
libgdtl Documentation

Release latest

Nov 01, 2019

1	libgdtl Introduction	1
2	Installation	3
3	Font substitution	5
4	libgdtl API	7

BiDi, shaping and basic text layout for Godot Engine.

1.1 Dependencies

- Godot 3.1+
- C++14 compiler
- Meson build system (for gdnative module build only)
- SCons build system

1.2 Compiling (as builtin module)

1.2.1 Build options

Name	Description	Default value
<code>builtin_runtime</code>	Use the built-in libraries	true
<code>use_graphite2</code>	Enable SIL Graphite 2 complementary shaper	true
<code>use_font_wrapper</code>	Enable Godot font wrapper for default controls	false

If `use_font_wrapper` is enabled, apply `patch_font.diff` from the root of this repository to the Godot engine source first.

1.2.2 Building *libdgtl* module

Clone this repository (without `-recursive` flag) into Godot's `modules` subfolder as `godot_tl`. Rebuild Godot engine as usual.

1.3 Compiling (as gdnative module)

1.3.1 Build options

Name	Description	Default value
godot-cpp-lib-name	godot-cpp static library name (without <i>.a</i> or <i>.lib</i> extension)	libgodot-cpp
static-runtime	Link libraries statically for better portability	false
builtin-runtime	Use the built-in libraries	false
use-graphite2	Enable SIL Graphite 2 complementary shaper	true

1.3.2 Building *godot-cpp* static library

See <https://github.com/GodotNativeTools/godot-cpp/blob/master/README.md#compiling-the-cpp-bindings-library>

1.3.3 Building *libgdtl* module

You can compile this module by executing:

```
meson {Target-Folder} -Dgodot-cpp-lib-name={Godot-CPP-Name} --buildtype=release
ninja -C {Target-Folder}
```

1.4 License

- The source code of the **libgdtl** module is released under unlicense.
For more information, see <http://unlicense.org/> or the accompanying UNLICENSE file.
- **Godot** and **GodotNativeTools** are licensed under MIT license.
For more information, see <https://github.com/godotengine/godot/blob/master/LICENSE.txt>.
- **HarfBuzz** is licensed under MIT-like License.
For more information, see <https://github.com/harfbuzz/harfbuzz/blob/master/COPYING>
- **ICU4C** is licensed under Unicode, Inc. License.
For more information, see <http://www.unicode.org/copyright.html#License>
- **FreeType** is licensed under FreeType License (BSD-like) or GNU General Public License (GPL), version 2.
For more information, see <https://www.freetype.org/license.html>
- **SIL Graphite engine** is licensed under GNU Lesser General Public License (LGPL), version 2.1+ or GNU General Public License (GPL), version 2 or Mozilla Public License.
For more information, see <https://github.com/silnrsi/graphite/blob/master/COPYING>

1.5 Demo data

Montserrat (<https://github.com/JuliettaUla/Montserrat/>), **Awami** **Nastaliq** (<https://software.sil.org/awami/download/>), **Comic Neue** (<http://comicneue.com/>) and **Noto** (<https://www.google.com/get/noto/>) fonts are published under the **SIL Open Font License, Version 1.1** (https://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&id=OFL)

Material Design icons by **Google** (<https://github.com/google/material-design-icons>) are published under the **Apache License Version 2.0** (<https://www.apache.org/licenses/LICENSE-2.0.txt>)

Noto Color Emoji font is cut down to single glyph (U+1F604) using **glyphhanger** (<https://github.com/filamentgroup/glyphhanger>).

2.1 Module

Use Godot editor and export templates compiled with the module.

Module is ready to use, no modification to the project is required.

2.2 GDNative

GDNative plugin is intended to be used with official Godot editor and export templates.

To install plugin:

1. Create *addons* folder in the root of your project.
2. Copy the contents of archive to the *addons* folder. (Do not drag-and-drop it into *FileSystem* tab of the editor. Right-click *addons* folder, select *Open in File Manager* and use your OS file manager to copy/extract files.)
3. After installing plugin, your file system should look like this:
 - [addons]
 - [libgdtl]
 - * [bin]
 - * [classes]
 - * [icons]
 - * constants.gd
 - * gdtl.gd
 - * libgdtl.gdnlb
 - * plugin.cfg
 - * tl_font_family_edit.gd
 - * tl_font_family_preview.gd
4. Go to *Project Settings*, click on the *Plugins* tab and activate *gdtl* plugin.

Font substitution

Font substitution is used to find a replacement for an unavailable character.

In addition to main list of the substitution fonts, you can specify preferred fonts for the script (writing system) and language.

Use [ISO 639-1](https://en.wikipedia.org/wiki/ISO_639-1) (https://en.wikipedia.org/wiki/ISO_639-1) codes for the language names, and [ISO 15924](https://en.wikipedia.org/wiki/ISO_15924) (https://en.wikipedia.org/wiki/ISO_15924) for the script names.

Substitution lists have following priority: *Language, Script, Main List*. Providing script and language information is not required, but can improve shaping speed if large number of fonts used.

4.1 TLBitmapFontFace

Inherits: *TLFontFace*

Category: Core

4.1.1 Brief Description

An AngelCode Bitmap Font Generator bitmap font for drawing text.

4.1.2 Properties

int	texture_flags	O: 2048
-----	---------------	----------------

4.1.3 Description

TLBitmapFontFace have limited shaping support.

TLBitmapFontFace doesn't support OpenType features.

4.2 TLDynamicFontFace

Inherits: *TLFontFace*

Category: Core

4.2.1 Brief Description

A TrueType, OpenType or Graphite font for drawing text.

4.2.2 Properties

bool	<i>force_automhinter</i>	false
int	<i>hinting</i>	2
float	<i>oversampling</i>	1.0
int	texture_flags	O: 2048

4.2.3 Methods

bool	<i>has__graphite</i> () const
------	--------------------------------

4.2.4 Enumerations

enum **DynamicFaceHinting**:

- **DF_HINTING_NONE** = **0** — Disable font hinting (smoother but less crisp)
- **DF_HINTING_LIGHT** = **1** — Use the light font hinting mode
- **DF_HINTING_NORMAL** = **2** — Use the default font hinting mode (crisper but less smooth)

4.2.5 Property Descriptions

- bool **force__autohinter**

<i>Default</i>	false
<i>Setter</i>	set_force__autohinter(value)
<i>Getter</i>	get_force__autohinter()

If **true**, prefers FreeType auto-hinter over the font’s native hinter. Default: **false**

- int **hinting**

<i>Default</i>	2
<i>Setter</i>	set_hinting(value)
<i>Getter</i>	get_hinting()

The font hinting mode used by FreeType auto-hinter. Default: **DF_HINTING_NONE**

- float **oversampling**

<i>Default</i>	1.0
<i>Setter</i>	set_oversampling(value)
<i>Getter</i>	get_oversampling()

Font oversampling factor. Default: 1.0

4.2.6 Method Descriptions

- bool **has__graphite** () const

Returns **true** if module is built with SIL Graphite 2 shaper support.

4.3 TLFontFace

Inherits:

Inherited By: *TLBitmapFontFace*, *TLDynamicFontFace*

Category: Core

4.3.1 Brief Description

Virtual class

A base font face class.

4.3.2 Properties

String	<i>font_path</i>	“”
int	<i>texture_flags</i>	0

4.3.3 Methods

void	<i>draw_glyph</i> (RID canvas_item, Vector2 pos, int codepoint, Color modulate, int size) const
void	<i>draw_glyph_outline</i> (RID canvas_item, Vector2 pos, int codepoint, Color modulate, int size) const
float	<i>get_ascent</i> (int size) const
int	<i>get_base_size</i> () const
float	<i>get_descent</i> (int size) const
Array	<i>get_glyph_outline</i> (Vector2 pos, int codepoint, int size) const
float	<i>get_height</i> (int size) const
bool	<i>load</i> (String resource_path)
Array	<i>unicode_scripts_supported</i> () const

4.3.4 Property Descriptions

- String **font_path**

<i>Default</i>	“”
<i>Setter</i>	set_font_path(value)
<i>Getter</i>	get_font_path()

- int **texture_flags**

<i>Default</i>	0
<i>Setter</i>	set_texture_flags(value)
<i>Getter</i>	get_texture_flags()

Font texture flags. Default: FLAG_VIDEO_SURFACE

4.3.5 Method Descriptions

- void **draw_glyph** (RID canvas_item, Vector2 pos, int codepoint, Color modulate, int size) const

Draws a single glyph.

- void **draw_glyph_outline** (RID canvas_item, Vector2 pos, int codepoint, Color modulate, int size) const

Draws single glyph outline.

- float **get_ascent** (int size) const

Returns ascent (distance from the baseline to the highest position characters extend to) of the font.

- int **get__base__size** () const

Returns default font size for bitmap fonts or 0 for dynamic fonts.

- float **get__descent** (int size) const

Returns descent (distance from the base line to the lowest point characters extend to) of the font.

- Array **get__glyph__outline** (Vector2 pos, int codepoint, int size) const
-

- float **get__height** (int size) const

Returns height (vertical distance between two consecutive baselines) of the font.

- bool **load** (String resource_path)

Loads font from speified file.

- Array **unicode__scripts__supported** () const

4.4 TLFontFamily

Inherits:

Category: Core

4.4.1 Brief Description

A set of fonts that make up a font family.

4.4.2 Methods

void	<i>add__face</i> (String style, <i>TLFontFace</i> ref)
void	<i>add__face__for__language</i> (String style, <i>TLFontFace</i> ref, String lang)
void	<i>add__face__for__script</i> (String style, <i>TLFontFace</i> ref, String script)
void	<i>add__face__unlinked</i> (String style, <i>TLFontFace</i> ref)
void	<i>add__language</i> (String style, String language)
void	<i>add__script</i> (String style, String script)
void	<i>add__style</i> (String style)
<i>TLFontIterator</i>	<i>get__face</i> (String style) const
<i>TLFontIterator</i>	<i>get__face__for__language</i> (String style, String lang) const
<i>TLFontIterator</i>	<i>get__face__for__script</i> (String style, String script) const
bool	<i>has__style</i> (String style) const
void	<i>remove__language</i> (String style, String language)
void	<i>remove__script</i> (String style, String script)
void	<i>remove__style</i> (String style)

4.4.3 Method Descriptions

- void **add__face** (String style, *TLFontFace* ref)
-
- void **add__face__for__language** (String style, *TLFontFace* ref, String lang)
-
- void **add__face__for__script** (String style, *TLFontFace* ref, String script)
-
- void **add__face__unlinked** (String style, *TLFontFace* ref)
-
- void **add__language** (String style, String language)
-
- void **add__script** (String style, String script)
-
- void **add__style** (String style)
-
- *TLFontIterator* **get__face** (String style) const
-
- *TLFontIterator* **get__face__for__language** (String style, String lang) const
-
- *TLFontIterator* **get__face__for__script** (String style, String script) const
-
- bool **has__style** (String style) const
-
- void **remove__language** (String style, String language)
-
- void **remove__script** (String style, String script)
-
- void **remove__style** (String style)

4.5 TLFontIterator

Inherits:

Category: Core

4.5.1 Brief Description

4.6 TLGDFontWrapper

Inherits:

Category: Core

4.6.1 Brief Description

4.6.2 Properties

<i>TLFontFamily</i>	<i>base_font</i>	
int	<i>base_font_size</i>	12
String	<i>base_font_style</i>	“Regular”
int	<i>cache_depth</i>	100

4.6.3 Description

Note: This class is only available if module is built with `use_font_wrapper=true` flag.

4.6.4 Property Descriptions

- *TLFontFamily* **base_font**

<i>Setter</i>	<code>set_base_font(value)</code>
<i>Getter</i>	<code>get_base_font()</code>

- int **base_font_size**

<i>Default</i>	12
<i>Setter</i>	<code>set_base_font_size(value)</code>
<i>Getter</i>	<code>get_base_font_size()</code>

- String **base_font_style**

<i>Default</i>	“Regular”
<i>Setter</i>	<code>set_base_font_style(value)</code>
<i>Getter</i>	<code>get_base_font_style()</code>

- int **cache_depth**

<i>Default</i>	100
<i>Setter</i>	<code>set_cache_depth(value)</code>
<i>Getter</i>	<code>get_cache_depth()</code>

4.7 TLICUDataLoader

Inherits:

Category: Core

4.7.1 Brief Description

Helper class that handles ICU data loading.

4.7.2 Properties

String	<i>data_path</i>	“”
--------	------------------	----

4.7.3 Methods

bool	<i>load</i> (String resource_path)
------	--------------------------------------

4.7.4 Property Descriptions

- String *data_path*

<i>Default</i>	“”
<i>Setter</i>	set_data_path(value)
<i>Getter</i>	get_data_path()

4.7.5 Method Descriptions

- bool *load* (String resource_path)

Loads ICU data file, should be done at most once in a process, before the first ICU operation. Returns `true` if function succeeds.

4.8 TLLabel

Inherits:

Category: Core

4.8.1 Brief Description

4.8.2 Properties

int	<i>align</i>	0
bool	<i>autowrap</i>	false
<i>TLFontFamily</i>	<i>base_font</i>	
int	<i>base_font_size</i>	12
String	<i>base_font_style</i>	“Regular”
bool	<i>clip_text</i>	false
String	<i>language</i>	“”
Control.MouseFilter	<i>mouse_filter</i>	O: 2
String	<i>ot_features</i>	“”
int	<i>size_flags_vertical</i>	O: 4
String	<i>text</i>	“”
int	<i>text_direction</i>	3
bool	<i>uppercase</i>	false
int	<i>valign</i>	0

4.8.3 Methods

int	<i>get_line_count</i> () const
int	<i>get_line_height</i> () const
int	<i>get_lines_skipped</i> () const
int	<i>get_max_lines_visible</i> () const

Continued on next page

Table 15 – continued from previous page

float	<i>get_percent_visible</i> () const
int	<i>get_total_character_count</i> () const
int	<i>get_visible_characters</i> () const
int	<i>get_visible_line_count</i> () const
void	<i>set_lines_skipped</i> (int lines_skipped)
void	<i>set_max_lines_visible</i> (int lines_visible)
void	<i>set_percent_visible</i> (float percent_visible)
void	<i>set_visible_characters</i> (int amount)

4.8.4 Enumerations

enum **Align**:

- **ALIGN_LEFT** = 0
 - **ALIGN_CENTER** = 1
 - **ALIGN_RIGHT** = 2
 - **ALIGN_FILL** = 3
-

enum **Valign**:

- **VALIGN_TOP** = 0
- **VALIGN_CENTER** = 1
- **VALIGN_BOTTOM** = 2
- **VALIGN_FILL** = 3

4.8.5 Property Descriptions

- int **align**

<i>Default</i>	0
<i>Setter</i>	<i>set_align</i> (value)
<i>Getter</i>	<i>get_align</i> ()

- bool **autowrap**

<i>Default</i>	false
<i>Setter</i>	<i>set_autowrap</i> (value)
<i>Getter</i>	<i>has_autowrap</i> ()

- *TLFontFamily* **base_font**

<i>Setter</i>	<i>set_base_font</i> (value)
<i>Getter</i>	<i>get_base_font</i> ()

- int **base_font_size**

<i>Default</i>	12
<i>Setter</i>	set_base_font_size(value)
<i>Getter</i>	get_base_font_size()

- String **base_font_style**

<i>Default</i>	“Regular”
<i>Setter</i>	set_base_font_style(value)
<i>Getter</i>	get_base_font_style()

- bool **clip_text**

<i>Default</i>	false
<i>Setter</i>	set_clip_text(value)
<i>Getter</i>	is_clipping_text()

- String **language**

<i>Default</i>	“”
<i>Setter</i>	set_language(value)
<i>Getter</i>	get_language()

- String **ot_features**

<i>Default</i>	“”
<i>Setter</i>	set_ot_features(value)
<i>Getter</i>	get_ot_features()

- String **text**

<i>Default</i>	“”
<i>Setter</i>	set_text(value)
<i>Getter</i>	get_text()

- int **text_direction**

<i>Default</i>	3
<i>Setter</i>	set_text_direction(value)
<i>Getter</i>	get_text_direction()

- bool **uppercase**

<i>Default</i>	false
<i>Setter</i>	set_uppercase(value)
<i>Getter</i>	is_uppercase()

- int **valign**

<i>Default</i>	0
<i>Setter</i>	set_valign(value)
<i>Getter</i>	get_valign()

4.8.6 Method Descriptions

- int **get_line_count** () const
-

- int **get_line_height** () const
-

- int **get_lines_skipped** () const
-

- int **get_max_lines_visible** () const
-

- float **get_percent_visible** () const
-

- int **get_total_character_count** () const
-

- int **get_visible_characters** () const
-

- int **get_visible_line_count** () const
-

- void **set_lines_skipped** (int lines_skipped)
-

- void **set_max_lines_visible** (int lines_visible)
-

- void **set_percent_visible** (float percent_visible)
-

- void **set_visible_characters** (int amount)
-

4.9 TLLineEdit

Inherits:

Category: Core

4.9.1 Brief Description

4.9.2 Properties

int	<i>align</i>	0
<i>TlFontFamily</i>	<i>base_font</i>	
int	<i>base_font_size</i>	12
String	<i>base_font_style</i>	“Regular”
bool	<i>caret_blink</i>	false
float	<i>caret_blink_speed</i>	0.65
int	<i>caret_position</i>	0
bool	<i>clear_button_enabled</i>	false
bool	<i>context_menu_enabled</i>	true
bool	<i>editable</i>	true
bool	<i>expand_to_text_length</i>	false
Control.FocusMode	<i>focus_mode</i>	O: 2
String	<i>language</i>	“”
int	<i>max_length</i>	0
Control.CursorShape	<i>mouse_default_cursor_shape</i>	O: 1
String	<i>ot_features</i>	“”
float	<i>placeholder_alpha</i>	0.6
String	<i>placeholder_text</i>	“”
bool	<i>secret</i>	false
String	<i>secret_character</i>	“*”
String	<i>text</i>	“”
int	<i>text_direction</i>	3

4.9.3 Methods

void	<i>append_at_cursor</i> (String text)
void	<i>clear</i> ()
void	<i>deselect</i> ()
PopupMenu	<i>get_menu</i> () const
void	<i>menu_option</i> (int option)
void	<i>select</i> (int from=0, int to=-1)
void	<i>select_all</i> ()

4.9.4 Signals

- **text_changed** (String new_text)

- **text_entered** (String new_text)

4.9.5 Enumerations

enum **Align**:

- **ALIGN_LEFT** = 0
- **ALIGN_CENTER** = 1
- **ALIGN_RIGHT** = 2
- **ALIGN_FILL** = 3

enum **MenuItems**:

- `MENU__CUT = 0`
- `MENU__COPY = 1`
- `MENU__PASTE = 2`
- `MENU__CLEAR = 3`
- `MENU__SELECT__ALL = 4`
- `MENU__UNDO = 5`
- `MENU__REDO = 6`
- `MENU__MAX = 7`

4.9.6 Property Descriptions

- `int align`

<i>Default</i>	0
<i>Setter</i>	<code>set_align(value)</code>
<i>Getter</i>	<code>get_align()</code>

- *TLFontFamily* `base_font`

<i>Setter</i>	<code>set_base_font(value)</code>
<i>Getter</i>	<code>get_base_font()</code>

- `int base_font_size`

<i>Default</i>	12
<i>Setter</i>	<code>set_base_font_size(value)</code>
<i>Getter</i>	<code>get_base_font_size()</code>

- `String base_font_style`

<i>Default</i>	“Regular”
<i>Setter</i>	<code>set_base_font_style(value)</code>
<i>Getter</i>	<code>get_base_font_style()</code>

- `bool caret_blink`

<i>Default</i>	false
<i>Setter</i>	<code>cursor_set_blink_enabled(value)</code>
<i>Getter</i>	<code>cursor_get_blink_enabled()</code>

- `float caret_blink_speed`

<i>Default</i>	0.65
<i>Setter</i>	cursor_set_blink_speed(value)
<i>Getter</i>	cursor_get_blink_speed()

- int **caret_position**

<i>Default</i>	0
<i>Setter</i>	set_cursor_position(value)
<i>Getter</i>	get_cursor_position()

- bool **clear_button_enabled**

<i>Default</i>	false
<i>Setter</i>	set_clear_button_enabled(value)
<i>Getter</i>	is_clear_button_enabled()

- bool **context_menu_enabled**

<i>Default</i>	true
<i>Setter</i>	set_context_menu_enabled(value)
<i>Getter</i>	is_context_menu_enabled()

- bool **editable**

<i>Default</i>	true
<i>Setter</i>	set_editable(value)
<i>Getter</i>	is_editable()

- bool **expand_to_text_length**

<i>Default</i>	false
<i>Setter</i>	set_expand_to_text_length(value)
<i>Getter</i>	get_expand_to_text_length()

- String **language**

<i>Default</i>	""
<i>Setter</i>	set_language(value)
<i>Getter</i>	get_language()

- int **max_length**

<i>Default</i>	0
<i>Setter</i>	set_max_length(value)
<i>Getter</i>	get_max_length()

- String **ot_features**

<i>Default</i>	""
<i>Setter</i>	set_ot_features(value)
<i>Getter</i>	get_ot_features()

- float **placeholder_alpha**

<i>Default</i>	0.6
<i>Setter</i>	set_placeholder_alpha(value)
<i>Getter</i>	get_placeholder_alpha()

- String **placeholder_text**

<i>Default</i>	""
<i>Setter</i>	set_placeholder(value)
<i>Getter</i>	get_placeholder()

- bool **secret**

<i>Default</i>	false
<i>Setter</i>	set_secret(value)
<i>Getter</i>	is_secret()

- String **secret_character**

<i>Default</i>	""*
<i>Setter</i>	set_secret_character(value)
<i>Getter</i>	get_secret_character()

- String **text**

<i>Default</i>	""
<i>Setter</i>	set_text(value)
<i>Getter</i>	get_text()

- int **text_direction**

<i>Default</i>	3
<i>Setter</i>	set_text_direction(value)
<i>Getter</i>	get_text_direction()

4.9.7 Method Descriptions

- void **append_at_cursor** (String text)

- void **clear** ()

- void **deselect** ()

- PopupMenu **get_menu** () const

- void **menu_option** (int option)

- void **select** (int from=0, int to=-1)

- void **select_all** ()

4.10 TLProtoControl

Inherits:

Category: Core

4.10.1 Brief Description

Rich text input control prototype.

4.10.2 Properties

Color	<i>back_color</i>	Color(1, 1, 1, 0)
Control.FocusMode	focus_mode	O: 2
float	<i>paragraph_spacing</i>	3.0
bool	<i>readonly</i>	false
bool	<i>selectable</i>	true

4.10.3 Methods

void	<i>add_attribute</i> (<i>TLProtoControlSelection</i> selection, int attribute, Variant value)
void	<i>clear</i> ()
void	<i>debug_draw</i> (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)
void	<i>debug_draw_as_hex</i> (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)

Continued on next page

Table 49 – continued from previous page

void	<i>debug_draw_logical_as_hex</i> (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)
Vector2	<i>get_caret_position</i> ()
String	<i>get_cluster_debug_info_hit_test</i> (Vector2 position)
Array	<i>get_cluster_glyphs_hit_test</i> (Vector2 position)
Rect2	<i>get_cluster_rect_hit_test</i> (Vector2 position)
<i>TLShapedParagraph</i>	<i>get_paragraph</i> (int index) const
int	<i>get_paragraphs</i> ()
<i>TLProtoControlSelection</i>	<i>get_selection</i> () const
int	<i>insert_paragraph</i> (<i>TLShapedParagraph</i> para, int index)
void	<i>remove_attribute</i> (<i>TLProtoControlSelection</i> selection, int attribute)
void	<i>remove_attributes</i> (<i>TLProtoControlSelection</i> selection)
void	<i>remove_paragraph</i> (int index)
void	<i>replace_sstring</i> (<i>TLProtoControlSelection</i> selection, <i>TLShapedString</i> text)
void	<i>replace_text</i> (<i>TLProtoControlSelection</i> selection, String text)
void	<i>replace_utf16</i> (<i>TLProtoControlSelection</i> selection, PoolByteArray text)
void	<i>replace_utf32</i> (<i>TLProtoControlSelection</i> selection, PoolByteArray text)
void	<i>replace_utf8</i> (<i>TLProtoControlSelection</i> selection, PoolByteArray text)
void	<i>set_paragraph</i> (<i>TLShapedParagraph</i> para, int index)
void	<i>set_paragraph_back_color</i> (<i>TLProtoControlSelection</i> selection, Color bcolor)
void	<i>set_paragraph_brk_flags</i> (<i>TLProtoControlSelection</i> selection, int flags)
void	<i>set_paragraph_halign</i> (<i>TLProtoControlSelection</i> selection, int halign)
void	<i>set_paragraph_indent</i> (<i>TLProtoControlSelection</i> selection, float indent)
void	<i>set_paragraph_jst_flags</i> (<i>TLProtoControlSelection</i> selection, int flags)
void	<i>set_paragraph_line_spacing</i> (<i>TLProtoControlSelection</i> selection, float line_spacing)
void	<i>set_paragraph_width</i> (<i>TLProtoControlSelection</i> selection, float width)
void	<i>set_selection</i> (<i>TLProtoControlSelection</i> selection)

4.10.4 Signals

- **cursor_changed** ()
-

- **paragraph_changed** ()

4.10.5 Property Descriptions

- Color **back_color**

<i>Default</i>	Color(1, 1, 1, 0)
<i>Setter</i>	set_back_color(value)
<i>Getter</i>	get_back_color()

- float **paragraph_spacing**

<i>Default</i>	3.0
<i>Setter</i>	set_paragraph_spacing(value)
<i>Getter</i>	get_paragraph_spacing()

- bool **readonly**

<i>Default</i>	false
<i>Setter</i>	set_readonly(value)
<i>Getter</i>	get_readonly()

- bool **selectable**

<i>Default</i>	true
<i>Setter</i>	set_selectable(value)
<i>Getter</i>	get_selectable()

4.10.6 Method Descriptions

- void **add_attribute** (*TLProtoControlSelection* selection, int attribute, Variant value)

- void **clear** ()

- void **debug_draw** (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)

- void **debug_draw_as_hex** (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)

- void **debug_draw_logical_as_hex** (RID rid, Vector2 position, Vector2 hit_position, bool draw_brk_ops, bool draw_jst_ops)

- Vector2 **get_caret_position** ()

- String **get_cluster_debug_info_hit_test** (Vector2 position)

- Array **get_cluster_glyphs_hit_test** (Vector2 position)

- Rect2 **get_cluster_rect_hit_test** (Vector2 position)

- *TLShapedParagraph* **get_paragraph** (int index) const

- int **get_paragraphs** ()

- *TLProtoControlSelection* **get_selection** () const

-
- `int insert_paragraph (TLShapedParagraph para, int index)`
-
- `void remove_attribute (TLProtoControlSelection selection, int attribute)`
-
- `void remove_attributes (TLProtoControlSelection selection)`
-
- `void remove_paragraph (int index)`
-
- `void replace_sstring (TLProtoControlSelection selection, TLShapedString text)`
-
- `void replace_text (TLProtoControlSelection selection, String text)`
-
- `void replace_utf16 (TLProtoControlSelection selection, PoolByteArray text)`
-
- `void replace_utf32 (TLProtoControlSelection selection, PoolByteArray text)`
-
- `void replace_utf8 (TLProtoControlSelection selection, PoolByteArray text)`
-
- `void set_paragraph (TLShapedParagraph para, int index)`
-
- `void set_paragraph_back_color (TLProtoControlSelection selection, Color bcolor)`
-
- `void set_paragraph_brk_flags (TLProtoControlSelection selection, int flags)`
-
- `void set_paragraph_halign (TLProtoControlSelection selection, int halign)`
-
- `void set_paragraph_indent (TLProtoControlSelection selection, float indent)`
-
- `void set_paragraph_jst_flags (TLProtoControlSelection selection, int flags)`
-
- `void set_paragraph_line_spacing (TLProtoControlSelection selection, float line_spacing)`
-
- `void set_paragraph_width (TLProtoControlSelection selection, float width)`
-
- `void set_selection (TLProtoControlSelection selection)`
-

4.11 TLProtoControlSelection

Inherits:

Category: Core

4.11.1 Brief Description

4.11.2 Properties

int	<i>caret_offset</i>	0
int	<i>caret_para</i>	0
int	<i>end_offset</i>	0
int	<i>end_para</i>	0
int	<i>start_offset</i>	0
int	<i>start_para</i>	0

4.11.3 Signals

- `selection_changed ()`

4.11.4 Property Descriptions

- int `caret_offset`

<i>Default</i>	0
<i>Setter</i>	<code>set_caret_offset(value)</code>
<i>Getter</i>	<code>get_caret_offset()</code>

- int `caret_para`

<i>Default</i>	0
<i>Setter</i>	<code>set_caret_para(value)</code>
<i>Getter</i>	<code>get_caret_para()</code>

- int `end_offset`

<i>Default</i>	0
<i>Setter</i>	<code>set_end_offset(value)</code>
<i>Getter</i>	<code>get_end_offset()</code>

- int `end_para`

<i>Default</i>	0
<i>Setter</i>	<code>set_end_para(value)</code>
<i>Getter</i>	<code>get_end_para()</code>

- int `start_offset`

<i>Default</i>	0
<i>Setter</i>	set_start_offset(value)
<i>Getter</i>	get_start_offset()

- int **start__para**

<i>Default</i>	0
<i>Setter</i>	set_start_para(value)
<i>Getter</i>	get_start_para()

4.12 TLShapedAttributedString

Inherits: *TLShapedString*

Category: Core

4.12.1 Brief Description

Golds shaped line of text with associated attributes.

4.12.2 Methods

void	<i>add_attribute</i> (int attribute, Variant value, int start, int end)
void	<i>clear_attributes</i> ()
void	<i>commit_attribute</i> ()
Variant	<i>get_attribute</i> (int attribute, int index) const
int	<i>get_attribute_end</i> (int attribute, int index) const
int	<i>get_attribute_start</i> (int attribute, int index) const
Array	<i>get_embedded_rects</i> ()
bool	<i>has_attribute</i> (int attribute, int index) const
void	<i>load_attributes_dict</i> (Array array)
void	<i>remove_attribute</i> (int attribute, int start, int end)
void	<i>remove_attributes</i> (int start, int end)
Array	<i>save_attributes_dict</i> () const

4.12.3 Enumerations

enum **TextAttribute**:

- **TEXT_ATTRIBUTE_FONT** = **1** — Font family. Attribute type: *TLFontFamily*
- **TEXT_ATTRIBUTE_FONT_STYLE** = **2** — Font style (Regular, Bold, Italic, Oblique etc.). Attribute type: String
- **TEXT_ATTRIBUTE_FONT_SIZE** = **3** — Font size. Attribute type: int
- **TEXT_ATTRIBUTE_FONT_FEATURES** = **4** — Comma separated list of OpenType feature tags. More info: <https://docs.microsoft.com/en-us/typography/opentype/spec/featuretags>. Attribute type: String
- **TEXT_ATTRIBUTE_LANGUAGE** = **5** — Language code. Attribute type: String
- **TEXT_ATTRIBUTE_REPLACEMENT_IMAGE** = **6** — Embedded image. Attribute type: Texture
- **TEXT_ATTRIBUTE_REPLACEMENT_RECT** = **7** — Reserved space for custom embedded object. Attribute type: Vector2

- **TEXT_ATTRIBUTE_REPLACEMENT_ID = 8** — Embedded object id key. Attribute type: Variant
 - **TEXT_ATTRIBUTE_REPLACEMENT_VALIGN = 9** — Embedded image/object in-line alignment. Attribute type: TEXT_VALIGN_*
 - **TEXT_ATTRIBUTE_COLOR = 31** — Text color. Attribute type: Color
 - **TEXT_ATTRIBUTE_OUTLINE_COLOR = 32** — Text outline color. Attribute type: Color
 - **TEXT_ATTRIBUTE_UNDERLINE_COLOR = 41** — Underline color. Attribute type: Color
 - **TEXT_ATTRIBUTE_UNDERLINE_WIDTH = 42** — Underline width. Attribute type: int
 - **TEXT_ATTRIBUTE_STRIKETHROUGH_COLOR = 51** — Strikethrough line color. Attribute type: Color
 - **TEXT_ATTRIBUTE_STRIKETHROUGH_WIDTH = 52** — Strikethrough line width. Attribute type: int
 - **TEXT_ATTRIBUTE_OVERLINE_COLOR = 61** — Overline color/ Attribute type: Color
 - **TEXT_ATTRIBUTE_OVERLINE_WIDTH = 62** — Overline width. Attribute type: int
 - **TEXT_ATTRIBUTE_HIGHLIGHT_COLOR = 71** — Highlight color. Attribute type: Color
 - **TEXT_ATTRIBUTE_META = 100** — User defined data, use TEXT_ATTRIBUTE_META + x to define multiple user attributes. Attribute type: Variant
-

enum **TextVAlign**:

- **TEXT_VALIGN_TOP = 0** — Inline vertical top alignment
- **TEXT_VALIGN_CENTER = 1** — Inline vertical center alignment
- **TEXT_VALIGN_BOTTOM = 2** — Inline vertical bottom alignment

4.12.4 Method Descriptions

- void **add_attribute** (int attribute, Variant value, int start, int end)

Sets **attribute** attribute to **value** for specified text range.

- void **clear_attributes** ()

Removes all attributes.

- void **commit_attribute** ()
-

- Variant **get_attribute** (int attribute, int index) const

Returns **attribute** attribute value for specified text position.

- int **get_attribute_end** (int attribute, int index) const
-

Returns last position of **attribute** attribute run enclosing specified position.

- int **get__attribute__start** (int attribute, int index) const

Returns first position of **attribute** attribute run enclosing specified position.

- Array **get__embedded__rects** ()

Returns bounding rects of embedded objects (TEXT_ATTRIBUTE_REPLACEMENT_RECT attributes).

- bool **has__attribute** (int attribute, int index) const

Returns **true** if spceified position has **attribute** attribute set.

- void **load__attributes__dict** (Array array)

Loads attributes from Array of Dictionary.

- void **remove__attribute** (int attribute, int start, int end)

Removes **attribute** attribute for specified text range.

- void **remove__attributes** (int start, int end)

Removes all attributes for specified text range.

- Array **save__attributes__dict** () const

Stores string attributes into Array of Dictionary.

4.13 TLShapedParagraph

Inherits:

Category: Core

4.13.1 Brief Description

4.13.2 Properties

Color	<i>back_color</i>	Color(1, 1, 1, 0)
int	<i>brk_flags</i>	2
int	<i>halign</i>	0
float	<i>indent</i>	0.0
int	<i>jst_flags</i>	1
float	<i>line_spacing</i>	1.0
<i>TLShapedAttributedString</i>	<i>string</i>	
float	<i>width</i>	-1.0

4.13.3 Methods

void	<i>copy_properties</i> (<i>TLShapedParagraph</i> source)
<i>TLShapedAttributedString</i>	<i>get_line</i> (int index) const
Array	<i>get_line_bounds</i> () const
int	<i>get_lines</i> () const
Vector2	<i>get_size</i> () const
Array	<i>get_word_bounds</i> () const

4.13.4 Signals

- `paragraph_changed` ()

4.13.5 Enumerations

enum `ParaHAlign`:

- `PARA_HALIGN_LEFT` = 0
- `PARA_HALIGN_CENTER` = 1
- `PARA_HALIGN_RIGHT` = 2
- `PARA_HALIGN_FILL` = 3

4.13.6 Property Descriptions

- Color `back_color`

<i>Default</i>	<code>Color(1, 1, 1, 0)</code>
<i>Setter</i>	<code>set_back_color(value)</code>
<i>Getter</i>	<code>get_back_color()</code>

- int `brk_flags`

<i>Default</i>	2
<i>Setter</i>	<code>set_brk_flags(value)</code>
<i>Getter</i>	<code>get_brk_flags()</code>

- int `halign`

<i>Default</i>	0
<i>Setter</i>	<code>set_halign(value)</code>
<i>Getter</i>	<code>get_halign()</code>

- float `indent`

<i>Default</i>	0.0
<i>Setter</i>	<code>set_indent(value)</code>
<i>Getter</i>	<code>get_indent()</code>

- int `jst_flags`

<i>Default</i>	1
<i>Setter</i>	set_jst_flags(value)
<i>Getter</i>	get_jst_flags()

- float **line__spacing**

<i>Default</i>	1.0
<i>Setter</i>	set_line_spacing(value)
<i>Getter</i>	get_line_spacing()

- *TLShapedAttributedString* **string**

<i>Setter</i>	set_string(value)
<i>Getter</i>	get_string()

- float **width**

<i>Default</i>	-1.0
<i>Setter</i>	set_width(value)
<i>Getter</i>	get_width()

4.13.7 Method Descriptions

- void **copy_properties** (*TLShapedParagraph* source)
-

- *TLShapedAttributedString* **get__line** (int index) const
-

- Array **get__line__bounds** () const
-

- int **get__lines** () const
-

- Vector2 **get__size** () const
-

- Array **get__word__bounds** () const
-

4.14 TLShapedString

Inherits:

Inherited By: *TLShapedAttributedString*

Category: Core

4.14.1 Brief Description

Holds shaped line of plain text.

4.14.2 Properties

int	<i>base_direction</i>	3
<i>TLFontFamily</i>	<i>base_font</i>	
int	<i>base_font_size</i>	12
String	<i>base_font_style</i>	“Regular”
String	<i>features</i>	“”
String	<i>language</i>	“en”
bool	<i>preserve_control</i>	false
String	<i>text</i>	“”

4.14.3 Methods

void	<i>add_sstring</i> (<i>TLShapedString</i> text)
void	<i>add_text</i> (String text)
void	<i>add_utf16</i> (PoolByteArray text)
void	<i>add_utf32</i> (PoolByteArray text)
void	<i>add_utf8</i> (PoolByteArray text)
Array	<i>break_jst</i> () const
Array	<i>break_lines</i> (float width, int flags) const
Array	<i>break_words</i> () const
int	<i>char_count</i> () const
int	<i>clusters</i> () const
void	<i>copy_properties</i> (<i>TLShapedString</i> source)
void	<i>draw</i> (RID canvas_item, Vector2 position, Color modulate)
void	<i>draw_as_hex</i> (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
Vector2	<i>draw_cluster</i> (RID canvas_item, Vector2 position, int index, Color modulate)
void	<i>draw_dbg</i> (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
void	<i>draw_logical_as_hex</i> (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
bool	<i>empty</i> () const
float	<i>extend_to_width</i> (float width, int flags)
float	<i>get_ascent</i> () const
<i>TextDirection</i>	<i>get_char_direction</i> (int position) const
float	<i>get_cluster_ascent</i> (int index) const
String	<i>get_cluster_debug_info</i> (int index) const
float	<i>get_cluster_descent</i> (int index) const
int	<i>get_cluster_end</i> (int index) const
<i>TLFontFace</i>	<i>get_cluster_face</i> (int position) const
float	<i>get_cluster_face_size</i> (int position) const
int	<i>get_cluster_glyph</i> (int index, int glyph) const
Vector2	<i>get_cluster_glyph_advance</i> (int index, int glyph) const
Vector2	<i>get_cluster_glyph_offset</i> (int index, int glyph) const
int	<i>get_cluster_glyphs</i> (int index) const
float	<i>get_cluster_height</i> (int index) const
int	<i>get_cluster_index</i> (int position) const
float	<i>get_cluster_leading_edge</i> (int index) const
Rect2	<i>get_cluster_rect</i> (int index) const

Continued on next page

Table 70 – continued from previous page

int	<i>get_cluster_start</i> (int index) const
float	<i>get_cluster_trailing_edge</i> (int index) const
float	<i>get_cluster_width</i> (int index) const
Array	<i>get_cursor_positions</i> (int position, int primary_dir) const
float	<i>get_descent</i> () const
float	<i>get_height</i> () const
Array	<i>get_highlight_shapes</i> (int start, int end) const
int	<i>get_para_direction</i> () const
PoolByteArray	<i>get_utf16</i> () const
PoolByteArray	<i>get_utf32</i> () const
PoolByteArray	<i>get_utf8</i> () const
float	<i>get_width</i> () const
int	<i>hit_test</i> (float position) const
int	<i>hit_test_cluster</i> (float position) const
bool	<i>is_valid</i> () const
int	<i>length</i> () const
int	<i>next_safe_bound</i> (int position) const
int	<i>pos_u16_to_wcs</i> (int position) const
int	<i>pos_wcs_to_u16</i> (int position) const
int	<i>prev_safe_bound</i> (int position) const
void	<i>replace_sstring</i> (int start, int end, <i>TLShapedString</i> text)
void	<i>replace_text</i> (int start, int end, String text)
void	<i>replace_utf16</i> (int start, int end, PoolByteArray text)
void	<i>replace_utf32</i> (int start, int end, PoolByteArray text)
void	<i>replace_utf8</i> (int start, int end, PoolByteArray text)
void	<i>set_utf16</i> (PoolByteArray data)
void	<i>set_utf32</i> (PoolByteArray data)
void	<i>set_utf8</i> (PoolByteArray data)
bool	<i>shape</i> ()
<i>TLShapedString</i>	<i>substr</i> (int start, int end, int trim) const

4.14.4 Signals

- `string_changed ()`
-

- `string_shaped ()`

4.14.5 Enumerations

enum `TextDirection`:

- `TEXT_DIRECTION_LTR = 0` — Left-to-right text writing direction
 - `TEXT_DIRECTION_RTL = 1` — Right-to-left text writing direction
 - `TEXT_DIRECTION_LOCALE = 2` — Text writing direction is derived from the locale's script according to the CLDR metadata
 - `TEXT_DIRECTION_AUTO = 3` — Text writing direction is derived from the first character in the string with BiDi class L, R, or AL or locale's script if text is not strongly directional
 - `TEXT_DIRECTION_INVALID = 4`
-

enum `TextJustification`:

- `TEXT_JUSTIFICATION_NONE = 0` — No text justification

- **TEXT_JUSTIFICATION_KASHIDA_AND_WHITESPACE = 1** — Use kashida and whitespace elongation to justify text
- **TEXT_JUSTIFICATION_KASHIDA_ONLY = 2** — Use kashida elongation to justify text
- **TEXT_JUSTIFICATION_WHITESPACE_ONLY = 3** — Use whitespace elongation to justify text
- **TEXT_JUSTIFICATION_KASHIDA_AND_WHITESPACE_AND_INTERCHAR = 4**
- **TEXT_JUSTIFICATION_KASHIDA_AND_INTERCHAR = 5**
- **TEXT_JUSTIFICATION_WHITESPACE_AND_INTERCHAR = 6**
- **TEXT_JUSTIFICATION_INTERCHAR_ONLY = 7**

enum **TextBreak**:

- **TEXT_BREAK_NONE = 0** — No line breaking
- **TEXT_BREAK_MANDATORY = 1** — Break lines only at mandatory break points
- **TEXT_BREAK_MANDATORY_AND_WORD_BOUND = 2** — Break lines at mandatory break points and word boundaries
- **TEXT_BREAK_MANDATORY_AND_ANYWHERE = 3** — Break lines at mandatory break points and grapheme cluster boundaries

enum **TextTrimMode**:

- **TEXT_TRIM_NONE = 0** — No substring trimming
- **TEXT_TRIM_BREAK = 1** — Trim line break characters for substring ends
- **TEXT_TRIM_BREAK_AND_WHITESPACE = 2** — Trim line break and whitespace characters for substring ends

4.14.6 Description

Note 1: Code points, Characters, Clusters and Glyphs

- A code point is a single encoding UTF-16 unit (Unicode character or half of the surrogate pair).
- A character is a full Unicode character.
- A grapheme cluster is the abstract unit of a writing system (a letter, a digit, or punctuation).
- A glyph is a shape used to render a character or a sequence of characters.

In general, code point, characters, clusters and glyphs do not have one-to-one correspondence.

Note 2: Encoding

TLShapedString uses UTF-16 encoding, all positions accepted and returned by TLShapedString function are measured in UTF-16 code points.

4.14.7 Property Descriptions

- int **base_direction**

<i>Default</i>	3
<i>Setter</i>	set_base_direction(value)
<i>Getter</i>	get_base_direction()

Base text writing direction. Default: `TEXT_DIRECTION_AUTO`

- *TLFontFamily* **base_font**

<i>Setter</i>	<code>set_base_font(value)</code>
<i>Getter</i>	<code>get_base_font()</code>

Base font family reference. Default: `null`

- **int base_font_size**

<i>Default</i>	<code>12</code>
<i>Setter</i>	<code>set_base_font_size(value)</code>
<i>Getter</i>	<code>get_base_font_size()</code>

Font size. Default: `12`

- **String base_font_style**

<i>Default</i>	<code>"Regular"</code>
<i>Setter</i>	<code>set_base_font_style(value)</code>
<i>Getter</i>	<code>get_base_font_style()</code>

Style name (Regular, Bold, Italic, Oblique etc.). Default: `"Regular"`

- **String features**

<i>Default</i>	<code>"</code>
<i>Setter</i>	<code>set_features(value)</code>
<i>Getter</i>	<code>get_features()</code>

Comma separated list of OpenType feature tags. More info: <https://docs.microsoft.com/en-us/typography/opentype/spec/featuretags>. Default: `"`

- **String language**

<i>Default</i>	<code>"en"</code>
<i>Setter</i>	<code>set_language(value)</code>
<i>Getter</i>	<code>get_language()</code>

Language code. Default: `"`

- **bool preserve_control**

<i>Default</i>	<code>false</code>
<i>Setter</i>	<code>set_preserve_control(value)</code>
<i>Getter</i>	<code>get_preserve_control()</code>

If **true** displays control character. Default: **false**

- String **text**

<i>Default</i>	“”
<i>Setter</i>	set_text(value)
<i>Getter</i>	get_text()

Text string. Default: ""

4.14.8 Method Descriptions

- void **add_sstring** (*TLShapedString* text)

- void **add_text** (String text)

Appends plain text string.

- void **add_utf16** (PoolByteArray text)

- void **add_utf32** (PoolByteArray text)

- void **add_utf8** (PoolByteArray text)

- Array **break_jst** () const

- Array **break_lines** (float width, int flags) const

Breaks text into lines that fit within a specified width.

Returns Array of line boundaries.

- Array **break_words** () const

Breaks text into words.

Returns Array of word boundaries.

- int **char_count** () const

Returns number of characters in the string.

- int **clusters** () const

Returns number of grapheme clusters, clusters are indexed in visual order.

- void **copy_properties** (*TLShapedString* source)

-
- void **draw** (RID canvas_item, Vector2 position, Color modulate)

Draws a string.

- void **draw__as__hex** (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
-

- Vector2 **draw__cluster** (RID canvas_item, Vector2 position, int index, Color modulate)

Draws single grapheme cluster. Returns advance.

- void **draw__dbg** (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
-

- void **draw__logical__as__hex** (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)
-

- bool **empty** () const

Returns **true** if the string is empty.

- float **extend__to__width** (float width, int flags)

Increase text width to the specified. Returns new line width.

- float **get__ascent** () const

Returns ascent of the line.

- *TextDirection* **get__char__direction** (int position) const

Return writing direction of a character writing direction.

- float **get__cluster__ascent** (int index) const

Returns cluster ascent.

- String **get__cluster__debug__info** (int index) const
-

- float **get__cluster__descent** (int index) const

Returns cluster descent.

- int **get__cluster__end** (int index) const
-

Returns last character position corresponding cluster.

- *TLFontFace* **get__cluster__face** (int position) const
-

- float **get__cluster__face__size** (int position) const
-

- int **get__cluster__glyph** (int index, int glyph) const

Returns glyph ID.

- Vector2 **get__cluster__glyph__advance** (int index, int glyph) const

Returns glyph advance.

- Vector2 **get__cluster__glyph__offset** (int index, int glyph) const

Returns glyph offset.

- int **get__cluster__glyphs** (int index) const

Returns number of glyphs in cluster.

- float **get__cluster__height** (int index) const

Returns cluster height.

- int **get__cluster__index** (int position) const

Returns cluster index corresponding to a specific character position in string.

- float **get__cluster__leading__edge** (int index) const

Returns cluster leading edge offset in pixels.

- Rect2 **get__cluster__rect** (int index) const

Returns cluster bounding rectangle.

- int **get__cluster__start** (int index) const

Returns first character position corresponding cluster.

- float **get__cluster__trailing__edge** (int index) const

Returns cluster trailing edge offset in pixels.

- float **get__cluster__width** (int index) const
-

Returns cluster width.

- Array **get__cursor__positions** (int position, int primary_dir) const

Returns an Array of float (up to two elements) offsets corresponding to the strong and weak cursor, at the specified character position.

- float **get__descent** () const

Returns descent of the line.

- float **get__height** () const

Returns height of the line.

- Array **get__highlight__shapes** (int start, int end) const

Returns an Array of Rect2 enclosing the selection/highlight in the specified range.

- int **get__para__direction** () const
-

- PoolByteArray **get__utf16** () const

Returns raw text string in UTF-16 encoding.

- PoolByteArray **get__utf32** () const

Returns raw text string in UTF-32 encoding.

- PoolByteArray **get__utf8** () const

Returns raw text string in UTF-8 encoding.

- float **get__width** () const

Returns width of the line.

- int **hit__test** (float position) const

Returns a cursor position corresponding to the specified pixel offset.

- int **hit__test__cluster** (float position) const
-

- bool **is__valid** () const

Returns **true** if the string is shaped successfully.

- int **length** () const
-

Returns number of UTF-16 codepoints in the string.

- `int next_safe_bound (int position) const`

Returns next whole character position in the string.

- `int pos_u16_to_wcs (int position) const`

Returns character position (Characters)

- `int pos_wcs_to_u16 (int position) const`

Retruns character position (UTF-16 codepoints)

- `int prev_safe_bound (int position) const`

Returns previous whole character position in the string.

- `void replace_sstring (int start, int end, TLShapedString text)`
-

- `void replace_text (int start, int end, String text)`

Replaces substring.

- `void replace_utf16 (int start, int end, PoolByteArray text)`
-

- `void replace_utf32 (int start, int end, PoolByteArray text)`
-

- `void replace_utf8 (int start, int end, PoolByteArray text)`
-

- `void set_utf16 (PoolByteArray data)`

Sets taw text string in UTF-16 encoding

- `void set_utf32 (PoolByteArray data)`

Sets taw text string in UTF-32 encoding

- `void set_utf8 (PoolByteArray data)`

Sets taw text string in UTF-8 encoding

- `bool shape ()`

Shapes string and returns `true` if the string is shaped successfully.

- `TLShapedString substr (int start, int end, int trim) const`
-