



1.2.2

US 623 Added experimental support for a single step mode in CeeCloudServer.

By specifying the "singleStepMode" configuration flag to RemoteModel.openModel(), the server will only load a single step at a time, thus greatly reducing the memory resource usage of animations on the server. Note that this might slow down the server.

US 622 Added support for reading OpenFOAM (*.foam) files. Note that polyhedra elements is still not supported.

US 621 Added support for copy on open in CeeCloudServer (using the "copyOnOpen" config flag passed to RemoteModel.openModel()).

US 620 Added support for reloading an analysis in the RemoteModel.

RemoteModel.reload() will reload the analysis files on the server and update the client with the changed data. The ModelDirectory will be updated with any new states and result variables.

US 619 The CeeCloudServer is now more proactive in closing any open VTFX files and releasing native session memory whenever a client explicitly closes a RemoteModel or is disconnected from the server. Previous versions relied on Node.js' garbage collection in order to close files and release native memory.

US 617 Added server side cleaning of part piece cache. Clear out data payload of normal part pieces after they have been sent (but retain the records to keep CRC working). Completely prune volatile part pieces from the server cache once they have been sent.

US 609 Now allowing non-power-of-two (NPOT) textures if the texture options comply with WebGL requirements: No mipmaps or tiling.

US 579 Added region selection of parts in the Geometry Model.

See geo.GeometryModel.regionIntersect() for more information.

1.2.1

US 608 Added helper class for producing color tables (cee.ColorTableFactory). Useful for scalar mapping in the geometry model, and to produce part colors (as in ug and C3DC).

US 607 Added PartPoints to the MarkupModel.

Use MarkupModel.addPointsPart to create a new instance of this class.

US 606 Added a technology preview version of a new geometry server (CeeCloudGeoServer).

The GeometryServer is a high performance stateless server implemented in TypeScript. It has a very low resource overhead on the server and support fast loading of remote geometry models.

Note: This is a tech. preview version and is not production ready.

US 605 Big performance increase when partially updating the geometry model (cee.geo.GeometryModel).

Updates when only some of the parts are changed is now a lot faster.

US 604 Added support for showing node averaged scalar results. New property: ScalarSettings.nodeAveragedValues.

US 603 Draw divider lines between viewports when showing multiple views in a viewer.

US 602 Added GeometryModel.deletePartsAt()