libgdtl Documentation

Release latest

Oct 31, 2019

General

CHAPTER 1

libgdtl Introduction

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
builtin_runtime	Use the built-in libraries	true
use_graphite2	Enable SIL Graphite 2 complementary shaper	true

1.2.2 Building *libdgtl* module

Clone this repository (without *–recursive* flag) into Godot's *modules* subfolder as *godot_tl*. Rebuild Godot engine as ususal.

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 *.a* or *.lib* 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 *libdgtl* module

You can compile this module by executing:

```
meson {Targer-Folder} -Dgodot-cpp-lib-name={Godot-CPP-Name} --buildtype=release
ninja -C {Targer-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/JulietaUla/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).

CHAPTER 2

libgdtl API

2.1 TLBitmapFontFace

Inherits: *TLFontFace*

Category: Core

2.1.1 Brief Description

An AngelCode Bitmap Font Generator bitmap font for drawing text.

2.1.2 Properties

int	texture_flags	**O:** 2048

2.1.3 Description

TLBitmapFontFace have limited shaping support.

TLBitmapFontFace doesn't support OpenType features.

2.2 TLDynamicFontFace

Inherits: *TLFontFace*

Category: Core

2.2.1 Brief Description

A TrueType, OpenType or Graphite font for drawing text.

2.2.2 Properties

bool	*force_autohinter*	false
int	*hinting*	2
float	*oversampling*	1.0
int	texture_flags	**O:** 2048

2.2.3 Methods

bool	*has_graphite* ( ) const

2.2.4 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)

2.2.5 Property Descriptions

- bool **force_autohinter**

Default	false
Setter	set_force_autohinter(value)
Getter	get_force_autohinter()

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

- int **hinting**

Default	2
Setter	set_hinting(value)
Getter	get_hinting()

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

- float **oversampling**

Default	1.0
Setter	set_oversampling(value)
Getter	get_oversampling()

Font oversampling factor. Default: `1.0`

2.2.6 Method Descriptions

- bool **has_graphite** ( ) const

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

2.3 TLFontFace

Inherits:

Inherited By: *TLBitmapFontFace*, *TLDynamicFontFace*

Category: Core

2.3.1 Brief Description

Virtual class

A base font face class.

2.3.2 Properties

String	*font_path*	“”
int	*texture_flags*	0

2.3.3 Methods

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

2.3.4 Property Descriptions

- String **font_path**

Default	“”
Setter	set_font_path(value)
Getter	get_font_path()

- int **texture_flags**

Default	0
Setter	set_texture_flags(value)
Getter	get_texture_flags()

Font texture flags. Default: `FLAG_VIDEO_SURFACE`

2.3.5 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

2.4 TLFontFamily

Inherits:

Category: Core

2.4.1 Brief Description

A set of fonts that make up a font family.

2.4.2 Methods

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)

2.4.3 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)

2.5 TLFontIterator

Inherits:

Category: Core

2.5.1 Brief Description

2.6 TLICUDataLoader

Inherits:

Category: Core

2.6.1 Brief Description

Helper class that handles ICU data loading.

2.6.2 Properties

String	*data_path*	“”

2.6.3 Methods

bool	*load* (String resource_path)

2.6.4 Property Descriptions

- String **data_path**

Default	“”
Setter	set_data_path(value)
Getter	get_data_path()

2.6.5 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.

2.7 TLLabel

Inherits:

Category: Core

2.7.1 Brief Description

2.7.2 Properties

int	*align*	0
bool	*autowrap*	false
TLFontFamily	*base_font*	
int	*base_font_size*	12
String	*base_font_style*	"Regular"
bool	*clip_text*	false
String	*language*	""
Control.MouseFilter	mouse_filter	**O:** 2
String	*ot_features*	""
int	size_flags_vertical	**O:** 4
String	*text*	""
int	*text_direction*	3
bool	*uppercase*	false
int	*valign*	0

2.7.3 Methods

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)

2.7.4 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**

2.7.5 Property Descriptions

- int **align**

Default	0
Setter	set_align(value)
Getter	get_align()

- bool **autowrap**

Default	false
Setter	set_autowrap(value)
Getter	has_autowrap()

- *TLFontFamily* **base_font**

Setter	set_base_font(value)
Getter	get_base_font()

- int **base_font_size**

Default	12
Setter	set_base_font_size(value)
Getter	get_base_font_size()

- String **base_font_style**

Default	“Regular”
Setter	set_base_font_style(value)
Getter	get_base_font_style()

- bool **clip_text**

Default	false
Setter	set_clip_text(value)
Getter	is_clipping_text()

- String **language**

Default	“”
Setter	set_language(value)
Getter	get_language()

- String **ot_features**

Default	“”
Setter	set_ot_features(value)
Getter	get_ot_features()

- String **text**

Default	“”
Setter	set_text(value)
Getter	get_text()

- int **text_direction**

Default	3
Setter	set_text_direction(value)
Getter	get_text_direction()

- bool **uppercase**

Default	false
Setter	set_uppercase(value)
Getter	is_uppercase()

- int **valign**

Default	0
Setter	set_valign(value)
Getter	get_valign()

2.7.6 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)

2.8 TLLineEdit

Inherits:

Category: Core

2.8.1 Brief Description

2.8.2 Properties

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

2.8.3 Methods

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* ( )

2.8.4 Signals

- **text_changed** (String new_text)

- **text_entered** (String new_text)

2.8.5 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**

2.8.6 Property Descriptions

- int **align**

Default	0
Setter	set_align(value)
Getter	get_align()

- *TLFontFamily* **base_font**

Setter	set_base_font(value)
Getter	get_base_font()

- int **base_font_size**

Default	12
Setter	set_base_font_size(value)
Getter	get_base_font_size()

- String **base_font_style**

Default	“Regular”
Setter	set_base_font_style(value)
Getter	get_base_font_style()

- bool **caret_blink**

Default	false
Setter	cursor_set_blink_enabled(value)
Getter	cursor_get_blink_enabled()

- float **caret_blink_speed**

Default	0.65
Setter	cursor_set_blink_speed(value)
Getter	cursor_get_blink_speed()

- int **caret_position**

Default	0
Setter	set_cursor_position(value)
Getter	get_cursor_position()

- bool **clear_button_enabled**

Default	false
Setter	set_clear_button_enabled(value)
Getter	is_clear_button_enabled()

- bool **context_menu_enabled**

Default	true
Setter	set_context_menu_enabled(value)
Getter	is_context_menu_enabled()

- bool **editable**

Default	true
Setter	set_editable(value)
Getter	is_editable()

- bool **expand_to_text_length**

Default	false
Setter	set_expand_to_text_length(value)
Getter	get_expand_to_text_length()

- String **language**

Default	“”
Setter	set_language(value)
Getter	get_language()

- int **max_length**

Default	0
Setter	set_max_length(value)
Getter	get_max_length()

- String **ot_features**

Default	“”
Setter	set_ot_features(value)
Getter	get_ot_features()

- float **placeholder_alpha**

Default	0.6
Setter	set_placeholder_alpha(value)
Getter	get_placeholder_alpha()

- String **placeholder_text**

Default	“”
Setter	set_placeholder(value)
Getter	get_placeholder()

- bool **secret**

Default	false
Setter	set_secret(value)
Getter	is_secret()

- String **secret_character**

Default	“*”
Setter	set_secret_character(value)
Getter	get_secret_character()

- String **text**

Default	“”
Setter	set_text(value)
Getter	get_text()

- int **text_direction**

Default	3
Setter	set_text_direction(value)
Getter	get_text_direction()

2.8.7 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** ( )

2.9 TLProtoControl

Inherits:

Category: Core

2.9.1 Brief Description

Rich text input control prototype.

2.9.2 Properties

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

2.9.3 Methods

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)

2.9.4 Signals

- **cursor_changed** ( )

- **paragraph_changed** ( )

2.9.5 Property Descriptions

- Color **back_color**

Default	Color(1, 1, 1, 0)
Setter	set_back_color(value)
Getter	get_back_color()

- float **paragraph_spacing**

Default	3.0
Setter	set_paragraph_spacing(value)
Getter	get_paragraph_spacing()

- bool **readonly**

Default	false
Setter	set_readonly(value)
Getter	get_readonly()

- bool **selectable**

Default	true
Setter	set_selectable(value)
Getter	get_selectable()

2.9.6 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)

2.10 TLProtoControlSelection

Inherits:

Category: Core

2.10.1 Brief Description

2.10.2 Properties

int	*caret_offset*	0
int	*caret_para*	0
int	*end_offset*	0
int	*end_para*	0
int	*start_offset*	0
int	*start_para*	0

2.10.3 Signals

- **selection_changed** ( )

2.10.4 Property Descriptions

- int **caret_offset**

Default	0
Setter	set_caret_offset(value)
Getter	get_caret_offset()

- int **caret_para**

Default	0
Setter	set_caret_para(value)
Getter	get_caret_para()

- int **end_offset**

Default	0
Setter	set_end_offset(value)
Getter	get_end_offset()

- int **end_para**

Default	0
Setter	set_end_para(value)
Getter	get_end_para()

- int **start_offset**

Default	0
Setter	set_start_offset(value)
Getter	get_start_offset()

- int **start_para**

Default	0
Setter	set_start_para(value)
Getter	get_start_para()

2.11 TLShapedAttributedString

Inherits: *TLShapedString*

Category: Core

2.11.1 Brief Description

Golds shaped line of text with associated attributes.

2.11.2 Methods

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

2.11.3 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 inline 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 widht. 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 moultiple 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

2.11.4 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.

2.12 TLShapedParagraph

Inherits:

Category: Core

2.12.1 Brief Description

2.12.2 Properties

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

2.12.3 Methods

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

2.12.4 Signals

- **paragraph_changed** ( )

2.12.5 Enumerations

enum **ParaHAlign**:

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

2.12.6 Property Descriptions

- Color **back_color**

Default	Color(1, 1, 1, 0)
Setter	set_back_color(value)
Getter	get_back_color()

- int **brk_flags**

Default	2
Setter	set_brk_flags(value)
Getter	get_brk_flags()

- int **halign**

Default	0
Setter	set_halign(value)
Getter	get_halign()

- float **indent**

Default	0.0
Setter	set_indent(value)
Getter	get_indent()

- int **jst_flags**

Default	1
Setter	set_jst_flags(value)
Getter	get_jst_flags()

- float **line_spacing**

Default	1.0
Setter	set_line_spacing(value)
Getter	get_line_spacing()

- *TLShapedAttributedString* **string**

Setter	set_string(value)
Getter	get_string()

- float **width**

Default	-1.0
Setter	set_width(value)
Getter	get_width()

2.12.7 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

2.13 TLShapedString

Inherits:

Inherited By: *TLShapedAttributedString*

Category: Core

2.13.1 Brief Description

Holds shaped line of plain text.

2.13.2 Properties

int	*base_direction*	3
TLFontFamily	*base_font*	
int	*base_font_size*	12
String	*base_font_style*	"Regular"
String	*features*	""
String	*language*	"en"
bool	*preserve_control*	false
String	*text*	""

2.13.3 Methods

void	*add_sstring* (*TLShapedString* text)
void	*add_text* (String text)
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
Array	*break_words* ( ) const
int	*char_count* ( ) const
int	*clusters* ( ) const
void	*copy_properties* (*TLShapedString* source)
void	*draw* (RID canvas_item, Vector2 position, Color modulate)
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)
void	*draw_dbg* (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_ops)

Continued on next pa

Table 1 – continued from previous page

void	*draw_logical_as_hex* (RID canvas_item, Vector2 position, Color modulate, bool draw_brk_ops, bool draw_jst_op
bool	*empty* ( ) const
float	*extend_to_width* (float width, int flags)
float	*get_ascent* ( ) const
TextDirection	*get_char_direction* (int position) const
float	*get_cluster_ascent* (int index) const
String	*get_cluster_debug_info* (int index) const
float	*get_cluster_descent* (int index) const
int	*get_cluster_end* (int index) const
TLFontFace	*get_cluster_face* (int position) const
float	*get_cluster_face_size* (int position) const
int	*get_cluster_glyph* (int index, int glyph) const
Vector2	*get_cluster_glyph_advance* (int index, int glyph) const
Vector2	*get_cluster_glyph_offset* (int index, int glyph) const
int	*get_cluster_glyphs* (int index) const
float	*get_cluster_height* (int index) const
int	*get_cluster_index* (int position) const
float	*get_cluster_leading_edge* (int index) const
Rect2	*get_cluster_rect* (int index) const
int	*get_cluster_start* (int index) const
float	*get_cluster_trailing_edge* (int index) const
float	*get_cluster_width* (int index) const
Array	*get_cursor_positions* (int position, int primary_dir) const
float	*get_descent* ( ) const
float	*get_height* ( ) const
Array	*get_highlight_shapes* (int start, int end) const
int	*get_para_direction* ( ) const
PoolByteArray	*get_utf16* ( ) const
PoolByteArray	*get_utf32* ( ) const
PoolByteArray	*get_utf8* ( ) const
float	*get_width* ( ) const
int	*hit_test* (float position) const
int	*hit_test_cluster* (float position) const
bool	*is_valid* ( ) const
int	*length* ( ) const
int	*next_safe_bound* (int position) const
int	*pos_u16_to_wcs* (int position) const
int	*pos_wcs_to_u16* (int position) const
int	*prev_safe_bound* (int position) const
void	*replace_sstring* (int start, int end, *TLShapedString* text)
void	*replace_text* (int start, int end, String text)
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)
void	*set_utf32* (PoolByteArray data)
void	*set_utf8* (PoolByteArray data)
bool	*shape* ( )
TLShapedString	*substr* (int start, int end, int trim) const

2.13.4 Signals

- **string_changed ( )**

- **string_shaped ( )**

2.13.5 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

2.13.6 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 charecter.
- 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.

2.13.7 Property Descriptions

- int **base_direction**

Default	3
Setter	set_base_direction(value)
Getter	get_base_direction()

Base text writing direction. Default: `TEXT_DIRECTION_AUTO`

- *TLFontFamily* **base_font**

Setter	set_base_font(value)
Getter	get_base_font()

Base font family reference. Default: `null`

- int **base_font_size**

Default	12
Setter	set_base_font_size(value)
Getter	get_base_font_size()

Font size. Default: `12`

- String **base_font_style**

Default	“Regular”
Setter	set_base_font_style(value)
Getter	get_base_font_style()

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

- String **features**

Default	“”
Setter	set_features(value)
Getter	get_features()

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

- String **language**

Default	“en”
Setter	set_language(value)
Getter	get_language()

Language code. Default: `""`

- bool **preserve_control**

Default	false
Setter	set_preserve_control(value)
Getter	get_preserve_control()

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

- String **text**

Default	“”
Setter	set_text(value)
Getter	get_text()

Text string. Default: `""`

2.13.8 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.

Retunrs Array of line boundaries.

- Array **break_words** ( ) const

Breaks text into words.

Retunrs 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 successfuly.

- 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 successfuly.

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