

Candera V3.4.0

Release Date: May 2017

To keep backwards compatibility for any interface change in the public API the old interface has been kept and marked as deprecated. The deprecated interfaces will be removed in future versions. Compiler warnings will be generated when using a deprecated interface through the usage of Candera Macro `CANDERA_DEPRECATED_3_4_0`.

- [Candera/FeatStd CMake Build System Changes](#)
- [Candera Engine 3D](#)
- [Candera Engine 2D](#)
- [Candera Base](#)
- [Candera Scripting](#)
- [Candera System](#)
- [Candera Device](#)
- [Candera Transitions](#)
- [Candera Globalization](#)
- [TextEngine](#)
- [Candera Widgets](#)
- [Candera Monitor](#)
- [Candera V3.4.0.1](#)

Candera/FeatStd CMake Build System Changes

Added flags:

Name	Description	Possible values
MONITOR_FRAME_DEBUGGER_ENABLE D	Enables basic frame debugger (frame timeline capture, single step mode, no OpenGL specific features).	ON/OFF
MONITOR_FRAME_DEBUGGER_OPENGL_CAPTURE_ENABLED	Enables OpenGL specific Frame Debugger features.	ON/OFF

Candera Engine 3D

Uniformbuffers for Light and Material

[Candera Light](#) and [Material](#) parameters are now passed as uniform buffers to shaders whenever possible. The [Candera](#) reference shader include files have been updated so that any existing shader using Candera's [Light](#) and/or [Material](#) automatically benefits from this new feature without any change necessary. The [Renderer::Statistics](#) has been enhanced by the number of instantiated uniform buffer objects and the number of uniform buffer objects that were updated.

Renderer Statistics Overlay

This feature enables the user to draw renderer statistics data into a render target for diagnostic purposes. A common case would be to draw it every frame after rendering all objects so that it constitutes an overlay presenting relevant rendering data to the user in realtime. The color of the overlay's text and values can be configured separately, as well as the properties of the appearance used to draw the overlay. Added the ability to draw only parts of a [VertexBuffer](#) to `RenderDevice`.

See also:

[RendererStatisticsOverlay](#).

New 3D Animation Property Setters

New animation property setters have been implemented for the Viewport ([Camera::GetViewport](#)) and ScissorRectangle ([Camera::GetScissorRectangle](#)) properties of a [Camera](#) object from a 3D [Scene](#). The corresponding classes for these are [Animation::CameraViewportPropertySetter](#) and [Animation::CameraScissorRectanglePropertySetter](#).

A new animation property setter has been implemented also for the vertex rectangle property of renderable canvas objects. The corresponding class for this is [Animation::CanvasVertexRectanglePropertySetter](#).

Transform Feedback

[Candera](#) 3.4.0 supports transform feedback on OpenGL ES3.0 platforms. This feature enables the user to utilize the vertex processing unit of the GPU to perform calculations and reuse the results. This can be used for debugging vertex transformations, speed up particle systems, or reuse skinned mesh data for physics calculations.

Pixel Buffer Objects

[Candera](#) 3.4.0 supports pixel buffer objects on OpenGL ES3.0 platforms. This feature can be used transfer data to and from video memory asynchronously. The [Renderer::Statistics](#) has been enhanced by the number of instantiated pixel buffer objects.

See also:

[Renderer::UploadPixelBuffer](#).

Submesh

This version introduces the ability to draw a subset of a vertex buffer.

See also:

[VertexBuffer::SetElementStart](#) and [VertexBuffer::SetElementCount](#).

RenderMode Rasterizer Discard

The [RenderMode](#) now offers an option to discard rasterization. This means that vertex shaders are processed while fragment shaders are not. This is useful in conjunction with transform feedback. This feature is OpenGL ES3.0 only.

See also:

[RenderMode::SetRasterizerDiscardEnabled](#).

Hashes for RenderOrderBin Comparison

During runtime, the RenderOrderBins have been compared via String compare, which is very time consuming, especially if long names are used that differ only in the last letters. To reduce CPU time, hashes are used now (instead of IDs in order to not break the existing interface).

See also:

[Node::GetRenderOrderBinAssignmentHash.](#)

Candera Engine 2D

Custom Effect Set

It is now possible to configure the list of Effects needed from an external source. For information on how to integrate this feature please refer to [Custom Effect Set](#).

New 2D Animation Property Setters

New animation property setters have been implemented for the Viewport ([Camera2D::GetViewport](#)) and ScissorRectangle ([Camera2D::GetScissorRectangle](#)) properties of a [Camera2D](#) object. The corresponding classes for these are [Animation::Camera2DViewportPropertySetter](#) and [Animation::Camera2DScissorRectanglePropertySetter](#).

New animation property setters have been implemented also for BoundingBox and ClippingRectangle properties of a [Node2D](#) object. The corresponding classes for these are [Animation::Node2DBoundingBoxPropertySetter](#) and [Animation::Node2DClippingRectanglePropertySetter](#).

A new animation property setter has been implemented also for the Layout [Margin](#) property of 2D nodes. The corresponding class for this is [Animation::LayoutMarginsPropertySetter](#).

New Math2D Functions

[Math2D::TransformScreenToWorld](#) accepts a camera object and a position in screen space. It returns the corresponding position in the world space.

[Math2D::TransformScreenToObject](#) accepts a camera object, a position in screen space and an object. It returns the corresponding position in the provided object's space.

Option to inherit the Layout Direction or to use the Culture

Previously the AutomaticDirection resulted in an inherited layout direction.

As a consequence the layout stayed fix at either left-to-right or right-to-left once a specific layout

direction has been chosen on a node for the complete sub tree.

Hence, two additional options for the layout direction have been introduced:

- InheritDirection: take the parent layout direction.
- CultureDirection: take the culture specific layout direction.

Now it is possible to change the setting again to the culture specific layout direction. For backward compatibility the AutomaticDirection will be by default interpreted as InheritDirection.

See also:

[2D Layout](#).

GridLayoutter effective Row/Column Data

Effective grid row/column data access interface has been added to [GridLayoutter](#).

See also:

[GridLayoutter::GetEffectiveRowData](#) and [GridLayoutter::GetEffectiveColumnData](#).

Candera Base

Dynamic Properties: Code Size Reduction

New macros have been introduced to reduce the code size by moving dynamic property code to implementation files. The new macros are used already in [Layouter](#) classes.

See also:

[CdaDynamicPropertiesDeclaration](#) and [CdaDynamicPropertiesDefinition](#).

Candera Scripting

Lua Update

Lua has been updated to version 5.3.4.

Candera System

Screenshot-Tool

[Candera](#) has a tool to create screenshots now. It is currently only available when 3D is enabled and OpenGL (ES) 3.0 is in use. The screenshot tool has an event listener which will be fired as soon as the screenshot is done. For more advanced usages on how to override the behavior, please have a look in the API.

See also:

[Diagnostics::ScreenshotTool](#).

SizeType Datatype

FeatStd::SizeType can now be used as widget property type for 32 and 64 bit targets.

Candera Device

RenderTarget events

RenderTarget now supports attaching listeners for catching events before and after SwapBuffers is called.

RenderDevice::ActivateShader deprecated

RenderDevice::ActivateShader has been deprecated. [Renderer::ActivateShader](#) shall be used instead.

TextBrush Line Spacing in Pixels/Factor of Height

A flag has been added to [TextBrush](#) for line spacing. When this is set to true, line spacing will be in pixels. When this is set to false, line spacing will be a factor by which

[TextRendering::Metrics::lineHeight](#) from [TextRendering::Style::GetMetrics\(\)](#) is multiplied.

See also:

[TextBrush::PixelwiseLineSpacing](#).

Candera Transitions

New Default Transition Type

The transition framework now supports triggering of animations that can be defined via [Transitions::Hint](#) parameters.

Reverse Transition

It is now possible to trigger reverse transitions. In case reversible fragments are identified, the already running transition fragments will go reverse, from their current state to their original one. An optional flag has been added to the [Transitions::Rule](#) to specify whether it shall be allowed to reverse a transition or not. Default value is false. A [Transitions::OnReverseTransitionFragment](#) corresponding event notifier has been added.

Candera Globalization

CultureChangeListener Callbacks

New methods have been added to [Globalization::CultureChangeListener](#) which are called before and after the culture has been changed.

See also:

[Globalization::CultureChangeListener::OnPreCultureChanged](#) and
[Globalization::CultureChangeListener::OnPostCultureChanged](#).

TextEngine

Language Dependent Fonts

Extended the [TextRendering::Style](#) to support language dependent fonts. A style can change the font to a culture dependent font when the culture has been changed. Further information can be found in the Changelog entry of SceneComposer and the [TextRendering::Style](#) documentation.

Alignment Behavior

The TextEngine provides a flag with

[TextRendering::LayoutingOptions::IsOrientationRealignmentEnabled](#) to switch between different solutions in case of a text which is too long for the maximum width. This flag decides which part of the text will be clipped in cases of centered or right-aligned texts. The flag provides two choices:

1. Show the text which is normally placed within the visual bounding box.
2. Show the beginning of the text within the visual bounding box providing the ability to read the beginning of the text.

The [TextNode2D](#) handles this by using the first solution when text trimming is disabled and the second one when it is enabled.

Minor Changes

Further optimized the visual output of justified text. And fixed some alignment issues.

Candera Widgets

More Property Types for Animations

It is now possible to animate widget properties of [Rectangle](#) and [Margin](#) types. Also, properties of [Vector4](#) type for widgets are now supported and can be animated. Access to the properties by specifying an index has been added.

Widget MetaInfo Improvements

- The [Widget](#) Macros no longer move everything after the macros into the public scope of a class (bug before the changes in [Candera](#)).
 - The internal CdaMetaInfo struct generated by the macros has been renamed to CMI (due to reduction in code size).
 - New method for Widgets: [WidgetBase::GetMetaInfo](#) (in macro CdaWidgetDefEnd).
 - Restructuring of the MetaInfo Macro implementation to reduce the code size dramatically.
-

Candera Monitor

Frame Debugger

Extended Monitor with the ability to capture frames and events.

Captured data includes frame timeline with important events (rendered nodes, camera, OpenGL calls) and color/depth/stencil buffer for selected nodes.

Candera V3.4.0.1

Release Info

- Release Date: June 22, 2017
 - For details please refer to the release document
-