pt 10pt 12pt 16pt 20pt 24pt 28pt 36pt 48pt

```
"[!Tahoma! Tahoma]"
```

Tahoma

```
"[+500 500dots]"
```

500dots

```
"[@4 Green text] [$(255.220.200) Pink background]"
```

Green text Pink background

```
"[%EN-US English language] [%CS-CZ Czech language]"
```

English language Czech language

```
"[^upp.sf.net^ Hyperlink] [Icompiler, linker; Index entry]"
```

Hyperlink Index entry

```
"[:label: Labeled paragraph]"
```

Labeled paragraph

```
"[< Left paragraph alignment]"
```

Left paragraph alignment

```
"[= Center paragraph alignment]"
```

Center paragraph alignment

```
"[> Right paragraph alignment]"
```

Right paragraph alignment

```
"[# Justify alignment. Just some text to demosntrate it... Just some text to demonstrate it...
Just some text to demonstrate it...]"
```

Justify alignment. Just some text to demosntrate it... Just some text to demonstrate it... Just some text to demonstrate it...

```
"[l1000 Left margin 1000dots]"
```

Left margin 1000dots

```
"[i1000 Indent 1000 dots.Just some text to demonstrate it... Just some text to demonstrate
it...]"
```

Indent 1000 dots.Just some text to demonstrate it... Just some text to demonstrate it...

```
"[r1000 Right margin 1000 dots.Just some text to demonstrate it... Just some text to demonstrate
it...]"
```

Right margin 1000 dots.Just some text to demonstrate it... Just some text to demonstrate it...

```
"Paragraph&[b200 Before 200dots]"
```

Paragraph
Before 200dots

```
"[a200 After 200dots]&Paragraph"
```

After 200dots

Paragraph

```
"[i200 [O0 bullet&][O1 bullet&][O2 bullet&][O3 bullet]]"
```

- bullet
- bullet
- bullet

bullet

```
"[O9i500 text bullet-|Just some text to demonstrate it... Just some text to demonstrate it...Just some text to demonstrate it... Just some text to demonstrate it...]"
```

text bullet Just some text to demonstrate it... Just some text to demonstrate it...Just some text to demonstrate it... Just some text to demonstrate it...

```
"[O9i200 [N1m.; -|level 1&][N1a -|level 2&][N1a -|level 2&][N1 -|level 1&][N1a -|level 2]]"
```

1. level 1
1.a level 2
1.b level 2
2 level 1
level 2

```
"[~300~=.2000~>-3000 -|Normal tab-|Centered tab-|Right tab]"
```

Normal tab Centered tabRight tab

```
"{{1:2 A1||A2||B1||B2}}"
```

A1	A2
B1	B2

```
"{{2:1G4g100F5f50 A1:: A2:: B1:: B2}}"
```

A1	A2
B1	B2

```
"{{1:2 A1::l40/60R6@3 A2::! B1:: B2}}"
```

A1	A2
B1	B2

```
"{{1:1:1|2 A1::-2 A2:: A3:: B1:: B2:: B3}}"
```

A1	A2	
	B2	B3

"{{1:2 A1:: A2:: B1:: {{1:2 a1:: a2:: a1:: a2}}}}"

A1	A2	
B1	a1	a2
	a1	a2

TEST

Test

Test