

microvg

User Manual



MICROEJ[®]

Reference:	TLT-XXX-MAN-microvg-microvg
Version:	6.0.1
Revision:	XXX

Confidentiality & Intellectual Property

All rights reserved. Information, technical data and tutorials contained in this document are confidential and proprietary under copyright Law of Industrial Smart Software Technology (IS2T S.A.) operating under the brand name MicroEJ®. Without written permission from IS2T S.A., *copying or sending parts of the document or the entire document by any means to third parties is not permitted*. Granted authorizations for using parts of the document or the entire document do not mean IS2T S.A. gives public full access rights.

The information contained herein is not warranted to be error-free. IS2T® and MicroEJ® and all relative logos are trademarks or registered trademarks of IS2T S.A. in France and other Countries.

Java™ is Sun Microsystems' trademark for a technology for developing application software and deploying it in cross-platform, networked environments. When it is used in this documentation without adding the ™ symbol, it includes implementations of the technology by companies other than Sun.

Java™, all Java-based marks and all related logos are trademarks or registered trademarks of Sun Microsystems Inc, in the United States and other Countries.

Other trademarks are proprietary of their authors.

Table of Contents

1. Data Structure Documentation	1
1.1. MICROVG_PATH_HEADER Struct Reference	1
1.1.1. Detailed Description	1
2. File Documentation	2
2.1. bsp/vg/inc/vg_configuration.h File Reference	2
2.1.1. Detailed Description	2
2.1.2. Macro Definition Documentation	2
2.2. bsp/vg/inc/vg_freetype.h File Reference	3
2.2.1. Detailed Description	3
2.3. bsp/vg/inc/vg_helper.h File Reference	3
2.3.1. Detailed Description	4
2.3.2. Macro Definition Documentation	5
2.3.3. Function Documentation	5
2.4. bsp/vg/inc/vg_path.h File Reference	6
2.4.1. Detailed Description	7
2.5. bsp/vg/src/LLVG_BVI_impl_stub.c File Reference	7
2.5.1. Detailed Description	7
2.6. bsp/vg/src/LLVG_FONT_impl_freetype.c File Reference	8
2.6.1. Detailed Description	8
2.7. bsp/vg/src/LLVG_FONT_impl_stub.c File Reference	8
2.7.1. Detailed Description	8
2.8. bsp/vg/src/LLVG_GRADIENT_impl_stub.c File Reference	9
2.8.1. Detailed Description	9
2.9. bsp/vg/src/LLVG_MATRIX_impl.c File Reference	9
2.9.1. Detailed Description	10
2.10. bsp/vg/src/LLVG_PAINTER_impl.c File Reference	10
2.10.1. Detailed Description	11
2.11. bsp/vg/src/LLVG_PATH_impl_dual.c File Reference	11
2.11.1. Detailed Description	11
2.12. bsp/vg/src/LLVG_PATH_impl_single.c File Reference	11
2.12.1. Detailed Description	12
2.13. bsp/vg/src/LLVG_PATH_impl_stub.c File Reference	12
2.13.1. Detailed Description	12
2.14. bsp/vg/src/vg_freetype_path.c File Reference	13
2.14.1. Detailed Description	13
2.15. bsp/vg/src/vg_helper.c File Reference	13
2.15.1. Detailed Description	14
2.15.2. Function Documentation	14
Index	16

List of Tables

2.1.	5
2.2.	14

Chapter 1. Data Structure Documentation

1.1. MICROVG_PATH_HEADER Struct Reference

Data Fields

- uint32_t
- uint8_t
- uint8_t
- uint8_t
- uint8_t
- float
- float
- float
- float

1.1.1. Detailed Description

Definition at line 54 of file `vg_path.h`.

The documentation for this struct was generated from the following file:
`bsp/vg/inc/vg_path.h`

Chapter 2. File Documentation

2.1. bsp/vg/inc/vg_configuration.h File Reference

MicroEJ MicroVG library low level API: enable some features according to the hardware capacities.

Macros

- `#define MICROVG_CONFIGURATION_VERSION (3)`
Compatibility sanity check value. This define value is checked in the implementation to validate that the version of this configuration is compatible with the implementation.
- `#define (1)`
- `#define (2)`
- `#define (1)`
- `#define (2)`
- `#define (1)`
- `#define (2)`
- `#define VG_FEATURE_PATH_SINGLE_ARRAY`
- `#define VG_FEATURE_GRADIENT_FULL`
- `#define VG_FEATURE_FONT_FREETYPE_VECTOR`
- `#define`
- `#define`
- `#define`
- `#define`
- `#define`
- `#define (80 * 1024)`
- `#define (80 * 1024)`

2.1.1. Detailed Description

MicroEJ MicroVG library low level API: enable some features according to the hardware capacities.

Author. MicroEJ Developer Team

2.1.2. Macro Definition Documentation

2.1.2.1. MICROVG_CONFIGURATION_VERSION

```
#define MICROVG_CONFIGURATION_VERSION (3)
```

Compatibility sanity check value. This define value is checked in the implementation to validate that the version of this configuration is compatible with the implementation.

This value must not be changed by the user of the CCO. This value must be incremented by the implementor of the CCO when a configuration define is added, deleted or modified.

Definition at line 34 of file `vg_configuration.h`.

2.2. bsp/vg/inc/vg_freetype.h File Reference

MicroEJ MicroVG library low level API: implementation over FreeType.

```
#include "vg_configuration.h"
#include "vg_path.h"
```

Typedefs

- `typedef jint(*) (VG_PATH_HEADER_t *path, jfloat *matrix, uint32_t color, bool fill_rule_even_odd, void *user_data)`

Functions

- `void (void)`
- `jint (VG_FREETYPE_draw_glyph_t drawer, const jchar *text, jint face_handle, jfloat size, const jfloat *matrix, uint32_t color, jfloat letter_spacing, jfloat radius, jint direction, void *user_data)`
- `jint (jint color)`

2.2.1. Detailed Description

MicroEJ MicroVG library low level API: implementation over FreeType.

Author. MicroEJ Developer Team

Version. 6.0.1

2.3. bsp/vg/inc/vg_helper.h File Reference

MicroEJ MicroVG library low level API: helper to implement library natives methods.

```
#include <stdio.h>
#include <sni.h>
#include "mej_log.h"
```

Macros

- `#define (fmt, ...)`
- `#define (fmt, ...) MEJ_LOG(ERROR, MICROVG, fmt, ## __VA_ARGS__)`
- `#define VG_HELPER_NULL_GRADIENT 0`

Set this define to monitor freetype heap evolution. It needs `MEJ_LOG_MICROVG` and `MEJ_LOG_INFO_LEVEL` defines to print the heap logs.

- `#define FT_FACE_FLAG_COMPLEX_LAYOUT (((uint32_t)1) << 31)`

Freetype supplementary flag for complex layout Uses a free bit in freetype face flags to convey the complex layout mode information with the freetype face. freetype.h must be checked on freetype update to ensure that this bit is still free.

- `#define 3.1415926535`
- `#define (r) ((r) * (180.0f / M_PI))`
- `#define (d) (((d) * M_PI) / 180.0f)`
- `#define (f) (*(uint32_t *)&(f))`
- `#define (i) (*(float *)&(i))`

Functions

- `void (void)`
- `int VG_HELPER_get_utf (const unsigned short *text, int length, int *offset)`

Gets the UTF character from a text buffer at the given offset and updates the offset to point to the next character.

- `void (int faceHandle, const unsigned short *text, int length)`
- `bool (int *glyph_idx, int *x_advance, int *y_advance, int *x_offset, int *y_offset)`
- `const jfloat * (const jfloat *matrix)`
- `uint32_t (uint32_t color, uint32_t alpha)`
- `void (jfloat *dest, jfloat x, jfloat y, const jfloat *matrix)`

2.3.1. Detailed Description

MicroEJ MicroVG library low level API: helper to implement library natives methods.

Author. MicroEJ Developer Team

Version. 6.0.1

2.3.2. Macro Definition Documentation

2.3.2.1. VG_HELPER_NULL_GRADIENT

```
#define VG_HELPER_NULL_GRADIENT 0
```

Set this define to monitor freetype heap evolution. It needs MEJ_LOG_MICROVG and MEJ_LOG_INFO_LEVEL defines to print the heap logs.

NULL Gradient value

Definition at line 57 of file vg_helper.h.

2.3.3. Function Documentation

2.3.3.1. VG_HELPER_get_utf()

```
int VG_HELPER_get_utf (const unsigned short * text, int length, int *  
offset)
```

Gets the UTF character from a text buffer at the given offset and updates the offset to point to the next character.

Some characters have some special values; they are made up of two Unicode characters in two specific ranges such that the first Unicode character is in one range (for example 0xD800-0xD8FF) and the second Unicode character is in the second range (for example 0xDC00-0xDCFF). This is called a surrogate pair.

If a surrogate pair is incomplete (missing second character), this function returns "0" (error) and does not update the offset.

Parameters .

Table 2.1.

in	text	text buffer encoded in UTF16 where to read UTF character.
in	length	length of the text buffer.
	[in/out]	offset: offset in the text buffer where to read UTF character. Updated to the next character position.

Returns. The decoded UTF character.

Definition at line 164 of file vg_helper.c.

```

164      {
165          highPart = GET_NEXT_CHARACTER(textCharRam, length, *offset);
166          ret = 0;
167
168          ((highPart >= MIN_HIGH_SURROGATE) && (highPart <=
169              MAX_HIGH_SURROGATE)) {
170              (*offset < (length - 1)) {
171                  lowPart = GET_NEXT_CHARACTER(textCharRam, length,
172                      *(offset) + 1);
173
174                  ((lowPart >= MIN_LOW_SURROGATE) && (lowPart <=
175                      MAX_LOW_SURROGATE)) {
176                      *offset += 2;
177
178                      ret = 0;
179                      ret += ((int)highPart - ()MIN_HIGH_SURROGATE);
180                      ret <= (int)10;
181                      ret += ((int)lowPart - ()MIN_LOW_SURROGATE);
182                      ret += (int)MIN_SUPPLEMENTARY_CODE_POINT;
183                  }
184              }
185          } {
186              *offset += 1;
187
188              ret = 0x0000FFFF & (int)highPart;
189          }
190
191          ret;
192      }

```

2.4. bsp/vg/inc/vg_path.h File Reference

MicroEJ MicroVG library low level API: implementation of Path.

```

#include "vg_configuration.h"
#include <sni.h>
#include <stddef.h>

```

Data Structures

- struct MICROVG_PATH_HEADER

Typedefs

- typedef struct MICROVG_PATH_HEADER

Functions

- uint8_t (void)
- uint32_t (jint command)
- void (void)
- uint32_t (void)
- uint32_t (jint command, uint32_t nbParams)
- uint32_t (jbyte *path, jint array_length, jint cmd)
- uint32_t (jbyte *path, jint array_length, jint cmd, jfloat x, jfloat y)
- uint32_t (jbyte *path, jint array_length, jint cmd, jfloat x1, jfloat y1, jfloat x2, jfloat y2)
- uint32_t (jbyte *path, jint array_length, jint cmd, jfloat x1, jfloat y1, jfloat x2, jfloat y2, jfloat x3, jfloat y3)
- uint32_t (jint command)

2.4.1. Detailed Description

MicroEJ MicroVG library low level API: implementation of Path.

Author. MicroEJ Developer Team

Version. 6.0.1

2.5. bsp/vg/src/LLVG_BVI_impl_stub.c File Reference

MicroVG library low level API: stubbed implementation of LLVG_BVI_impl.h.

```
#include "vg_configuration.h"
#include <LLVG_BVI_impl.h>
```

Functions

- void (MICROUI_Image *ui, MICROVG_Image *vg)
- void (MICROUI_GraphicsContext *gc)

2.5.1. Detailed Description

MicroVG library low level API: stubbed implementation of LLVG_BVI_impl.h.

Author. MicroEJ Developer Team

Version. 6.0.1

2.6. bsp/vg/src/LLVG_FONT_impl_freetype.c File Reference

MicroEJ MicroVG library low level API: implementation over FreeType.

```
#include "vg_configuration.h"
```

2.6.1. Detailed Description

MicroEJ MicroVG library low level API: implementation over FreeType.

Author. MicroEJ Developer Team

Version. 6.0.1

2.7. bsp/vg/src/LLVG_FONT_impl_stub.c File Reference

MicroEJ MicroVG library low level API: implementation over FreeType.

```
#include "vg_configuration.h"  
#include <LLVG_FONT_impl.h>  
#include <LLVG_PAINTER_impl.h>
```

Functions

- jint (jchar *font_name, jboolean complex_layout)
- jfloat (jchar *text, jint faceHandle, jfloat size, jfloat letterSpacing)
- jfloat (jchar *text, jint faceHandle, jfloat size)
- jfloat (jint faceHandle, jfloat size)
- jfloat (jint faceHandle, jfloat size)
- void (jint faceHandle)
- bool (void)

2.7.1. Detailed Description

MicroEJ MicroVG library low level API: implementation over FreeType.

Author. MicroEJ Developer Team

Version. 6.0.1

2.8. bsp/vg/src/LLVG_GRADIENT_impl_stub.c File Reference

MicroVG library low level API: stub of the linear gradient implementation.

```
#include "vg_configuration.h"
```

2.8.1. Detailed Description

MicroVG library low level API: stub of the linear gradient implementation.

This implementation is only compatible with the configuration VG_FEATURE_GRADIENT_FIRST_COLOR. It only stores the very first gradient's color as current color.

Author. MicroEJ Developer Team

Version. 6.0.1

2.9. bsp/vg/src/LLVG_MATRIX_impl.c File Reference

MicroEJ MicroVG library low level API: basic implementation of matrix APIs.

```
#include <math.h>
#include <string.h>
#include <LLVG_MATRIX_impl.h>
#include "vg_helper.h"
```

Functions

- void (jfloat *matrix)
- void (jfloat *dest, const jfloat *src)
- void (jfloat *dest, const jfloat *a, const jfloat *b)
- void (jfloat *matrix, jfloat x, jfloat y)
- void (jfloat *matrix, jfloat sx, jfloat sy)
- void (jfloat *matrix, jfloat degrees)
- void (jfloat *dest, const jfloat *a, const jfloat *b)
- void (jfloat *matrix, jfloat x, jfloat y)

- void (jfloat *matrix, jfloat scaleX, jfloat scaleY)
- void (jfloat *matrix, jfloat angleDegrees)
- void (jfloat *matrix, const jfloat *other)
- void (jfloat *matrix, jfloat dx, jfloat dy)
- void (jfloat *matrix, jfloat sx, jfloat sy)
- void (jfloat *matrix, jfloat degrees)
- void (jfloat *matrix, const jfloat *other)
- void (jfloat *x, jfloat *y, const jfloat *matrix)

2.9.1. Detailed Description

MicroEJ MicroVG library low level API: basic implementation of matrix APIs.

Author. MicroEJ Developer Team

Version. 6.0.1

2.10. bsp/vg/src/LLVG_PAINTER_impl.c File Reference

This file implements all MicroVG drawing native functions.

```
#include <LLVG_PAINTER_impl.h>
#include <LLUI_DISPLAY.h>
#include <LLVG_FONT_impl.h>
#include <LLVG_MATRIX_impl.h>
#include "vg_drawing.h"
#include "vg_helper.h"
#include "vg_trace.h"
```

Macros

- #define (fn) LOG_MICROVG_START(LOG_MICROVG_DRAWING_ID, CONCAT_DEFINES(LOG_MICROVG_DRAW_, fn))
- #define (fn) LOG_MICROVG_END(LOG_MICROVG_DRAWING_ID, CONCAT_DEFINES(LOG_MICROVG_DRAW_, fn))

Functions

- jint (MICROUI_GraphicsContext *gc, jbyte *pathData, jint x, jint y, jfloat *matrix, jint fillRule, jint blend, jint color)
- jint (MICROUI_GraphicsContext *gc, jbyte *pathData, jint x, jint y, jfloat *matrix, jint fillRule, jint alpha, jint blend, jint *gradientData, jfloat *gradientMatrix)

- jint (MICROUI_GraphicsContext *gc, jchar *text, jint faceHandle, jfloat size, jfloat x, jfloat y, jfloat *matrix, jint alpha, jint blend, jfloat letterSpacing)
- jint (MICROUI_GraphicsContext *gc, jchar *text, jint faceHandle, jfloat size, jfloat x, jfloat y, jfloat *matrix, jint alpha, jint blend, jfloat letterSpacing, jint *gradientData, jfloat *gradientMatrix)
- jint (MICROUI_GraphicsContext *gc, jchar *text, jint faceHandle, jfloat size, jint x, jint y, jfloat *matrix, jint alpha, jint blend, jfloat letterSpacing, jfloat radius, jint direction)
- jint (MICROUI_GraphicsContext *gc, jchar *text, jint faceHandle, jfloat size, jint x, jint y, jfloat *matrix, jint alpha, jint blend, jfloat letterSpacing, jfloat radius, jint direction, jint *gradientData, jfloat *gradientMatrix)
- jint (MICROUI_GraphicsContext *gc, MICROVG_Image *image, jint x, jint y, jfloat *matrix, jint alpha, jlong elapsed, const float color_matrix[])

2.10.1. Detailed Description

This file implements all MicroVG drawing native functions.

See also. LLVG_PAINTER_impl.h file comment

Author. MicroEJ Developer Team

Version. 6.0.1

2.11. bsp/vg/src/LLVG_PATH_impl_dual.c File Reference

MicroVG library low level API: implementation of path.

```
#include "vg_configuration.h"
```

2.11.1. Detailed Description

MicroVG library low level API: implementation of path.

This implementation uses a 32-bit "integer" value to store a path command and a 32-bit "float" value to store each command parameter.

The encoding can be overridden, see "[optional]: weak functions" in "vg_path.h"

Author. MicroEJ Developer Team

Version. 6.0.1

2.12. bsp/vg/src/LLVG_PATH_impl_single.c File Reference

MicroVG library low level API: implementation of path.

```
#include "vg_configuration.h"
```

2.12.1. Detailed Description

MicroVG library low level API: implementation of path.

This implementation uses a 32-bit "integer" value to store a path command and a 32-bit "float" value to store each command parameter.

The encoding can be overridden, see "[optional]: weak functions" in "vg_path.h"

Author. MicroEJ Developer Team

Version. 6.0.1

2.13. bsp/vg/src/LLVG_PATH_impl_stub.c File Reference

MicroVG library low level API: stubbed implementation of LLVG_PATH_impl.h and LLVG_PATH_PAINTER_impl.h.

```
#include "vg_configuration.h"
#include <LLVG_PATH_impl.h>
```

Functions

- jint (jbyte *jpath, jint length)
- jint (jbyte *jpath, jint length, jint cmd, jfloat x, jfloat y)
- jint (jbyte *jpath, jint length, jint cmd, jfloat x1, jfloat y1, jfloat x2, jfloat y2)
- jint (jbyte *jpath, jint length, jint cmd, jfloat x1, jfloat y1, jfloat x2, jfloat y2, jfloat x3, jfloat y3)
- void (jbyte *jpath)

2.13.1. Detailed Description

MicroVG library low level API: stubbed implementation of LLVG_PATH_impl.h and LLVG_PATH_PAINTER_impl.h.

Author. MicroEJ Developer Team

Version. 6.0.1

2.14. bsp/vg/src/vg_freetype_path.c File Reference

MicroEJ MicroVG library low level API: implementation over Freetype.

```
#include "vg_configuration.h"
```

2.14.1. Detailed Description

MicroEJ MicroVG library low level API: implementation over Freetype.

Author. MicroEJ Developer Team

Version. 6.0.1

2.15. bsp/vg/src/vg_helper.c File Reference

MicroEJ MicroVG library low level API: helper to implement library natives methods.

```
#include "vg_configuration.h"
#include <LLVG_MATRIX_impl.h>
#include "vg_helper.h"
#include "vg_trace.h"
#include "vg_drawing.h"
```

Macros

- #define ((unsigned short)0xD800)
- #define ((unsigned short)0xDBFF)
- #define ((unsigned short)0xDC00)
- #define ((unsigned short)0xDFFF)
- #define 0x010000
- #define (t, l, o) ((o) >= (l) ? (unsigned short)0 : (t)[o])
- #define true

Functions

- void (void)
- int VG_HELPER_get_utf (const unsigned short *textCharRam, int length, int *offset)

Gets the UTF character from a text buffer at the given offset and updates the offset to point to the next character.

- `const jfloat * (const jfloat *matrix)`
- `uint32_t (uint32_t color, uint32_t alpha)`
- `void (jfloat *dest, jfloat x, jfloat y, const jfloat *matrix)`

Variables

- `int32_t`

2.15.1. Detailed Description

MicroEJ MicroVG library low level API: helper to implement library natives methods.

Author. MicroEJ Developer Team

Version. 6.0.1

2.15.2. Function Documentation

2.15.2.1. VG_HELPER_get_utf()

```
int VG_HELPER_get_utf (const unsigned short * text, int length, int *  
offset)
```

Gets the UTF character from a text buffer at the given offset and updates the offset to point to the next character.

Some characters have some special values; they are made up of two Unicode characters in two specific ranges such that the first Unicode character is in one range (for example 0xD800-0xD8FF) and the second Unicode character is in the second range (for example 0xDC00-0xDCFF). This is called a surrogate pair.

If a surrogate pair is incomplete (missing second character), this function returns "0" (error) and does not update the offset.

Parameters .

Table 2.2.

in	text	text buffer encoded in UTF16 where to read UTF character.
in	length	length of the text buffer.
	[in/out]	offset: offset in the text buffer where to read UTF character. Updated to the next character position.

Returns. The decoded UTF character.

Definition at line 164 of file `vg_helper.c`.

```
164      {
165          highPart = GET_NEXT_CHARACTER(textCharRam, length, *offset);
166          ret = 0;
167
168          ((highPart >= MIN_HIGH_SURROGATE) && (highPart <=
169              MAX_HIGH_SURROGATE)) {
170              (*offset < (length - 1)) {
171                  lowPart = GET_NEXT_CHARACTER(textCharRam, length,
172                      *(offset) + 1);
173
174                  ((lowPart >= MIN_LOW_SURROGATE) && (lowPart <=
175                      MAX_LOW_SURROGATE)) {
176                      *offset += 2;
177
178                      ret = 0;
179                      ret += ((int)highPart - ()MIN_HIGH_SURROGATE);
180                      ret <= (int)10;
181                      ret += ((int)lowPart - ()MIN_LOW_SURROGATE);
182                      ret += (int)MIN_SUPPLEMENTARY_CODE_POINT;
183                  }
184              }
185          } {
186              *offset += 1;
187
188              ret = 0x0000FFFF & (int)highPart;
189          }
190
191          ret;
192 }
```

Index

B

- bsp/vg/inc/vg_configuration.h, 2
- bsp/vg/inc/vg_freetype.h, 3
- bsp/vg/inc/vg_helper.h, 3
- bsp/vg/inc/vg_path.h, 6
- bsp/vg/src/LLVG_BVI_impl_stub.c, 7
- bsp/vg/src/LLVG_FONT_impl_freetype.c, 8
- bsp/vg/src/LLVG_FONT_impl_stub.c, 8
- bsp/vg/src/LLVG_GRADIENT_impl_stub.c, 9
- bsp/vg/src/LLVG_MATRIX_impl.c, 9
- bsp/vg/src/LLVG_PAINTER_impl.c, 10
- bsp/vg/src/LLVG_PATH_impl_dual.c, 11
- bsp/vg/src/LLVG_PATH_impl_single.c, 11
- bsp/vg/src/LLVG_PATH_impl_stub.c, 12
- bsp/vg/src/vg_freetype_path.c, 13
- bsp/vg/src/vg_helper.c, 13

M

- MICROVG_CONFIGURATION_VERSION
 - vg_configuration.h, 2
- MICROVG_PATH_HEADER, 1

V

- vg_configuration.h
 - MICROVG_CONFIGURATION_VERSION, 3
- vg_helper.c
 - VG_HELPER_get_utf, 14
- vg_helper.h
 - VG_HELPER_get_utf, 5
 - VG_HELPER_NULL_GRADIENT, 5
- VG_HELPER_get_utf
 - vg_helper.c, 14
 - vg_helper.h, 5
- VG_HELPER_NULL_GRADIENT
 - vg_helper.h, 5