Release notes: CEETRON Envision

Latest online release notes: https://techsoft3d.atlassian.net/servicedesk/customer/portal/18/article/1978957825

CEETRON Envision for Web **CEETRON Envision for Desktop**

CEETRON Envision 1.2.2 - March 20, 2023 - SHA:

New version of Ceetron Access

Updated Ceetron Access (file readers) to patch version 1.0.7

CEETRON Envision 1.2.1 - 16 Feb 2023 - SHA: d026d7a8

Bug

CAE-1065 Can get null reference on mouse wheel while RemoteModel is loading

Using the mouse wheel early when the model is loading and having `mouseWheelZoomAndWalkBehavior` set to other than `TO_CENTER` might cause a throw and thus error logs in the console.

CAE-1067 ImportCAE OpenFOAM reader crashes on models with polyhedron elements

The built-in OpenFOAM reader in ImportCAE might cause a crash when opening models with polyhedron elements. This fixes a regression introduced in Envision 1.1

CAE-1071 VTFx component: Exported file can't be opened in the viewer (wrong vendor id)

Files exported with the VTFx components (standalone export) was not signed properly and thus not able to open in the free desktop 3D viewer.

CAE-1072 Simple null vectors would sometimes not be positioned correctly on some hardware platforms

In some rare cases on some hardware platforms the null vectors could be drawn in the wrong position.

CEETRON Envision 1.2.0 - 27 Jan 2023 - SHA: 18704d8b

Bug

CAE-920 UgModule is missing some exports

DisplacementSettingsProperties, ScalarSettingsProperties, VectorSettingsProperties were missing from the UgModule TypeScript d.ts file.

CAE-921 UgServer is sometimes confused about the current state id after applying a VTFx case

When applying cases from VTFx files, in some cases the current state was not properly synced between client and server, needing an extra update to get in sync.

CAE-925 PVD/VTM/VTU: Displacement vector fields are not interpreted as displacements result unless their name starts with "displacement"

The VTK reader now will use any result with the word “displacement” as part of the result name as a displacement result. See CAE-971 for an option to treat all vector results in VTK files as displacements.

CAE-947 UgServer: Memory is not freed by the c++ heap manager on Linux

Added option to trim heap memory usage on Linux/gcc using `malloc_trim()`. The feature can be enabled by setting the environment variable `CEW_UG_ENABLE_MEMORY_TRIM`

This functionality was added after getting reports of growing memory usage (and eventually crash) when running Linux version of ugServer in a Docker container. When enabled, the feature performs regular calls to `malloc_trim()` to manually do "garbage collection" on the c/c++ heap manager. Note that doing this could probably has a negative impact on performance.

CAE-962 Updating a cug model + removing another causes assert error

Fix assert error that occurs on update a CUG model then remove another.

CAE-964 Asset error on update cug part then call getBoundingBoxForPart()

Fix assert error that occurs on update CUG model then call getBoundingBoxForPart()

CAE-984 CugComposer crashes on VTFx files with IndexedFaceSets

The CugComposer did not handle (legacy) VTFx files with Indexed Face Sets

CAE-1019 Reference counting did not work properly from Python

Reference counting is now handled properly in the Python bindings for Envision for Python in the same way as for C# .

CAE-1039 ExportVTFx uses wrong vendor code, can't open in Viewer

The VTFx files exported from Envision for Desktop is now properly signed and can be viewed in the CEETRON 3D Viewer.

CAE-1049 CugComposer cannot pass unicode characters in arguments paths

The CugComposer now handles unicode filenames, both for input VTFx files and output folder/files.

Feature

CAE-782 Camera changed callback

Added `cee.Camera.setViewChangeHandler()` that sets a handler to be invoked each time the camera view changes.

CAE-802 Enable/disable mirroring per part

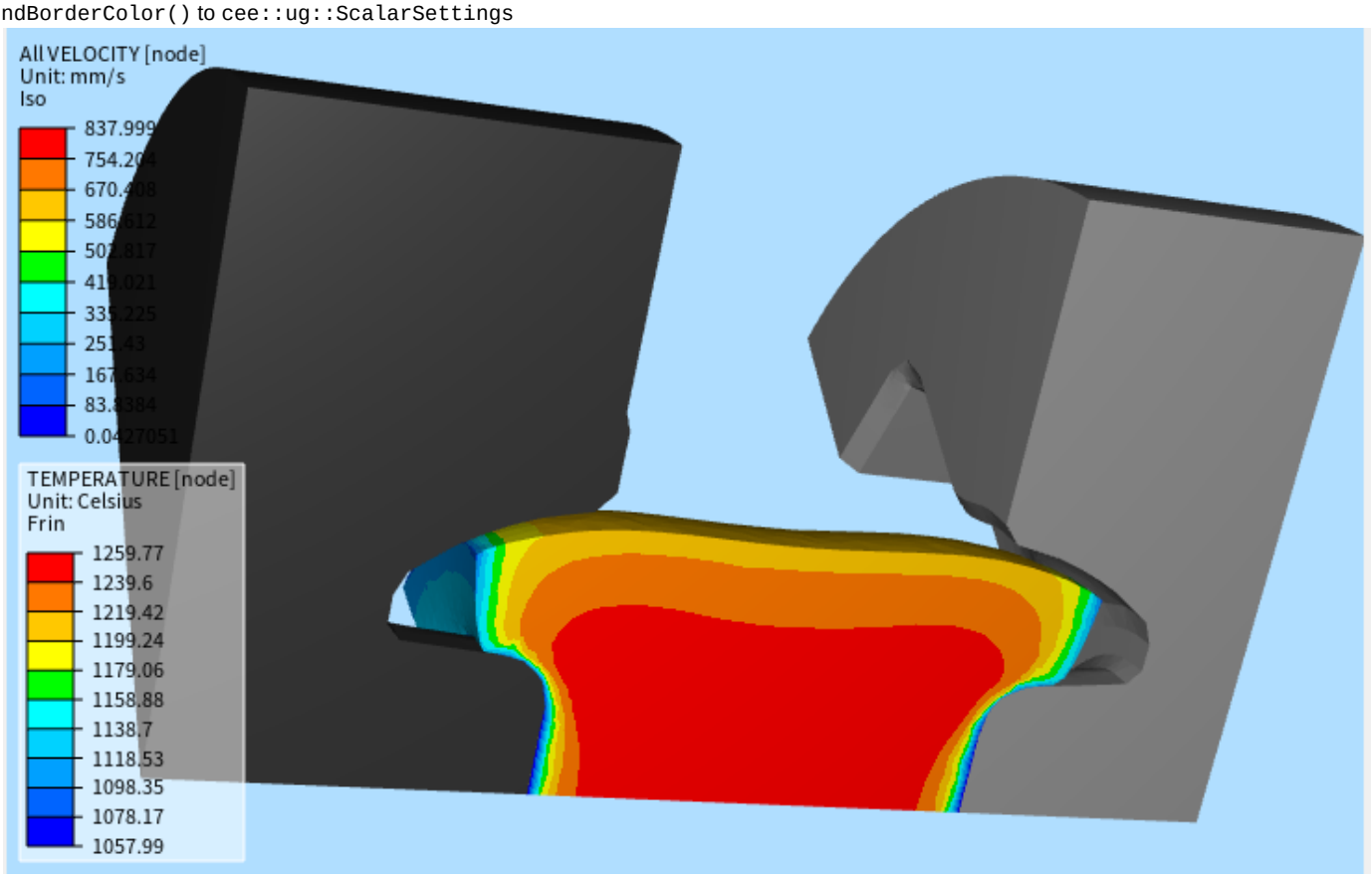
Added an option in UnstructGrid PartSettings to control if a part should be mirrored or not.

Added `cee::ug::PartSettingsdisableMirroring()` method (desktop) and `cee.ug.PartSettings.disableMirroring` property (Web).

CAE-810 Specify a background color for color legends

Desktop: Added support for a background for color legends. This helps making text more readable in some situations. The background color and opacity, as well as border color can be specified.

Added `setDrawLegendBackground()`, `setLegendBackgroundColor()`, `setLegendBackgroundOpacity()` and `setLegendBackgroundBorderColor()` to `cee::ug::ScalarSettings`



CAE-882 Vectors fields visualization: optionally display a vector a a point when length is 0

Desktop: Added an option to draw vectors with zero length (null vectors) as a point or a sphere.

CAE-918 Add visible property to cug.ParticleTraceGroup

The visibility of particle trace groups can now be controlled in the ConstrantRemoteModel.

Added `cee.cug.ParticleTraceGroup.visbile` property.

CAE-919 Add method for getting BoundingBox for single part in ConstantRemoteModel

You can now get the bounding box of a part in the ConstantRemoteModel.

Added `cee.cug.ConstantRemoteModel.getBoundingBoxForPart()`

CAE-926 Add support for computing fit view eye position from a bounding box

Added a new method to Camera to compute the fitView eye position based on a given bounding box.

Added `cee.Camera.computeFitViewEyePosition()`

CAE-931 Web: Query to get feature extraction data

Added a new query to get the display model of a feature extraction item (cutting plane, isosurface or isovolume). Triangle and lines with the corresponding per vertex scalar result is provided as a result of the query. This is useful for client-side post processing of the feature extraction, as well as to visualize it in various ways.

Added `cee.ug.QueryFeatureExtraction`

CAE-949 Improve "Assertion failed" reporting mechanism

Added support for specifying an assertion failed handler that is invoked if a code assert fails on the client.

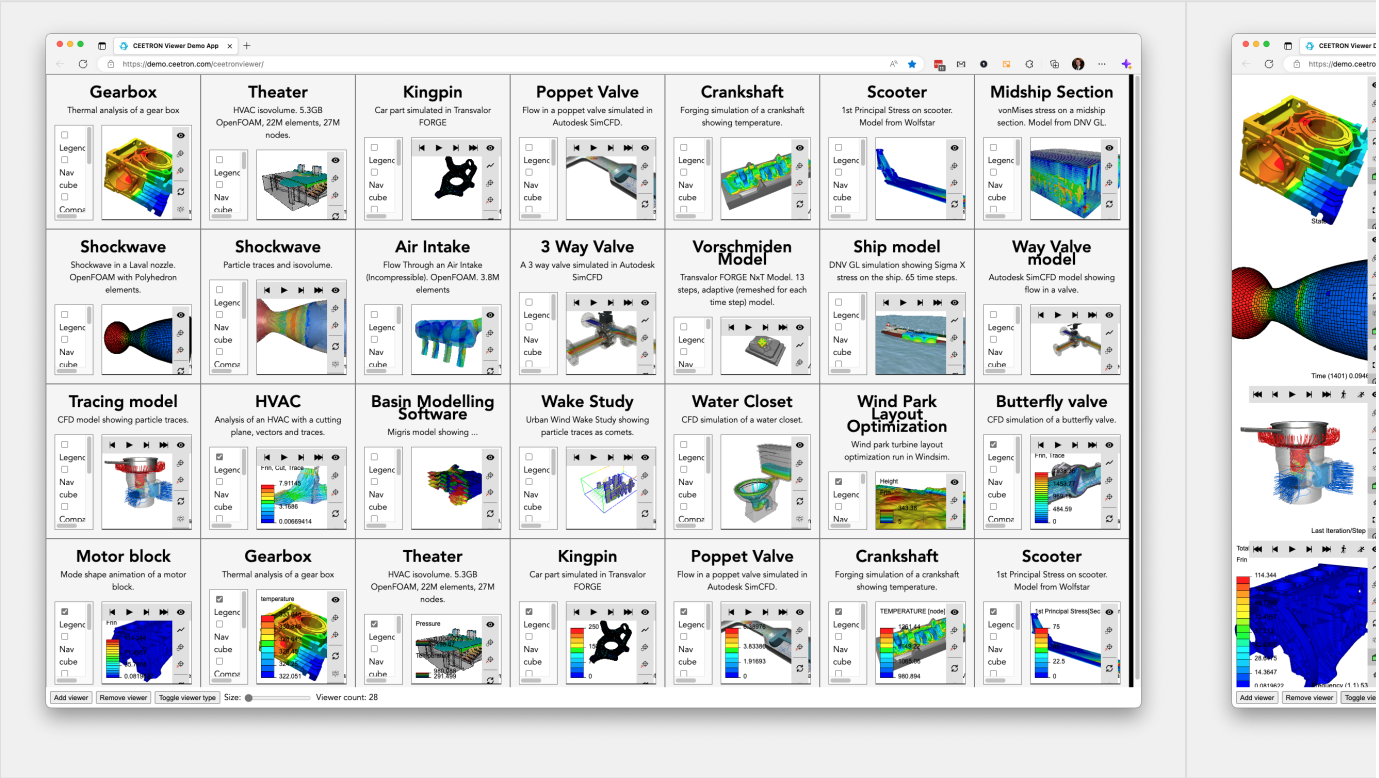
Added `cee.setAssertFailedHandler()`.

CAE-953 Add support for custom view layouts

Added support for specifying a custom layout of views within a viewer. This enables using many views in one viewer, and thus working around the limit of only 16 WebGL canvases on one page.

Added `cee.View.setViewport()`

Here is an example with 28 views with different models on one page (within one viewer):



CAE-971 VTK: Reader option to provide all vector results as displacements

Added an option for the VTK (vtu/pvtu) reader to treat all results as displacements, and thus making them available as displacements, vectors and individual scalar results.

Added `cee::imp::cae::ReaderSettingsVtk& ReaderSettings::vtk()` with `setReadVectorsAsDisplacements()`.

Added VTK setting `ALL_VECTORS_AS_DISPLACEMENT` to be used in the `cee.ug.RemoteModel.setReaderOptions()` method.

Example:

```
this.m_model.setReaderOptions("VTK", {
  "ALL_VECTORS_AS_DISPLACEMENT": true,
});
```

CAE-988 Add support for supplying license code as environment variable to CugComposer

Added use of the environment variable `CEW_TECH_SOFT_LICENSE_CODE` in the CugComposer and the UgServer. There are now 4 ways to provide a license to the CugComposer:

1. `-license` command line option
2. `CEW_TECH_SOFT_LICENSE_CODE` environment variable
3. `-licenseFile` command line option
4. `CEW_TECH_SOFT_LICENSE_FILE` environment variable

CAE-1008 Added option to disable spatial partitioning when optimizing parts in UnstructGridModel

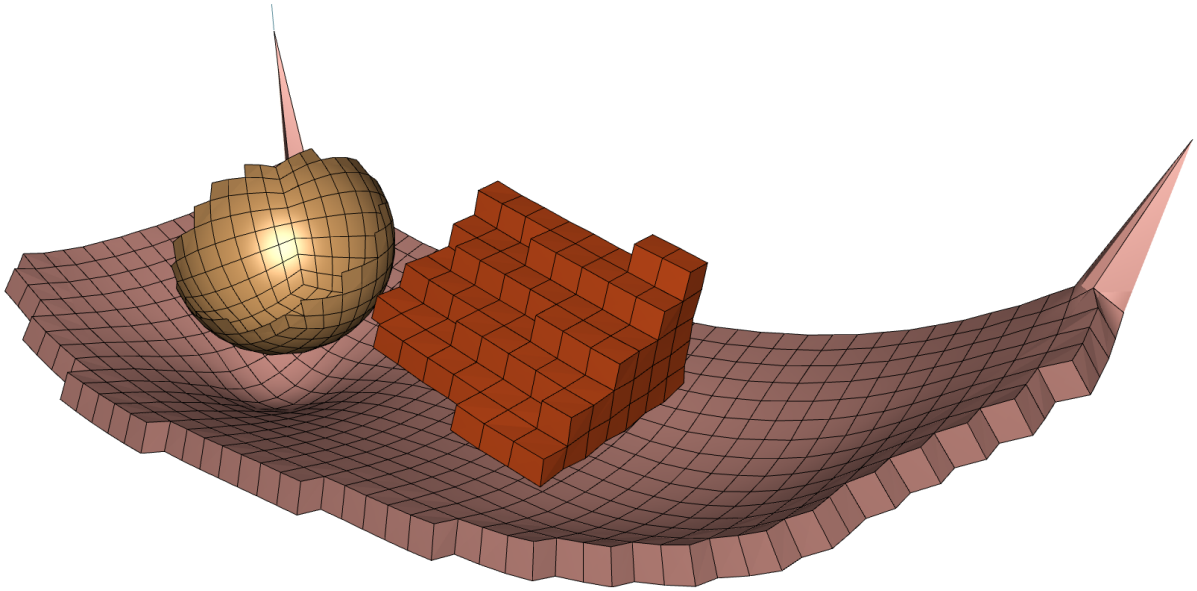
Added an option to specify only to merge equal parts based on appearance and not do spatial partitioning into equal sized chunks.

Added new parameter to `cee.ug.ModelSettings.setOptimizePartRendering(bool optimize, bool spatiallyPartitionParts = false)`

Default is now to only merge parts with equal appearance.

CAE-1035 Added support for CrincleCut cutting planes of 2d elements

Crinkle-cut cutting planes now also work as expected for 2d elements, showing the entire 2d element if intersected by the cutting plane.



CAE-1046 CugServer: Support absolute paths for CONFIGURE_EXPRESS_APP_JS_FILE

The app config file for the CugServer can now be specified with an absolute path.

CEETRON Envision 1.1.0 - 25 Nov 2022 - SHA: 68b113ab

Bug

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

Fixed a crash when trying to do a surface path query starting on a beam or a point element.

CAE-852 Default value for MODEL_PATH in server/UgServer/Main.js is wrong

Fixed the model path based on the package reorganization for Envision 1.0

CAE-853 CugServer build fails for typescript 3.9

Fixed so you can compile the CugServer with TypeScript 3.9

CAE-868 Web: BuildYourFirstApp/2-Viewer: The initial camera is not set as intended

The callback to the openModel() was not correct in the example so setting the initial camera did not work.

CAE-878 ug.ScalarSettings.scalingConstantTerm and scalingFirstDegreeTerm not applied to legends

The use of unit conversion (setting a linear transformation of the results on the legend) did not always work as intended.

CAE-880 Assertion failed if using continuous legend bar and setting same min and max range values

When using a continuous legend mapping (numLevels = 0) the client would assert if min was set to the same as max.

CAE-892 HitItem returns undefined intersectionPointResult value for cutting plane when having parts without results

The intersectionPointResult value in cee.ug.HitItem was undefined or wrong if a model had some parts without any results defined.

CAE-902 CugServer: CEW_CUGS_CONFIGURE_EXPRESS_APP_JS_FILE should be relative to current dir rather than Main.js

Changed the behavior of the CEW_CUGS_CONFIGURE_EXPRESS_APP_JS_FILE environment.

Feature

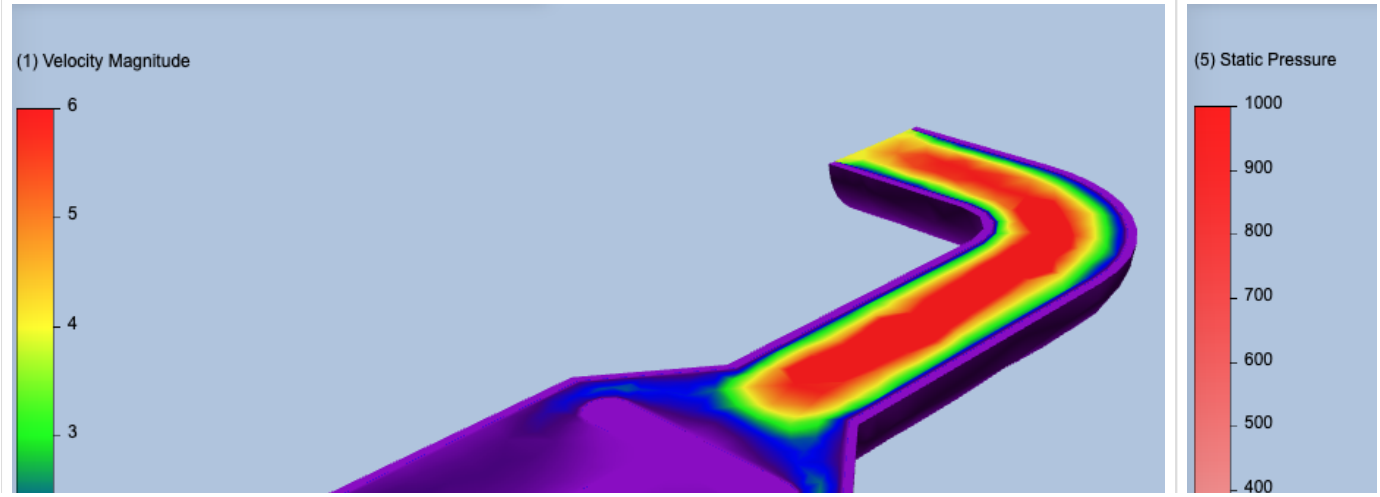
CAE-848 cee.ug.ScalarSettings: Custom continuous color scheme support

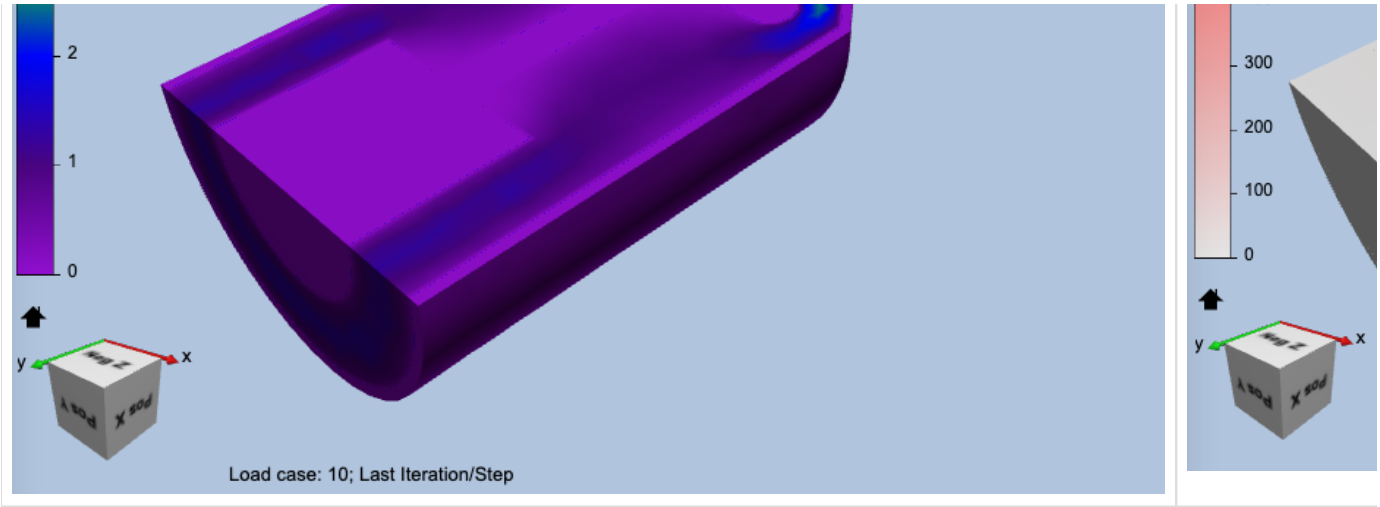
Added support for specifying a custom color mapping for continuous color legends. You can provide an array of colors and the legend will then be interpolated between these colors. If values are provided the legend will have the given color at the given value. If no values are provided the colors will be evenly spaced.

```
scalarSettings.setCustomContinuousColorArr([
        {r: 148/255, g:   0/255, b: 211/255},
        {r:  75/255, g:   0/255, b: 130/255},
        {r:   0/255, g:   0/255, b: 255/255},
        {r:   0/255, g: 255/255, b:   0/255},
        {r: 255/255, g: 255/255, b:   0/255},
        {r: 255/255, g: 127/255, b:   0/255},
        {r: 255/255, g:   0/255, b:   0/255},
    ], [0,1,2,3,4,5,6]);
```

```
scalar
    [


    ],
```





See `cee.ug.ScalarSettings.setCustomContinuousColorArr()` for more info.

Note: Setting `cee.ug.ScalarSettings..colorScheme = cee.ug.ColorScheme.CUSTOM` was never intended to be a valid operation and this is now documented and the property will throw if this is tried.

CAE-867 Element edge, face and node DataSets

Added support for element edge, element face and element node sets in Envision Web and VTFx.

CeeVTFx:

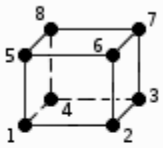
- Added ELEMENT_FACE, ELEMENT_EDGE, ELEMENT_NODE and NODE as supported item types in `cee::vtfx::SetBlock`.
- Added itemSubIndex to `cee::vtfx::SetItemGroup` in order to specify a local face, edge, node index for these new set types.
- Added new example VTFxSets which showcases how to use sets in VTFx.

Envision Web: (cee.ug)

- Added `elementFaceIndex`, `elementEdgeIndex` and `elementNodeIndex` to `cee.ug.HitItem`.
- Added `itemType` to `cee.ug.SetInfo`
- Added new query `QuerySetItems` to get items in a given set. This has an option to also return the tesselation of the set items (triangles for element face sets, lines for element edge sets and points for element node/node sets). This is useful for showing the sets by feeding these tessellations into the Markup or Geometry model.

Updated documentation to show element edge and surface order for Envision Desktop and VTFx:

enumerator **HEXAHEDRONS**



Hexahedron volume element with 8 nodes.

edge	nodes	edge	nodes	face	nodes
1	2, 6	7	6, 7	1	1, 4, 3, 2
2	1, 2	8	5, 6	2	5, 6, 7, 8
3	1, 5	9	5, 8	3	1, 2, 6, 5
4	1, 4	10	4, 8	4	2, 3, 7, 6
5	2, 3	11	7, 8	5	3, 4, 8, 7
6	3, 7	12	3, 4	6	1, 5, 8, 4

Note: Full support in Envision Desktop is coming in a later version, as well as support in Data Providers.

CAE-870 Add cug-server to examples docker installation

The Envision distribution demo (http://envision.ceetron.com) now has a CugServer running to the CUG examples should work as intended.

CAE-876 cee.ug.CuttingPlane: Eye-lift support for cutting planes

Added a setting for giving cutting planes a visual priority (eye-lift) to control z-fighting when the cutting plane is competing against other geometry.

See `cee.ug.CuttingPlane.eyeLift` for more info.

CAE-879 Added Data Provider Framework documentation to the Envision Desktop documentation

The documentation for the Data Provider Framework is now a part of the CEETRON Envision Documentation.

CAE-881 The web ugServer is now built with CentOS7 and only requires glibc 2.17

The server for RemoteModel (server/UgServer) is now built with CentOS7 and requires only glibc 2.17. The previous requirement was Ubuntu16 (glibc 2.23).

CAE-885 Add support for serving web apps through CUG server

Add support to CUG server for serving web apps over same port

- Add CEW_CUGS_PUBLIC_DIR env var for specifying dir to be served statically
- Add CEW_CUGS_CONFIGURE_EXPRESS_APP_JS_FILE env var for providing extra routes required by the app
- Move status route to /status
- Rename docker container models dir from "TestModels" to more generic name "Models"

CAE-888 Add DataProviderFramework/ResultCalculatorFramework to Envision Desktop distribution

Added the Data Provider Framework and the Result Calculator Framework distribution to Envision Desktop.

CAE-895 Add interface name to SimulationInfo

Added name of interface (file reader / data provider) currently used by RemoteModel to SimulationInfo (e.g. NASTRAN, ABAQUS, OpenFOAM, etc)

See `cee.ug.SimulationInfo.interfaceName` for more info.

CAE-904 cee.ug.CuttingPlane: Support for turning the light on/off

Add support for disabling lighting of cutting planes. This can be useful to see the colors on the cutting plane without shading effects.

See `cee.ug.CuttingPlane.lighting` for more info.

CAE-907 Add OpenFOAM data provider to Envision Desktop distribution

The OpenFOAM data provider is now distributed as a part of CEETRON Envision for Desktop.

CAE-910 Upgrade Envision Desktop and Web to use CEETRON SAM 1.0.1

CEETRON Envision 1.0.0 - 28 Sep 2022 - SHA: 39e4a5d5

Note: CEETRON Envision 1.0 requires a new license and a new way to provide that license to the toolkit. Please see the license section below.

Rename Ceetron Cloud Components (C3) to CEETRON Envision for Web

The product Ceetron Cloud Components (C3) has been renamed to CEETRON Envision for Web. The product is almost identical and has the same features, with the change of the license system being the most notable change (see below). There are also some changes to the

organization of the distribution archive, where the servers are now grouped into the server folder, and the CeeCloudServer is now in the `server/ugServer` folder.

The product documentation is now also part of the Tech Soft 3D documentation system and can be found **here**.

Note: The demo site running the docker versions of Envision Web can now be found on: https://envision.ceetron.com (previously http://c3.ceetron.com).

Rename Ceetron Desktop Components to CEETRON Envision for Desktop

The product Ceetron Desktop Components (CDC) has been renamed to CEETRON Envision for Desktop. The product is almost identical and has the same features, with the change of the license system being the most notable change. (see below). Ceetron Python Modules [CPM] is now a part of the Envision Desktop distribution and no longer a separate product. See **CAE-850** below for more info.

The product documentation is now also part of the Tech Soft 3D documentation system and can be found **here**.

New unified documentation for all CEETRON products (including Envision Desktop and Web)

We have a new look and feel for the entire CAE product line at Tech Soft 3D. This is using the same system as for the new HOOPS documentation.

The main entry point is https://docs-test.techsoft3d.com/ceetron/latest/main/index.html. Here you can find an overview of the entire CEETRON product line and browse or search for documentation for any of the products.

New unified HOOPS license system in Envision (Web and Desktop)

Envision (Web and Desktop) now uses the unified HOOPS license. The same license can be used across all CEETRON and HOOPS products, given that the license has the needed features enabled. This license key is a long string and it replaces the two keys (KeyA and KeyB) that were used in C3 and CDC.

For **CEETRON Envision Web**, the license can be specified with the `ugServer.setLicenseCode(myCode)`. You can also copy your `hoops_license.h` (downloaded from the Developer Zone) into the same folder as the server code. For more help, please see the documentation **here.**

For **CEETRON Envision Desktop**, you need to provide the string to `cee::CoreComponent::initialize()`. One way to do this is to use `hoops_license.h` (downloaded from the Developer Zone) as shown below:

```
#include "hoops_license.h"
    ...
    // Initialize the parts of CEETRON Envision Desktop that we'll use.
    g_componentInstance = cee::CoreComponent::initialize(HOOPS_LICENSE);
```

For more information, please see the documentation **here**.

Changes to versioning

CEETRON Envision Desktop and Web will follow the same version numbers and starts at 1.0. We will strictly follow Semantic Versioning (https://semver.org/).

For Envision Desktop there have been some changes to the `CeeCore/Version.h` file with its macros:

Old Define	New Define
`CEE_VERSION`	`CEE_VERSION_NUMERIC`
`CEE_MAJOR_VERSION`	`CEE_VERSION_MAJOR`
`CEE_MINOR_VERSION`	`CEE_VERSION_MINOR`
`CEE_PATCH_VERSION`	`CEE_VERSION_PATCH`
`CEE_SPECIAL_BUILD`	`CEE_VERSION_PRE_RELEASE`

Note: The `CEE_VERSION_STRING` is now: `"MAJOR.MINOR.PATCH"` (for pre-releases: `"MAJOR.MINOR.PATCH-PRE_RELEASE"`). Examples: `1.2.3` or `1.4.0-RC1`

The is a new method in CoreComponent to get the full version with the build SHA:

`CoreComponent::versionStringWithBuildSha()` Returns full version number. Examples: `1.2.1+e73b7a84` or `1.2.0-rc1+e73b7a84`.

There is now a `VersionInfo.json` file in the root folder of both Desktop and Web distributions detailing the version info of the distribution:

Web

```
{
    "productName": "Ceetron
Envision Web",
    "major": 1,
    "minor": 0,
    "patch": 0,
    "preRelease": "",
    "buildSha": "39e4a5d5",
    "version": "1.0.0",
    "versionWithBuildSha":
"1.0.0+39e4a5d5",
    "cugDataFormatVersion": 2
}
```

Desktop

```
{
    "productName": "Ceetron
Envision Desktop",
    "major": 1,
    "minor": 0,
    "patch": 0,
    "preRelease": "",
    "buildSha": "39e4a5d5",
    "version": "1.0.0",
    "versionWithBuildSha":
"1.0.0+39e4a5d5",
    "dataProviderFramework":
"7.0",

"resultCalculatorFramework":
"4.0"
}
```

Changes to environment variables

Most environment variables have changed from C3_* to CEW_*.

Old environment variable name	New environment variable name
C3_UG_PORT	CEW_UG_PORT
C3_UG_USE_HTTPS	CEW_UG_USE_HTTPS
C3_UG_MODEL_PATH	CEW_UG_MODEL_PATH
C3_UG_ENABLE_REST_API_UPLOAD	CEW_UG_ENABLE_REST_API_UPLOAD
C3_UG_ENABLE_REST_API_QUERY	CEW_UG_ENABLE_REST_API_QUERY
C3_UG_ENABLE_CORS	CEW_UG_ENABLE_CORS
C3_UG_DATA_PROVIDER_FOLDER	CEW_UG_DATA_PROVIDER_FOLDER
C3_UG_RESULT_CALCULATOR_FOLDER	CEW_UG_RESULT_CALCULATOR_FOLDER
C3_UG_ADD_ON_FULL_FILENAME	CEW_UG_ADD_ON_FULL_FILENAME
C3_UG_DISABLE_IMPORT_CAE	CEW_UG_DISABLE_IMPORT_CAE
C3_UG_LOG_CONSOLE_OUTPUT_TYPE	CEW_UG_LOG_CONSOLE_OUTPUT_TYPE
C3_UG_LOG_FILE_OUTPUT_TYPE	CEW_UG_LOG_FILE_OUTPUT_TYPE
C3_UG_LOG_FILENAME	CEW_UG_LOG_FILENAME
C3_UG_LOG_LEVEL	CEW_UG_LOG_LEVEL
C3_CUGS_PORT	CEW_CUGS_PORT
C3_CUGS_FILE_DS_PATH	CEW_CUGS_FILE_DS_PATH
C3_CUGS_LOG_CONSOLE_OUTPUT_TYPE	CEW_CUGS_LOG_CONSOLE_OUTPUT_TYPE

C3_CUGS_LOG_FILE_OUTPUT_TYPE	CEW_CUGS_LOG_FILE_OUTPUT_TYPE
C3_CUGS_LOG_FILENAME	CEW_CUGS_LOG_FILENAME
C3_CUGS_LOG_LEVEL	CEW_CUGS_LOG_LEVEL
C3_GS_PORT	CEW_GS_PORT
C3_GS_USE_HTTPS	CEW_GS_USE_HTTPS
C3_GS_REDIS_DS_HOST	CEW_GS_REDIS_DS_HOST
C3_GS_FILE_DS_PATH	CEW_GS_FILE_DS_PATH
C3_GS_LOCAL_FILE_SOURCE_PATH	CEW_GS_LOCAL_FILE_SOURCE_PATH

Bug

CAE-831 Missing vector filtering "skip by" when mapped to iso/cut

Specifying a skip-by for drawing of vectors did not work as expected for isosurfaces, isovolumes and cutting planes.

CAE-826 Query of results from VTFx files which reference result values by ID did not work

Querying results from a model backed by a VTFx file did not work as expected if the result values were referenced by ID (not the recommended way to reference result values).

CAE-811 [Web] ug.QueryNodeInfo returned -1 for node id if displacements were enabled

If a model were using displacements, the result returned from QueryNodeInfo would not contain the correct node ids.

CAE-801 [Web] VTFx properties did not apply info on visible sets and how to show them properly

Loading VTFx models with properties did not apply proper settings for visible sets.

CAE-762 Wrong visualization of mirrored parts on a cutting plane

Cutting planes for mirrored parts did not render correctly.

Feature

CAE-850 Added support for Python 3.7-3.10 in one unified distribution

CEETRON Envision Desktop now supports Python 3.7-3.10. In the distribution archive, there is a `Python` folder that contains what used to be Ceetron Python Modules [CPM].

Note that we have changed the python folder from `cee` to `cee_envision` to prepare for distribution on PyPI. So all old scripts using CPM needs to be updated to reflect this change.

CAE-774 [Web] Add support for dynamic symmetric arrows for vectors

Added support for drawing symmetric vector arrows which direction (inward or outward) is controlled by the sign of a user defined scalar result. This could be used to show if a stress result is compression or tension.

Added a new vector type (`DYNAMIC_SYMMETRIC_ARROW`) and `VectorSettings.setVectorTypeDynamicSymmetricArrow(scalarId: number)` to specify this new visualization.

Note: The property `VectorSettings.vectorType` now is a read-only property and the set property is deprecated. It will still work, but will be removed in future versions. So please use the new `setVectorTypeArrow()`, `setVectorTypeSymmetricArrow()` and `setVectorTypeReverseSymmetricArrow()` functions in `VectorSettings` to change the vector type.

CAE-698 [Web] Client-side element picking

Added support for client side picking of visible triangles within a region (rectangle) or along a path (set of points). This is a fast way to find all visible triangles within a region. You can then use the output of this to highlight the triangles, or to query the server to get which elements these triangles represent.

Added `RemoteModelElementPicker` which can be produced by `RemoteModel.createElementPicker(view: View)`. Use the `getVisibleElementsInRectangle` or `getVisibleElementsAtPoints` methods to perform the query. Note that a new `RemoteModelElementPicker` must be created if you change the state of the RemoteModel. See documentation for more info.

CAE-827 [Web] Add interfaceName to cee.ug.SimulationInfo

Added the name of the interface/reader/data provider that is currently in use for the RemoteModel in `cee.ug.SimulationInfo.interfaceName`

CAE-760 [Web] Create new markup part for lines

Added new Markup Part `cee.mrk.PartLines`. This is a part that can show lines in a MarkupModel. Added cee.utils.`PathGenerator` which is a helper class to generate vertices for a path that can then be shown with the PartLines markup part.

CAE-740 Support for polyhedrons in CEETRON Envision VTFx component

Polyhedrons are now fully supported in the VTFx export component. Added `cee::vtfx::ElementBlock::addPolyhedronElements()` for adding polyhedron elements to a VTFx block.

CAE-794 Add result unit information to VTFx export and Envision Desktop

Added `cee::ug::ResultInfo::setUnit()` and `unit()` to Envision Desktop. This is already present in Envision Web.

Added `cee::vtfx::ResultBlock::setUnit()` and `unit()` to CEETRON Envision VTFx component.

CAE-836 Add robustness for empty DataPartVector/DataPartScalar

CEETRON Envision now handles parts with no result a bit more robust, as it is now allowed to have part objects with no results and not only `null ptr` in the part array.

For previous releases, please see Release notes: Ceetron Desktop Components [CDC] or Release notes: Ceetron Cloud Components [C3]