Geometric Tools Engine 8 Update History

Contents

The version release dates are listed here. Versions released before the current version may be obtained by email request.

- Version 8.0 posted May 12, 2025.

The updated files and related notes are provided for the versions in each of the ensuing sections. Modified files are colored gold, new files are colored green, and deleted files are colored red. Source code is colored Violet. Source code modifications that occur in a section *Updates to Version x.y* are pushed to github and uploaded to the Geometric Tools website. However, the zip file is still for version $x.y$. The modified files will be contained in the zip file for version $x.(y + 1)$ and posted once that version is ready for shipping.

1 Updates to Version 8.0

August 20, 2025 Implemented the occlusion query for the OpenGL graphics engine.

GTE/Graphics/GL46/GL46Engine.{h,cpp}

August 8, 2025 The function OpenGLReportListener was incorrect regarding throwing exceptions. The throw occurred inside the default case of the switch statement. It had to be outside the switch statement and inside the code block for when code is not zero. After making this change, the Intel Graphics driver on my laptop threw an exception in the GLSL reflection code when checking for a GL error in glProgramInterfaceiv for programInterface is GL_TRANFORM_BUFFER. I have seen this behavior over the years. I thought that the OpenGL specification required this to be handled. I added a block of code to trap this special case and not call ReportGLError. I also commented out the preprocessor definition GTE_GL46_THROW_ON_REPORT_LISTENER_WARNING in case other such GL errors are unexpectedly generated.

GTE/Graphics/GL46/GL46.cpp

July 17, 2025. Modified the default version of OpenGL from `#version 430` to `#version 460`. Modified comments to reference OpenGL 4.6 instead of OpenGL 4.5.

GTE/Graphics/GL46/GL46Engine.h
GTE/Graphics/GL46/GL46ProgramFactory.{h,cpp}

July 14, 2025. Removed two unused class members.

GTE/Samples/Graphics/MultipleRenderTargets/MultipleRenderTargetsWindow3.h

June 26, 2025. Fixed a typographical error in a comment.

GTE/Mathematics/DistAlignedBoxAlignedBox.h

June 24, 2025. After removing support for MSVS 2019, some files in the GenerateProject tools were deleted. However, I had failed to remove the project references to those files, and the project failed to compile.

GTE/Tools/GenerateProject/GenerateProject.v17.vcxproj
GTE/Tools/GenerateProject/GenerateProject.v17.vcxproj.filters
GTE/Tools/GenerateProject/ProjectTemplate.cpp
GTE/Tools/GenerateProject/GenerateProject.cpp

June 8, 2025. For the OpenGL trace system, I added wrappers to hide the Microsoft OpenGL 1.0 and 1.1 DLL functions. This is a hack for now which uses the gte namespace. Fighting the gl* names for the DLL functions when the remaining extensions have wrappers around them with gl* names was painful. A later modification of GTE OpenGL will rename all the wrappers Gl* to avoid the hack.

GTE/Graphics/GL46/GL46.h
GTE/Graphics/GL46/GL46.cpp

2 Initial Distribution of Version 8.0

May 12, 2025. The main modifications are listed here.

- According to the Microsoft Product Fixed Lifecycle Policy, the Mainstream Support for Microsoft Visual Studio 2019 ended in April 2024. I have removed the solutions and projects for Microsoft Visual Studio 2019, now supporting only Microsoft Visual Studio 2022. My contract load is sufficiently large that reducing development and support time for my freely available GTE is necessary. If you need to build the code using Microsoft Visual Studio 2019, you can download the Community Edition and select the platform toolset to be v142 (MSVS 2019) instead of the default v143 (MSVS 2022).

- The GTE OpenGL graphics code supported only through OpenGL 4.5. The features for OpenGL 4.6 are extremely small, so I added support for them. Any names involving GL45 now use GL46 instead. This includes class names and file names. I downloaded the latest headers from Khronos (such as glcorearb.h) and use them instead of the older ones that I was using for OpenGL 4.5.

- I added the ability to generate traces for OpenGL function calls. You can enable this by adding to your OpenGL projects the preprocessor symbol GTE_ENABLE_GLTRACE. Alternatively, you can expose the define of this symbol in GTE46.cpp (line 139). When enabled, the trace is written to the file GLTrace.txt in the working directory (when running inside Visual Studio) or the executable directory (when running outside Visual Studio). I found the traces to be useful when rapid prototyping using GTE but then extracting the native OpenGL code for clients to use.