1. Rendering single line of text.

Images and other embedded objects sholud respect BiDi reordering too.
Replace them with "Object Replacement Character".

Input

Unicode string: (UTF-16)

T	e	s	t	␣	Ξ	ε	σ	κ	ε	π	ά	ζ	ω	␣	ر	ح	ي	م	␣	ر	ح	ي	م	␣	e	obj	t	e	◌́	n	d	↵	i	t	ş	ţ	ş	ţ	
0054	0065	0073	0074	0020	039E	03B5	03C3	03BA	03B5	03C0	03AC	03B6	03C9	0020	0631	062D	064A	0645	0020	0631	062D	064A	0645	0020	0065	FFFC	0074	0065	0301	006E	0064	000A	0069	0074	015F	0162	015F	0162	0000

Base direction:

TEXT_DIRECTION_AUTO
Text direction is detected based on first encountered code point (AUTO), current locale (LOCALE) or specified by user. Base direction is used to initialize BiDi iterator .

OpenType font features: (optional*)

OT features tags

dlig	{none}

Font: (optional*)

Sans	Nastaliq	Sans

Language: (optional*)

Language tags

{auto}	en	ro

* in case of plain (non attributed) string font/language and features are specified for entire string.

1.1. Split the string into BiDi and Script runs, then combine attribute/bidi and script runs into monotone runs.

BiDi Runs:

LTR, 0	RTL, 1	LTR, 2

BiDi

Script Runs:

Script tags

Latn	Grek	Arab	Latn

Script Iterator

1.1.1. Combine into monotone runs.

Monotone Runs:

Order:	0	1	3	2	4	5	6	7
Script:	Latn	Grek	Arab	Arab	Arab	Latn	Latn	Latn
Direction:	LTR	LTR	RTL	RTL	LTR	LTR	LTR	LTR
Language:	{auto}	{auto}	{auto}	{auto}	{auto}	{auto}	en	ro
Font:	Sans	Sans	Sans	Nastaliq	Nastaliq	Sans	Sans	Sans
Font features:	dlig	{none}	{none}	{none}	{none}	{none}	{none}	{none}

1.2. Shape runs in the order provided by BiDi algorithm.

HarfBuzz

Glyph

Ligature forced by "dlig" OpenType feature tag

T	e	st	3
37	48	356	3
0...0	1...1	2...3	4...4

Input string code units

Grapheme cluster

Shaping failed (some returned glyphs have index "0").

1.2.1. Shape runs with fallback font.

0	0	0	0	π	0	0	0	0	3
				71					
5...5	6...6	7...7	8...8	9...9	10...10	11...11	12...12	13...13	14...14

HarfBuzz

Ξ	ε	σ	κ	π	ε	ά	ζ	ω	3
16b	181	18f	186	181	18c	178	182	195	
5...5	6...6	7...7	8...8	9...9	10...10	11...11	12...12	13...13	14...14

118	- 13	◌ 183	3d7	◌ 1eb	◌ ec
23...23	22...22		21...21		20...20

3	◌ 1ed	◌ 1fa	◌ 1c0	◌ 31
19...19	18...18	17...17	16...16	15...15

3
24...24

Multiple code points combined into one cluster

No cluster for control code point (\n)

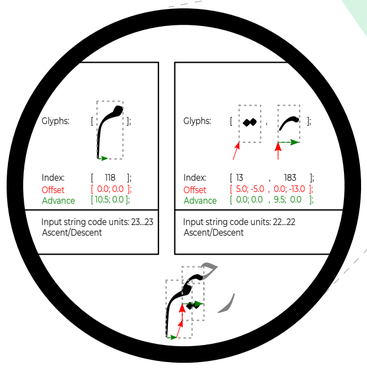
e		t	é	n	d	i	t
48	0	57	ab	51	47	4c	57
25...25	26...26	27...27	28...29	30...30	31...31	33...33	34...34

Replace with actual image

ş	ţ
64	67
35...35	36...36

Same code point can be shaped into different glyphs!

ş	ţ
74	87
37...37	38...38

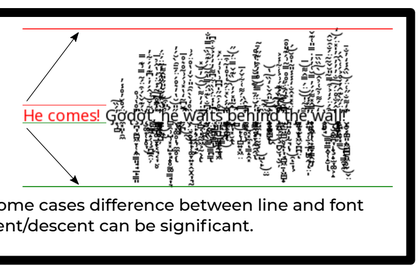


HarfBuzz returns font specific glyph indicies, offsets, advanced and character ranges that corrisponds to the cluster.
Ascent and descent of cluster is calculated based on the font ascent/descent and glyph offsets.

"HarfBuzz is a text-shaping engine. If you give HarfBuzz a font and a string containing a sequence of Unicode code points, HarfBuzz selects and positions the corresponding glyphs from the font, applying all of the necessary layout rules and font features. HarfBuzz then returns the string to you in the form that is correctly arranged for the language and writing system."
What is HarfBuzz?

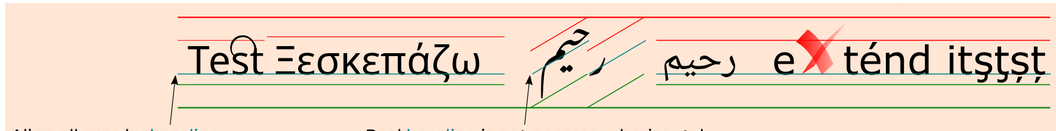
ICU BiDi API References
UAX #9: The Bidirectional Algorithm

Character - ambiguous term, can mean anything
Code unit - atomic unit of the encodeing (16-bits in case of UTF-16), represent a full code point, or part of a code point
Code point - atomic unit of the Unicode standard (independent of encoding)
Glyph - an image, atomic element of the font (font specific, doesn't have any meaning outside of the font)
Grapheme cluster - atomic unit of a writing system



In some cases difference between line and font ascent/descent can be significant.

1.3. Baseline align, ascent and descent.



Align all runs by baseline

Real baseline is not necessary horizontal

Calculate line ascent and descent, counting all glyph offset

T	e	st	3	Ξ	ε	σ	κ	π	ε	ά	ζ	ω	3	◌	-	◌	◌	◌	◌	◌	◌	3	◌	◌	◌	◌	3	e	x	t	é	n	d	i	t	ş	ţ	ş	ţ
37	48	356	3	16b	181	18f	186	181	18c	178	182	195	3	118	13	183	3d7	1eb	ec	3	1ed	1fa	1c0	31	3	48	5b	57	ab	51	47	4c	57	64	67	74	87
0...0	1...1	2...3	4...4	5...5	6...6	7...7	8...8	9...9	10...10	11...11	12...12	13...13	14...14	23...23	22...22		21...21		20...20	19...19	18...18	17...17	16...16	15...15	24...24	25...25	26...26	27...27	28...29	30...30	31...31	33...33	34...34	35...35	36...36	37...37	38...38

1.4. Draw glyphs left-to-right using advances and offsets of the glyphs.

Output

Test Ξεσκεπάζω رحيم رحيم e xténd itşţşţ