Release notes: Ceetron Desktop Components [CDC]

Version CDC 3.11.0 - 17 Jun 2022 -

Bug

CAE-731 High memory usage in CeeGeometry

Fixed an issue where the Geometry Model pre-allocated a lot of memory which impacted the use of a lot of geometry models at the same time.

CAE-712 Isosurfaces computed from per element results does not have the correct scalar value (the iso value)

Iso surfaces now have the correct (constant iso) value if the result is per element and the same result is mapped on the the surface as the one being used to compute the surface.

CAE-756 VisualizationPartQuery: Crash when computing surface path on nodes connected to null-surface triangles

Small fix to VisualizationPartQuery to handle collapsed triangles.

Feature

CAE-754 Optimized cee::ug::UnstructGridModel::rayIntersect()

Optimized rayIntersect() by doing an early reject ray/sphere test before doing a ray/triangle test. This should give up to 3-4x performance gain and is especially useful on parts with a lot of visible element surfaces.

CAE-745 Modified the Qt Platform Integration version check

Modified some checks for the Qt Platform Integration source to work on Qt 5.14.

CAE-723 GeometryModel::regionIntersectPartItems() needs width and height > 0

Add documentation and a param assert on the width and height parameters to regionIntersectPartItems().

Version CDC 3.10.0 - 22 Mar 2022 - ed466b6f

Note: CDC 3.10.x requires **Ceetron Data Provider Framework (CDP) 7.x**

Bug

CAE-702 CDPElementGroup::addPolyhedronData() did not work as intended if called multiple times for one group

The data provider framework did not properly handle multiple calls to addPolyhedronData. If only called once per part (usual usage) it worked fine.

CAE-676 Particle traces produce extremely long lasting traces due to numerical precision

This impacted particle trace visualization styles that depend on time (comets and spheres) as well as animations. In some rare cases the particle traces would become extremely long lasting and thus taking an extremely long time. This also caused different behavior between C3 and CPM /CDC.

CAE-675 Particle traces does not contribute to bounding box

The particle traces was in some cases clipped by front/back clipping plane and also did not contribute to the bounding box used by fitView etc. This only occurred when the volume part was invisible, showing only traces without the part they were computed from.

CAE-659 setSkipOverlappingTickMarkLabels() does no work for horizontal legends

The new horizontal legends did not take the skipOverlappingTickMarkLabels setting into account.

Feature

CAE-687 Add poly-line segment indices and vertex indices to GeometryModel::regionIntersectPartItems

Improvements to `GeometryModel::regionIntersectPartItems`: Added `vertexIndexArr()` and `polylineSubIndexArr()` to `cee::geo::PartHitItems.` Vertex index will contain all vertices inside the given rectangle if acceptPartiallyContainedItems is true. `polylineSubIndexArr` will be populated for poly lines and contain the segment index of the given polyline (which index is in the itemIndexArr). See documentation for more info.

CAE-668 Add reader options to load either global or local mesh in OpenFOAM data provider

The OpenFOAM reader now has two new options: `LOAD_GLOBAL_MESH` and `LOAD_LOCAL_MESHES`. Both are set to default `true`.

CAE-663 ScalarSettings: Added setTickMarkLabelsMinimumDistance() to specify a minimal distance between tick mark labels

`ScalarSettings`: Added `setTickMarkLabelsMinimumDistance()` to specify a minimal distance between tick mark labels. Labels closer than this distance will be hidden.

CAE-647 Add support for providing error messages on init in CDP (Data Providers)

Added a new parameter to the `CDPDataProvider::init()` method:

```
init(const CDPString& modelKey, const CDPInitOptions& initOptions, CDPError* error) = 0;
```

The error objects allows for the data provider to specify an error message which is then accessible from the host.

`cee::ug::Error`: Added new error type `ERR_USER_DEFINED_MESSAGE` and `userDefinedErrorMessage`

Note: This bumps the version of CDP to 7.0 and requires data providers built with CDP 7.0 or later to work.

Version CDC 3.9.0 - 22 Feb 2022 - c3703e2b

Bug

CAE-633 MacOS library loader paths should not point to build server paths

The libraries now point to `@executable_path/../Frameworks/*.dylib`

CAE-632 GeometryModel::regionIntersectPartItems() did not work properly for partially intersected poly-lines

Partial selection of lines did not work as expected when using regionIntersectPartItems().

CAE-543 Scalar fringes causes a shader compiler error on OpenGLES 3.0

This problem occurs when using e.g. Angle with Qt and forcing OpenGL ES 3.0 to be able to do weighted average transparency.

CAE-534 Particle traces using spheres, comets and arrows do not work with clipping

Clipping was not properly implemented for particle traces drawn as spheres, comets and arrows.

CAE-533 Issues with Parallel VTU (pvtu) reader

Fixed a crash and results not being read correctly in some cases.

Feature

CAE-656 Update the readers in ImportCAE (VdmTools 4.6.1)

Added support for ANSYS 2022 and ABAQUS 2022. Several fixes for other formats.

CAE-582 ResultsQuery: ResultsQueryResultPosition now return ids (when exists) instead of indexes

The result from a `cee.ug.ResultsQuery` query (ResultsQueryResultPosition) now contains item ids if they exists, if not indices (before there was always indices).

CAE-553 Add Reference Type to StateInfo

The state reference type (TIME, FREQUENCY, LOADCASE, OTHER) can now be found in `StateInfo.referenceType()`

CAE-538 Rename plot axis (for FRF curves)

`cee.app.Plots` now renames the X axis based on the reference type (see CAE-553).

CAE-537 Add option to specify if particle traces should be clipped or not

Added support in CDC for specifying if particle traces should be clipped (view or cutting plane clipping) or not.
Added `ParticleTraceGroup::setIgnoreClipping()` Set this to true to always show the particle traces (no clipping). Default is to use clipping (ignore=false).

CAE-512 Plots: Add logarithmic scaling

Added X/Y logarithmic scaling settings to `cee.plt.AxisSettings`.

Version CDC 3.8.1 - 19 Jan 2022 - d2e28e7f

Bug

CAE-551 Text without background was in some cases not anti-aliased even if there was no transparent parts

---

Version CDC 3.8.0 - 17 Dec 2021 - 7a4ff130

Note: CDC 3.8.x requires **Ceetron Data Provider Framework (CDP) 6.x** and **Ceetron Result Calculator Framework (CRC) 4.x.**

Bug

CAE-496 Python: Adding BulkValuesArr and other missing Array classes

This fixes a crash with `BulkCalculation.getCutplaneValues()`

CAE-493 Particle traces fails to continue into a new part when using displacements

This fixes issues with particle traces on CFD models with moving parts using displacements.

CAE-487 Rare crash and holes for cutting planes computed from Polyhedron

On some very specific models computing a cutting plane would cause a crash on the server as well as producing some holes in polyhedrons and second-order elements.

CAE-218 cee::vis::MarkupPartLabel objects aren't displayed properly in front of transparent models

MarkupPartLabel and MarkupPartText3d parts without background was not rendered properly in front of transparent models.

Feature

CAE-513 DataProvider 6.0: Added support for specifying default part colors

You can now specify the default colors of parts from a data provider using the `CDPGeometryInfo::addElementGroupInfoWithColor()` or `CDPElementGroupInfo::setPartColor()`

CAE-505 Parallelization of computing node average and result calculator results

Results from Result Calculators are now computed in parallel as well as all node average computations. This should give a speed-up on multi-core systems.

CAE-497 Added CRCError parameter to CRCResultCalculator::initialize() to be able to communicate an error message to the host application

You can now provide an error message from result calculator initialize() method that can be picked up by the host application.

CAE-490 Disable auto generation of derived results from Ceetron Export

Added `Case::setDerivedResultsDisabled()` in Ceetron VTFx Export SDK.

CAE-478 Extend GeometryModel::regionIntersect() to optionally return contained indices within parts

Added `GeometryModel::regionIntersectPartItems()` which will return per part the indices of all part items that were intersected by the given screen-spaced region.

Version CDC 3.7.0 - 29 Oct 2021 - ab80041b

Note: CDC 3.7.x requires data providers built with Ceetron Data Provider Framework (CDP) version 5.x

Bug

CAE-448 RegionIntersect does not work when the the entire region is within an element

CAE-400 Legend tickmarks: wrong position when having undefined values

CAE-399 Add empty CDPResultValueGroup (no result for the part) was not correctly handled

CAE-397 NaN values are not correctly shown with undefined color

CAE-395 Outline edges are not correct when having overlapping shell and volume elements

CAE-379 cee::ug::OverlayColorLegend crashes if legend size is set too small

Feature

CAE-449 Added handleClientRequest() to CDPDataProvider

A CDC app can now communicate directly with the DataProvider (if used). Use the `cee::imp::cae::DataSourceCae::executeRequest()` to send a message to the data provider. The provider receives this in the handleClientRequest() and can report back a user-defined response as well as a change notification if any of the data it provides has changed.

This opens up for a wider use of data providers where the client can instruct the provider to modify the data and then see the results.

Note: CDC 3.7.x requires data providers built with Ceetron Data Provider Framework (CDP) version 5.x

CAE-396 Added cee::ug::ParticleTraceUtils::computeGridPoints()

CAE-387 Added support for horizontal color legends and center top/bottom positions

Legends can now be set to horizontal by the `cee::ug::ScalarSettings::setLegendDirection()` method.

Version CDC 3.6.0 - 09 Sep 2021 - 19295b98

Bug

CAE-175 setForceTickMarkLinesToCoverLegend(true) does not work when tick marks are left aligned

Feature

CAE-235 Optimized cee::exp::ExportConstantRemoteModel to use more optimal chunks

Added a better spatial partitioning algorithm for CUG models. This will result in a better LOD generation and a better front-to-back streaming of the ConstantRemoteModel.

CAE-204 Added cee::ug::VectorSettings::drawMaxCount to limit the number of vectors drawn

CAE-203 DemoApps: Fixed show/hide overlay items

CAE-180 Colorlegend: Create properties for force tickmark lines and tickmark alignment

CAE-177 Create CUG bundles using CPM and embed or static host CUG models in C3

Implemented support for creating CUG bundles using CeeExport module in CDC.
`ExportConstantRemoteModel` can now export to binary CUG bundles and Javascript CUG bundles, as well as the old CUG database format.

CUG bundles can be either downloaded as a static resource, opened as a local file or encoded as a JavaScript object. A CUG bundle is a single file, either containing binary data (`cugbundle.bin`) or encoded into a JavaScript object (`cugbundle.js`). If encoded as a JavaScript object, the model can be embedded into a single HTML page to create a standalone 3D result visualization without any server requirement

CAE-169 Improved performance when adding many states

CAE-163 Added settings for OverlayColorLegend Marker spacing and height

See `ug::ScalarSettings::setColorMarkerBoxHeight()` and `setColorMarkerVerticalMargin()` for more info.

CAE-143 C#: Add CeeApp component to distribution

CAE-136 Improve scalar mapping for color mappers with very uneven levels

Added a setting which enables a non-linear mapping of scalar result to the texture used for scalar mapping on the model (fringes). This setting will enable equal amount of texture pixels for each level. This will ensure that legends with very uneven levels (e.g. large range but then some very small level around key value ranges) gets a much better scalar mapping.

Use `ug::ScalarSettings::setForceUniformTextureLevels(true)` to enable this feature.

CAE-118 Find the specific key result values and its node and element ID

CAE-26 Result Calculators in CDC

Added support for result calculators in CDC. The Result Calculator Framework can be used to create result calculators as plugins (dll/so/dylib) that can be loaded into a CDC based application.

The Result Calculator Framework (CRC) has been refactored to support multiple calculator instances from a single calculator (dll/so/dylib).

Added methods for manipulating result calculators in DataSourceInterface. See documentation for more info.

Version CDC 3.5.0 - 17 Jun 2021

Bug

CAE-115 Isovolume issue in higher order elements and undefined midnodes

CAE-110 Issue with ExportDataSourceVTFX when model only have surface elements and the flag ExportSurfaceOnly is enabled

CAE-107 Error in log when using Dual Depth Peeling

CAE-54 Assert when changing drawstyle without calling NORMAL updateVisualization()

CAE-11 Report: Embedding of VTFx does not work on newer PowerPoint

Feature

CAE-137 Add setPriority() for overlay items and made sure overlay items are rendered above any model items (including use2dPixelProjection())

CAE-135 Debug libraries now have a 'd' postfix to separate from release versions

CAE-106 ImportCAE: Update to ANSYS, ABAQUS and HyperWorks support to 2021. Various other reader fixes

CAE-105 Add Qt 6 support

CAE-81 Added CuttingPlaneData::allOriginalTriangles() to get non-splitted triangles from cutting planes

CAE-15 Added Font::setLineSpacing() to change the default linespacing of a font

CAE-14 Plot: Add option for legend position

CAE-13 Plot: Add option to show/hide min/max values in axis titles

CAE-12 Improved visualization of undefined results

CAE-4 C#: Add missing classes and properties and sync with current CDC

CAE-2 Add global search capability to help pages