# orxTEXT structure

## Summary

### Text

```
[TextTemplate]
String = "string to display"
Font   = FontTemplate
```

### Font

```
[FontTemplate]
Texture            = path/to/ImageFile.ext
TextureOrigin      = <vector>
TextureSize        = <vector>
KeepInCache        = <bool>
CharacterList      = "ordered list of character"
CharacterSize      = <vector>
CharacterHeight    = <float>
CharacterWidthList = <float>
CharacterSpacing   = <vector>
```

## Details

### Text

Here's a list of available properties for an `orx`TEXT structure:

- `Font`: Specifies a font to use for this text. If no value is specified, orx's default font [1] will be used. **NB: If its value begins with a dollar ('$'), the rest of the value will be used as a key for the localization module.**
- `String`: Defines the content of this orxTEXT as plain text. **NB: If its value begins with a dollar ('$'), the rest of the value will be used as a key for the localization module.** See below for an example of multiline text.

Here's a small example.

```
[Text1]
String = This is our first text
Font   = Font1

[Text2]
String = $Greetings
Font   = $FontKey
```

```
[MultilineText]
String = "First line
Second line"
Font   = Font1
```

`Text1` has a static plain text content with a static custom font called `Font1` whereas `Text2` points to the localization keys named `Greetings` for its text content and `FontKey` for its font. This means that its content and font are defined in the orxLOCALE module and will always be expressed according to the current selected language.

*NB: If the current language is changed at runtime, the text's content and font will be automagically updated without requiring any code.*

### Font

Here's a list of available properties for an `orxFONT` structure:

- `Texture`: Specifies which bitmap file to use as texture for our font object.
- `TextureOrigin`: Specifies the top left origin of the first defined character in the bitmap file, in pixels (Z coordinate being ignored). By default its value is (0, 0, 0) which means the texture for the font object will begin at the top left of the bitmap file. Note: TextureSize must also be supplied for this to take effect.
- `TextureSize`: Specifies the size, in pixels, of the bitmap file where characters are defined (Z coordinate being ignored). By default it will use the whole bitmap, ie. its value will be the size of the bitmap.
- `CharacterList`: Specifies all the characters that are defined in the bitmap file, ie. all the characters with a glyph in the texture file will have to be specified in order of appearance. The string can be encoded in strict ANSI or UTF-8. **ISO-Latin-1 is not supported!** To avoid special characters to be handled by orx's config parser [2], the string is likely to be defined as a block using the block marker ('"'). **In order to specify '"' itself in the string, this character needs to be doubled and not in the first position!**
- `CharacterSpacing`: A vector specifying the empty spaces between characters, in pixels (Z coordinate being ignored).
- `CharacterSize`: A vector specifying the size of a character, in pixels (Z coordinate being ignored). **If this property is defined, orx assumes the font is monospaced (ie. fixed width, all the characters need to be organized in a grid manner). See the command line tool orxFontGen to see how to create custom bitmap fonts from a given TrueType font file.**
- `CharacterHeight`: This is only used for variable width fonts, ie. when `CharacterSize` is **not** defined. In this case, this property has to be a strictly positive value that will be the font character's height.
- `CharacterWidthList`: Only used for variable width fonts, ie. when `CharacterSize` is **not** defined. In this case, this contains a list of widths for every character defined in `CharacterList` in the same order.

An example of how to use custom fonts can be found in the localization tutorial [3].

# orxFontGen

orxFontGen is a command-line tool that ships with Orx which allows you create bitmap fonts which can be define with the configuration detailed above. It can be found in the orx/tools/orxFontGen/bin folder. Available to compile for Windows, Linux and Mac. To learn how to use it follow the orxFontGen tutorial.

## Latest config settings for the Development Version

We endeavor to keep the config properties on this page up to date as often as possible. For up to the minute config information for the latest version of Orx, check the most recent published at:

CreationTemplate.ini and

SettingsTemplate.ini

Additionally these files can be found under your orx source tree in the `orx/code/bin` folder.

1)
Dina, an open source font by Jørgen 'Jibs' Ibsen
2)
such as '#', ';', '“', '~', '$', ...
3)
at the end of 10_Locale.ini

From:
https://www.orx-project.org/wiki/ - **Orx Learning**

Permanent link:
**https://www.orx-project.org/wiki/en/orx/config/settings_structure/orxtext?rev=1692950400**

Last update: **2025/09/30 17:26 (14 months ago)**