

# SpeedText 2

## 函数指南

Copyright © Christian Klaussner

## 1. 函数

### Function **TextInitialize**(buffer)

初始化SpeedText库。此函数需要在此函数的任何其他函数之前调用库否则，所有其他功能都将失败。

参数缓冲区表示默认的渲染缓冲区。通常BackBuffer（）用于默认缓冲器您可以在初始化后通过调用TextSetBuffer来更改它。

如果初始化成功，则返回True，否则返回False。

### Function **TextDeinitialize**()

取消初始化SpeedText库并删除其所有资源（已创建的字体除外使用TextLoadFont。您应该在取消初始化之前使用TextFreeFont删除它们）。

如果取消初始化成功，则返回True，否则返回False。

### Function **TextSetBuffer**(buffer)

设置当前渲染缓冲区。此缓冲区将成为TextDraw和TextDrawRect的目标。

### Function **TextGetBuffer**()

返回当前渲染缓冲区。

### Function **TextFreeFont**(font)

删除指定的字体。调用此函数后，您不能再使用字体句柄

### Function **TextSetFont**(font)

设置绘图和度量计算的当前字体，例如TextStringWidth和TextFontWidth。为字体指定零以使用默认字体（Courier）

## Function **TextGetFont()**

返回当前选定的字体。如果使用TextSetFont未选择任何字体，则返回值为零。

## Function **TextLoadFont**(fontname\$, height, bold, italic, underline, quality, filename\$)

加载具有指定特征的字体。参数fontname\$表示字体的名称，用于例如“Times New Roman”。如果字体未安装在系统上，但可作为TTF或OTF使用文件，可以为参数filename\$指定文件名。请注意，字体的真实名称仍然需要待指定

参数height确定以像素为单位的字体高度。参数粗体、斜体和下划线可以设置为正确或错误。

参数质量可以是以下值之一：

TEXT_DEFAULT	使用操作系统使用的质量设置.
TEXT_NONANTIALIASED	不使用抗锯齿文本.
TEXT_ANTIALIASED	使用抗锯齿文本.
TEXT_CLEARTYPE	使用ClearType抗锯齿文本.

如果加载成功，则返回新的字体句柄，否则为零..

## Function **TextDraw**(x, y, text\$, ax, ay, encoding)

将文本绘制到当前选定的缓冲区的位置（x，y）。参数ax和ay表示各个轴的对齐。。

水平对齐的值:

TEXT_LEFT	参考点在文本的左边缘..
TEXT_CENTER	文本水平居中于参考点.
TEXT_RIGHT	参考点在文本的右边缘.

垂直对齐的值:

TEXT_TOP	参考点在文本的上边缘.
TEXT_MIDDLE	文本垂直居中于参考点.
TEXT_BOTTOM	参考点在文本的底部边缘.

有关有效的编码类型，请参阅“文本编码”一章。

Function **TextDrawRect**(x, y, width, height, text\$, ax, format, encoding)

使用矩形和格式选项将文本绘制到当前选定的缓冲区中。格式为基于由宽度和高度指定的矩形。如果您不知道矩形的大小，请指定-宽度和高度均为1。参数ax可以是TEXT\_LEFT、TEXT\_CENTER或TEXT\_RIGHT。有关详细信息，请参见TextDraw。

参数格式可以是以下值之一:

TEXT_WORDWRAP	如果单词不适合矩形，则换行。
TEXT_DONTCLIP	不适合矩形的文本将不会被剪裁。

有关有效的编码类型，请参阅“文本编码”一章。

Function **TextLockBuffer**()

锁定当前选定的缓冲区以加快文本渲染。缓冲区只能用于文本呈现在锁定时使用TextDraw和TextDrawRect.

Function **TextUnlockBuffer**()

解锁当前选定的缓冲区。

Function **TextFontWidth**()

返回当前选定字体中最宽字符的宽度

Function **TextFontHeight**()

返回当前选定字体中最高字符的高度。

Function **TextFontAscent**()

返回当前所选字体的升序。字体的上升和下降之和等于身高。

Function **TextFontDescent()**

返回当前选定字体的降序。字体的上升和下降之和等于身高。

Function **TextStringWidth**(text\$, encoding)

返回字符串文本的宽度。当前选择的字体用于测量。

有关有效的编码类型，请参阅“文本编码”一章。

Function **TextStringHeight**(text\$, encoding)

返回字符串文本的高度。当前选择的字体用于测量。

有关有效的编码类型，请参阅“文本编码”一章。

Function **TextSetColor**(red, green, blue)

设置以RGB格式进行文本渲染的颜色。

Function **TextSetBackground**(red, green, blue)

设置以RGB格式进行文本渲染的背景色。为所有颜色指定-1以使背景透明。

Function **TextColorRed**()

返回当前选定颜色的红色成分。

Function **TextColorGreen**()

返回当前选定颜色的绿色成分。

Function **TextColorBlue()**

返回当前选定颜色的蓝色成分。

Function **TextBackgroundRed()**

返回当前所选背景色的红色成分。

Function **TextBackgroundGreen()**

返回当前所选背景色的绿色成分。

Function **TextBackgroundBlue()**

返回当前所选背景色的蓝色成分。

## 2. 文本编码

文本可以使用ANSI或Unicode ( UTF-8 ) 进行编码。标准Blitz3DIDE使用的ANSI字符串足够大多数西方语言，如英语或德语。如果你需要中文或阿拉伯语文本，你必须使用记事本等合适的编辑器将其保存为UTF-8格式。所有接受文本字符串作为参数的SpeedText函数都具有额外的参数编码。根据文本的方式，可以指TEXT\_ANSI或TEXT\_UTF8编码。

## 3. 性能问题

在Windows Vista中使用SpeedText时存在性能问题。标准Blitz3D的绘图性能在大多数情况下，由于Windows Vista的图形体系结构，text比SpeedText更快。没有出现此问题在Windows XP和Windows 7 RC上。