# ENHANCEMENT SUMMARY

## STM32F4 CHROM-ART SW-RENDER FOR NO O/S (8.0.1)

*Altia® DeepScreen®*

*March 18, 2016*

## Please read these important notices before using this release.

**This document assumes the Altia software is installed in c:\usr\altia**

Replace occurrences of c:\usr\altia with YOUR_INSTALL_DIR if the installation directory is different.

## TABLE OF CONTENTS

# 1.0 Licensing

This target requires the purchase of DeepScreen license 1085.

# 2.0 Altia Design Compatibility

This target is compatible with Altia Design 11.3 and later.

# 3.0 Release 8.0.1 (March 18, 2016)

## 3.1 New Features

No new features were added.

## 3.2 Enhancements and Fixes

- March 10, 2016, US4817 - TA6447 – Flickering issue caused by blocking NOR flash accesses. DIB format generation was removed which causes continuous accesses to the external memory each time the image is rendered. The concurrent accesses to the Flexible Memory Controller by both the CPU and LCD controller causes a starvation to the LTDC frame buffer updates.
- March 10, 2016, US4817 – TA11031 – Alpha Mask Objects/feature did not render correctly in previous release. Color Allocation in the driver was updated to treat the mask part of Alpha Mask object. Also, FBNOTRANS was uncommented in servermd to enable the correct rendering of alpha pixels.
- March 10, 2016, US4817 – TA11030 – Trailing Artifacts with moving/animated images. This was corrected by kicking off the DMA2D ChromART from the VSYNC interrupt, instead of driver_StartGraphics (before the graphics are rendered to the back buffer). The re-architecture corrected the delayed access to the external SDRAM via the Flexible Memory Controller.
- March 16, 2016, – Distorted scaled/stretched Text issue. This was fixed by removing the use of the DIB image format generation. The DIB image format has performance issues that results in display flickering.
- March 16, 2016, US4817 – TA11033 – Fixed issue with transformed images (Scaled/Stretched/Distorted) appear with black pixels in transparent areas. This occurred only with opaque images i.e. images without any alpha component. The issue was eliminated when the custom driver fbCopyArea was disabled and replaced by the standard fbCopyArea. The custom fbCopyArea had no Alpha blending capability.
- March 16, 2016, US7054 – TA11503 – Fixed Frame Buffer corruption that was due to invalid Parameters being passed to the DMA2D peripheral by the altiaGL pipeline. The issue was resolved by removing the custom driver fbCopyArea.
- February 16, 2016, DE929: Fixed issue where altiaAPIServer.c was not sending the proper value to an integer client from an external connection registered as float (for example, float events coming from a slider in DeepScreen to be sent to Altia block in Simulink executable which is built as an integer client). This issue is resolved.
- February 18, 2016, DE1166: Fixed corrupted pixels resulting from transformed draw operations inside the Mask container of an Alpha Mask object.

- February 16, 2016, DE1165: Fixed issue where transformed Image Objects (from the file system) would not draw properly inside the Mask container of an Alpha Mask object.
- February 16, 2016, DE1144: Fixed issue where objects would not be drawn in a small alpha mask object positioned on the top or right side of the screen.
- February 12, 2016, DE1143: Fixed issue where images inside an Alpha Mask would not draw correctly when using a DDB .gen file at code generation time.
- February 10, 2016, DE1124: Fixed issue when drawing overlapping Alpha Mask objects where the bottom-most object is transformed.
- February 10, 2016, DE1125: Fixed undefined symbol when using Alpha Mask objects with monochrome fonts and software image scaling.
- February 1, 2016, DE1132: Fixed text macro for error messages in altiaLibAlphaMask.c when using the Unicode build configuration.
- January 27, 2016, DE1127: Fixed a crash when the 3D Scene Object dimensions are greater than the render to texture dimension. See the 3D Scene Object section of the User's Guide for instructions to override the default dimensions of 1024x768.
- January 26, 2016, DE1126: Fixed a rendering issue where transparent 3D meshes are showing their back faces.
- January 25, 2016, DE1123: Fixed a link failure when the design includes 3D Scene Objects and "Image Objects use file system" code gen option is checked.

## 3.3 Known Issues

Listed below are known issues new for this release. Known issues for previous releases also still apply unless they are indicated as being fixed for this release or a previous release. Some of the items listed below apply to DeepScreen in general (i.e., this target and other types of targets) and are listed here because they may be of special interest to users generating code for the Intel x86 Software Render Windows target.

- January 21, 2015, DE1121: The 3D Scene object will not render when rotated or scaled on the Intel x86 Software Render Windows Target.
- December 9, 2015, DE1105: Thick lines inside a clip object can cause the clip to shift its origin by half the line thickness. This is true when the thick line object is at the bottom-most or left-most edge of the clip's contents. The result is the clip will appear to be smaller than its specified width/height. This issue has always existed in the DeepScreen engine.
- October 27, 2015, DE1068: Hierarchical draw attributes (like color) will propagate to the children of an Alpha Mask, but changing the draw attributes will not force a redraw of the Alpha Mask's buffers. The buffers will not update until a child inside the Alpha Mask changes (due to direct animation manipulation).
- Nov 11, 2015, DE1082: Rotating an Alpha Mask which contains sibling-justified Text objects can result in a change of text position when the Text objects receive new display text. This issue is not unique to the Alpha Mask object. Instead it's caused by trying to justify the text in the X-direction while the object is rotating. To minimize the problem, use a small sibling centered both horizontally and vertically with the text.

- November 12, 2012, RTC-1589. For a Drawing Area Object (DAO) taken from the models library drawingarea.dsn, the DeepScreen generated code version of the DAO may not work because the DAO is not properly initialized from a Control code EXTERN statement. A DAO taken from the models library drawingarea.dsn (e.g., c:\usr\altia102\models\drawingarea.dsn if Altia Design is installed in c:\usr\altia102) has a Control code WHEN statement for altiaInitDesign that calls an EXTERN ROUTINE to initialize the DAO. The name of the EXTERN ROUTINE is taken from the contents of a Text I/O object using the Text I/O object's "text" animation. For code generation from Altia Design 10.2 or newer, the name of the EXTERN ROUTINE is not correctly taken from the Text I/O. The workaround is to change the Control code on the DAO to set the function name explicitly in the EXTERN ROUTINE instead of taking the name from a Text I/O "text" animation. For example, change the EXTERN ROUTINE statement to look like this:

```
EXTERN  ROUTINE   initialize
PARAMS:  altiaSetObj    _global_altiaApp
LIBRARY:  plotdraw.dll
RESULT:  _global_altiaApp
```

# 4.0 Release 8.0.0 Beta (February 5, 2016)

## 4.1 New Features

- August 7, 2015, US4298: Added Runtime Font engine support.
- July 25, 2015, US4699: Optimized Chrom-Art usage and Eliminated screen tearing by making Chrom-Art the default and only option and draw to 2 separate buffers i.e. Double Buffering. Also includes fix to obtain Pixmap correctly.
- January 28, 2016, US6524: External Resource Configuration to store Image Assets and Fonts to NOR Flash Section (Configure ASSETS section to be stored in external NOR)
- November 2, 2015, US5927: Added support for new "mode" animation in the Alpha Mask Object. A mode of '1' will invert the Mask when it's combined with the Content.
- September 17, 2015, I57: Improved text animation performance by delaying processing for text objects that are not visible, in hidden deck cards, or inactive layers. The processing is performed when the text object first becomes visible after the text animation. This provides a significant performance improvement when loading a language using the language object.

## 4.2 Enhancements and Fixes

- August 7, 2015, US4764: Image Formatter Issue with DDB format for ARGB32 bit formate
- January 27, 2016, US6518, US6506: Updated Makefile and gen files for Breville port, Updated driver.c for new defines in lcd_config header include. Renamed 16

bpp gen files to DIB since there is no dependency on color depth. Either 32bpp or 16bpp can be used with DIB

- February 1, 2016, US6623: OS_Wrapper changes to use CMSIS OS APIs for Delays and timer tick
- January 11, 2016, DE1116: Fixed issue on software render targets that use double buffering. The issue manifests itself when multiple objects in multiple layers (using a display object) are animated in a single update, such as timer expiration. The defect is apparent when the objects being rendered on the lower layer (bottom layer in z order) are boxed with either garbage data or whatever is in the framebuffer of layer above it, which may just be the background color of the display object. The fix is an added check to ensure that a the extent is valid for that layer that is being render before the layer extents are copied to the frame buffer.
- December 8, 2015, DE668: Fixed issue where MLTO would not wrap text when using dynamic memory configuration and text was sent to the MLTO before the MLTO was first drawn.
- December 8, 2015, DE645: Added fail-safe logic to dirty region processing in DeepScreen engine. The new logic will trap the "out of extents" error and rebuild the extent list using the full dimensions of the screen or each redrawn layer.
- December 8, 2015, DE777: Changed comment structure for control code to use double-slash instead of slash-asterisk. This allows the User to create control code comments containing /* and */ characters in the Editor that will safely compile in DeepScreen.
- December 7, 2015, DE1090: Fixed infinite recursion for some designs using nested Mask Objects (introduced with DE753).
- November 3, 2015, DE1078: Fixed text drifting issue when using sibling justified text in an Alpha Mask Object while simultaneously animating the Alpha Mask Object such that it changes position.
- October 15, 2015, DE1043: Fixed typecasting errors in Multiplot object's line fill draw logic. The errors create a filled concave polygon draw scenario which is not supported by some of the target graphic drivers.
- October 8, 2015, DE1028: Multiplot object's line fill draw has rendering errors if the target's graphic driver does not support filled concave polygon draw. This issue has been fixed.
- September 23, 2015, DE1012: Fixed issue where layers would not update in a Layer Manager without a corresponding draw. Introduced with I57.
- September 19, 2015, DE1007: Fixed issue in Multiplot external interface functions that set annotation colors. On some Deepscreen targets the annotations are being rendered with inverted color value.
- September 17, 2015, US5336: Made enhancments to Multiplot object line draw function. Rendering time is greatly reduced by drawing a multi-line segment instead of multiple individual lines for the plot line.
- September 11, 2015, DE964: Fixed issue where the Drawing Area Object would not redraw after calling altiaExUpdate. Also fixed position metrics when using inside a layer that’s not the same height as the display.

- August 25, 2015, DE930: Fixed a clip issue for Multi-Plot and Clip Objects when used inside an Alpha Mask Object.
- August 25, 2015, DE646: Fixed compile issue when using the MultiPlot Object (missing stdio.h header include).
- August 24, 2015, DE929: Fixed issue when setting multiple 3D animations to the same frame in the 3D Scene object. Setting multiple animations to the same frame will now function correctly.
- August 24, 2015, DE922: Fixed issue where 3D Scene generates a fault when setting width greater than 1024 pixels.
- August 24, 2015, US4455: New custom ID feature for 3D Scene object animations.
- August 24, 2015, DE923: Fixed issue when setting multiple 3D animations to the same frame in the 3D Scene object. Setting multiple animations to the same frame will now function correctly.
- August 24, 2015, DE922: Fixed issue where 3D Scene generates a fault when setting width greater than 1024 pixels.
- August 24, 2015, US4458 & US4459: New custom ID feature for 3D Scene object animations.
- August 7, 2015, US3192 & US4089: MISRA 2012 compliance changes for altiaAPI.c and altiaAPICbk.c.
- August 5, 2015, DE891: Fixed issue with the Tickmark object when the radius is 0 and the label is 0. In some cases instead of displaying 0 a value of 1.4305115e-006 (or similar) is displayed.
- July 28, 2015, DE884: The 3D Scene object was not able to open a mesh file when code was generated with the "Unicode Font Characters" code generation option. This issue is now resolved
- July 15, 2015, DE868: Fixed compile issue when using CHILD layers with bit-field optimizations disabled. This build combination will now compile and function properly.
- July 15, 2015, DE869: Fixed issue with Control Code generation when using string to float conversion along with Unicode. The generated control code will now properly use wcstod instead of atof for the Unicode character strings.
- July 14, 2015, DE866: Fixed build errors in DeepScreen generated code for designs containing vector path objects. Errors were due to missing header file include. This issue impacts only the OpenVG targets.
- June 11, 2015, DE833: Fixed syntax error message when generating code for some designs that utilize EXTERN statements in control code. The error occurred when the EXTERN statements used string animations in the parameters for the external API.
- June 1, 2015, US3524: Added support for storing 3D scene assets in memory instead of on a file system. This is now the default behavior. To store scene assets on a file system, check the “3D Scene objects use file system” code generation option.

- May 28, 2015, DE805. Fixed issue when generating code for a design that uses EXTERN control code statements. The external functions were not declared resulting in unknown function errors when compiling control.c
- May 27, 2015, DE806. Fixed issue when generating code for a design containing a Drawing Area Object (DAO) without enabling the Stimulus option. This would result in an unknown function error when compiling stimulus.c

Refer to the target documentation (Overview.pdf) for usage requirements on the new features.

## 4.3 Known Issues

Listed below are known issues new for this release. Known issues for previous releases also still apply unless they are indicated as being fixed for this release or a previous release. Some of the items listed below apply to DeepScreen in general (i.e., this target and other types of targets) and are listed here because they may be of special interest to users generating code for the Intel x86 Software Render Windows target.

- An object that has been moved and scaled may sometimes render off by one pixel. Work item # DE727.
- The 3D Rect object is incorrectly identified as not supported by the Altia Design validator for this target. Work item # DE818.
- When a Drawing Area Object (DAO) is used, rendering issues may occur when text is written to it or objects in the foreground overlay it. Additionally, objects in the foreground which overlay a DAO may appear as completely opaque, regardless of the objects opacity attribute setting. Work items # DE819, DE820, DE821, DE822.
- Incorrect control code may be generated when the following conditions are met: 1: the "Unicode Font Characters" code generation option is used to generate code, 2: the design control code is using a custom ROUTINE and 3: non-string variables are being passed as arguments to the custom ROUTINE. Work item # DE609.
- Dec 9, 2015, DE1105: Thick lines inside a clip object can cause the clip to shift its origin by half the line thickness. This is true when the thick line object is at the bottom-most or left-most edge of the clip's contents. The result is the clip will appear to be smaller than its specified width/height. This issue has always existed in the DeepScreen engine.
- October 27, 2015, DE1068: Hierarchical draw attributes (like color) will propagate to the children of an Alpha Mask, but changing the draw attributes will not force a redraw of the Alpha Mask's buffers. The buffers will not update until a child inside the Alpha Mask changes (due to direct animation manipulation).
- Nov 11, 2015, DE1082: Rotating an Alpha Mask which contains sibling-justified Text objects can result in a change of text position when the Text objects receive new display text. This issue is not unique to the Alpha Mask object. Instead it's caused by trying to justify the text in the X-direction while the object is rotating. To minimize the problem, use a small sibling centered both horizontally and vertically with the text.
- November 12, 2012, RTC-1589. For a Drawing Area Object (DAO) taken from the models library drawingarea.dsn, the DeepScreen generated code version of the DAO may not work because the DAO is not properly initialized from a Control code EXTERN statement. A DAO taken from the models library drawingarea.dsn

(e.g., c:\usr\altia102\models\drawingarea.dsn if Altia Design is installed in c:\usr\altia102) has a Control code WHEN statement for altiaInitDesign that calls an EXTERN ROUTINE to initialize the DAO. The name of the EXTERN ROUTINE is taken from the contents of a Text I/O object using the Text I/O object's "text" animation. For code generation from Altia Design 10.2 or newer, the name of the EXTERN ROUTINE is not correctly taken from the Text I/O. The workaround is to change the Control code on the DAO to set the function name explicitly in the EXTERN ROUTINE instead of taking the name from a Text I/O "text" animation. For example, change the EXTERN ROUTINE statement to look like this:

```
EXTERN ROUTINE  initialize
PARAMS:  altiaSetObj  _global_altiaApp
LIBRARY:  plotdraw.dll
RESULT:  _global_altiaApp
```

- Oct 13, 2012, RTC-1527. When DeepScreen code is generated for Control and the Control code uses GLOBALS, the GLOBALS are defined as static variables in the generated altia/control.c source file. They are not initialized (as in: static int _whatever = 0;). It is expected that all C compilers initialize static variables to zero if they are not explicitly initialized. Some embedded compilers, such as TI Code Composer Studio, do not initialize static variables to zero. The workaround is to explicitly initialize Control GLOBALS to 0 in a Control WHEN altiaInitDesign block within the design.

- Oct 16, 2012, RTC-1532. Distorted Text I/O objects that use a sibling justify mode must have a sibling to which it can justify. If the sibling is missing, the object may drift when text changes. To resolve the issue, put a sibling object into the Group, Deck card, or Clip containing the Text I/O so that it has a sibling object to which it can justify itself (since a sibling justify mode was chosen).

# 5.0 Release 7.2.1 (May 22, 2015)

## 5.1 New Features

- Chrom-ART (DMA2D) support added to target. BSP dependencies must be satisfied for this feature to build and function properly. Work item # 3291.

Refer to the target documentation (Overview.pdf) for usage requirements on the new features.

## 5.2 Resolved Issues

Resolved the following issues:

- Updated target to use new BSP APIs to work-around BSP system timer API limitations / defects. Work item # US3924.
- Updated target to address Chrom-ART timeouts reported by the STM32 HAL. When a Chrom-ART timeout occurred, an unexpected graphical artifact could appear on the screen. Work item # US3949, DE790.
- Fixed screen fading/flashing and object screen ghosting during CPU intensive render operations. Work items # DE788, US3925.
- Fixed objects drawing outside of canvas or wrapping around to other side of canvas. Work item # DE792.
- Objects having a transparency mask that was applied via the editor are now consistently drawn correctly. Work items # DE791, DE797.
- Fixed build system issue wrt path names containing white space character(s). Work items # DE799.

## 5.3 Known Issues

The following issues are known to exist in this version of the target:

- An object that has been moved and scaled may sometimes render off by one pixel. Work item # DE727.
- The 3D Rect object is incorrectly identified as not supported by the Altia Design validator for this target. Work item # DE818.
- When a Drawing Area Object (DAO) is used, rendering issues may occur when text is written to it or objects in the foreground overlay it. Additionally, objects in the foreground which overlay a DAO may appear as completely opaque, regardless of the objects opacity attribute setting. Work items # DE819, DE820, DE821, DE822.
- Incorrect control code may be generated when the following conditions are met: 1: the "Unicode Font Characters" code generation option is used to generate code, 2: the design control code is using a custom ROUTINE and 3: non-string variables are being passed as arguments to the custom ROUTINE. Work item # DE609.

# 6.0 Release 7.2.0

## 6.1 New Features

- Chrom-ART (DMA2D) support added to target. BSP dependencies must be satisfied for this feature to build and function properly. Work item # 3291.

Refer to the target documentation (Overview.pdf) for usage requirements on the new features.

## 6.2 Resolved Issues

Resolved the following issues:

- n/a

## 6.3 Known Issues

The following issues are known to exist in this version of the target:

- n/a

# 7.0 Release 7.1.0 Beta

## 7.1 New Features

- Added compile time support for 16bpp (default) vs 32bpp framebuffers using LCD_CONFIG_16_BPP and LCD_CONFIG_32_BPP respectively. The target driver currently expects this C preprocessor macro to be defined in the BSP's LCD.h file. Work item # US3599.

Refer to the target documentation (Overview.pdf) for usage requirements on the new features.

## 7.2 Resolved Issues

Resolved the following issues:

- Updated target touch screen driver (input.c) to provide work-around for erroneous "release" events being reported by the BSP's low level touch screen interface APIs. Work item # US3599.
- Verified IAR compiler defect concerning improper sign extension of bit fields is fixed. IAR 7.30.4, or later, is required. Removed work-around .gen files from target and updated documentation. Work item # US3599.
- Miscellaneous MISRA 2012 compliance changes for various common source files. Work items # US3252, DE710, DE711.
- Fixed accidental ALTIA_NO_BACKGROUND_FILL #define override in altiaBase.h. Work item # DE754.
- Sibling justification fixed for text IO and multi-line text objects. Work item # DE735.
- Raster RAM data optimizations. Work items # US3429, US3430.
- Altia tasking feature improved to better handle slower draw times when the task time is in single digit milliseconds. Work item # US3774.

### 7.3 Known Issues

The following issues are known to exist in this version of the target:

- Target does not yet have Chrom-ART accelerator support. Once added, overall target performance will increase. Work item # 3291.

## 8.0 Release 7.0.0 Beta

### 8.1 New Features

- n/a

Refer to the target documentation (Overview.pdf) for usage requirements on the new features.

### 8.2 Resolved Issues

Resolved the following issues:

- n/a

### 8.3 Known Issues

The following issues are known to exist in this version of the target:

- Target does not yet have Chrom-ART accelerator support. Once added, overall target performance will increase. Work item # 3291.

## 9.0 Acknowledgement of 3rd Party Software Used by this Software Program

This Software Program may include software developed by the Apache Software Foundation:

> Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at: <http://www.apache.org/licenses/LICENSE-2.0>
>
> Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This Software Program may include the open source libjpeg library, the work of the Independent JPEG Group, <http://ijg.org>.

This Software Program may include the open source zlib library, the work of Jean-loup Gailly and Mark Adler, <http://www.zlib.net>.

This Software Program may include the open source libpng library, the work of Guy Eric Schalnat, Andreas Dilger, Glenn Randers-Pehrson and others, <http://www.libpng.org>.

The GNU make utility distributed with this target is covered by the GPLv3 license: http://www.gnu.org/licenses/gpl-3.0.html. Source code for the GNU make utility is available from the MinGW website: http://www.mingw.org/.

Portions of this software may include software from The FreeType Project. These portions are copyright (c) 2006-2008 The FreeType Project (<http://www.freetype.org>). All rights reserved. These portions fall under The FreeType Project LICENSE agreement:

```
                    The FreeType Project LICENSE
                    ----------------------------
                            2006-Jan-27

                    Copyright 1996-2002, 2006 by
          David Turner, Robert Wilhelm, and Werner Lemberg

Introduction
============

  The FreeType  Project is distributed in  several archive packages;
  some of them may contain, in addition to the FreeType font engine,
  various tools and  contributions which rely on, or  relate to, the
  FreeType Project.

  This  license applies  to all  files found  in such  packages, and
  which do not  fall under their own explicit  license.  The license
  affects  thus  the  FreeType   font  engine,  the  test  programs,
  documentation and makefiles, at the very least.

  This  license   was  inspired  by  the  BSD,   Artistic,  and  IJG
  (Independent JPEG  Group) licenses, which  all encourage inclusion
  and  use of  free  software in  commercial  and freeware  products
  alike.  As a consequence, its main points are that:
```

- We don't promise that this software works. However, we will be interested in any kind of bug reports. (`as is' distribution)

- You can use this software for whatever you want, in parts or full form, without having to pay us. (`royalty-free' usage)

- You may not pretend that you wrote this software. If you use it, or only parts of it, in a program, you must acknowledge somewhere in your documentation that you have used the FreeType code. (`credits')

We specifically permit and encourage the inclusion of this software, with or without modifications, in commercial products. We disclaim all warranties covering The FreeType Project and assume no liability related to The FreeType Project.

Finally, many people asked us for a preferred form for a credit/disclaimer to use in compliance with this license. We thus encourage you to use the following text:

"""
Portions of this software are copyright © <year> The FreeType
Project (www.freetype.org). All rights reserved.
"""

Please replace <year> with the value from the FreeType version you actually use.

## Legal Terms

### 0. Definitions

Throughout this license, the terms `package', `FreeType Project', and `FreeType archive' refer to the set of files originally distributed by the authors (David Turner, Robert Wilhelm, and Werner Lemberg) as the `FreeType Project', be they named as alpha, beta or final release.

`You' refers to the licensee, or person using the project, where `using' is a generic term including compiling the project's source code as well as linking it to form a `program' or `executable'. This program is referred to as `a program using the FreeType engine'.

This license applies to all files distributed in the original FreeType Project, including all source code, binaries and documentation, unless otherwise stated in the file in its original, unmodified form as distributed in the original archive. If you are unsure whether or not a particular file is covered by this license, you must contact us to verify this.

The FreeType Project is copyright (C) 1996-2000 by David Turner, Robert Wilhelm, and Werner Lemberg. All rights reserved except as specified below.

### 1. No Warranty

THE FREETYPE PROJECT IS PROVIDED `AS IS' WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT WILL ANY OF THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY DAMAGES CAUSED BY THE USE OR THE INABILITY TO USE, OF THE FREETYPE PROJECT.

### 2. Redistribution

This license grants a worldwide, royalty-free, perpetual and irrevocable right and license to use, execute, perform, compile, display, copy, create derivative works of, distribute and sublicense the FreeType Project (in both source and object code forms) and derivative works thereof for any purpose; and to authorize others to exercise some or all of the rights granted herein, subject to the following conditions:

- Redistribution of source code must retain this license file (`FTL.TXT') unaltered; any additions, deletions or changes to the original files must be clearly indicated in accompanying documentation. The copyright notices of the unaltered, original files must be preserved in all copies of source files.

- Redistribution in binary form must provide a disclaimer that states that the software is based in part of the work of the FreeType Team, in the distribution documentation. We also encourage you to put an URL to the FreeType web page in your documentation, though this isn't mandatory.

These conditions apply to any software derived from or based on the FreeType Project, not just the unmodified files. If you use our work, you must acknowledge us. However, no fee need be paid to us.

### 3. Advertising

Neither the FreeType authors and contributors nor you shall use the name of the other for commercial, advertising, or promotional purposes without specific prior written permission.

We suggest, but do not require, that you use one or more of the following phrases to refer to this software in your documentation or advertising materials: `FreeType Project', `FreeType Engine', `FreeType library', or `FreeType Distribution'.

As you have not signed this license, you are not required to accept it. However, as the FreeType Project is copyrighted material, only this license, or another one contracted with the authors, grants you the right to use, distribute, and modify it. Therefore, by using, distributing, or modifying the FreeType Project, you indicate that you understand and accept all the terms of this license.

4. Contacts
-----------

There are two mailing lists related to FreeType:

- o freetype@nongnu.org

  Discusses general use and applications of FreeType, as well as future and wanted additions to the library and distribution. If you are looking for support, start in this list if you haven't found anything to help you in the documentation.

- o freetype-devel@nongnu.org

  Discusses bugs, as well as engine internals, design issues, specific licenses, porting, etc.

Our home page can be found at

http://www.freetype.org