Coreform IGA for Abaqus 2026.8

Table of contents

Coreform IGA for Abaqus 2026.8 introduces a redesigned Python interface and Abaqus/CAE application, experimental local refinement, more robust CAD trimming and boundary processing, and faster Coreform IGA Mesh and Coreform IGA Interop workflows. This release also improves implicit-dynamics and nearly incompressible analysis workflows, simplifies installation, and expands job automation.

> **i** Note
>
> Capabilities marked **Production**, **Beta**, or **Alpha** below use the support levels defined in Current capabilities and limitations. **Experimental** identifies an early workflow with the specific restrictions stated in these notes and its linked documentation. The Capabilities matrix provides the detailed scope, applicable versions, limitations, and verification evidence for each capability.

1 Abaqus integration and automation

1.1 Redesigned Python interface

The Coreform IGA scripting interface has been redesigned to be more consistent with other Abaqus scripting interfaces. This is now available in the `coreform` and `coreform.abaqus` Python packages. Portable specifications provide a consistent way to describe probes, IGA meshes, jobs, and their execution settings, while Abaqus-aware records connect those definitions to models, parts, and assembly instances.

The new interface supports creating, editing, copying, renaming, and deleting Coreform definitions. Job automation now covers IGA mesh generation, Abaqus input-file creation, submission, status monitoring, termination, and opening completed results.

1.2 Rebuilt Abaqus/CAE application

The Coreform IGA for Abaqus user interface has been rebuilt on the same model records used by the Python interface. Probe, mesh, and job managers remain synchronized with their saved model data, including changes to instance dependency, suppression, and ownership. Mesh-manager refreshes are also more responsive and no longer become progressively slower as a session is used.

Coreform IGA definitions are recorded alongside native Abaqus operations in recovery and journal files. Replaying a journal reproduces Coreform creations, edits, and deletions, providing a reliable starting point for repeatable model-authoring workflows.

1.3 Persistent Coreform model data

Probe, mesh, and job definitions created with this release are stored with the Abaqus model database. Opening and saving that model database in plain Abaqus/CAE preserves the Coreform records so they are available when the database is reopened in Coreform IGA for Abaqus.

2 Meshing and geometry

2.1 Experimental local refinement

Coreform IGA Mesh introduces **Experimental** local refinement for immersed rectilinear meshes. The `refinecf` utility can select regions using a named CAD surface, a sphere, or an axis-aligned bounding box, then apply multiple levels of hierarchical refinement with optional 2:1 balancing. This concentrates resolution near features of interest without refining the entire background mesh.

Local refinement is currently a command-line workflow and the meshing step must be run serially. See Local refinement toward a feature for a complete workflow and current limitations.

2.2 More robust CAD trimming and boundary tessellation

The trimming pipeline has been expanded for higher-genus and multi-component solid geometry, including parts containing through-holes or disconnected regions. The robustness of the trimming pipeline has also been improved through expanded testing and various bug fixes.

2.3 Faster preprocessing

Boundary mapping, inverse mapping, and repeated boundary evaluations have been reorganized to reduce preprocessing time and memory use, particularly for larger models with many boundary facets. Coreform IGA Interop also reuses boundary information more efficiently as it translates the model and creates Abaqus boundary elements.

3 Analysis and Abaqus interoperability

3.1 Improved implicit direct-integration dynamics

The initial **Alpha** implicit direct-integration dynamics workflow now handles initial acceleration, preload histories, restarts, and half-increment residual calculations more consistently. The implementation also improves Hilber—Hughes—Taylor time integration for both linear and nonlinear response.

3.2 Nearly incompressible analysis

The **Alpha** locking-resistant formulation for nearly incompressible material response has been substantially revised. The new formulation uses a condensed, degree-reduced pressure field and supports small- and finite-strain workflows, including mixed step kinematics. This remains an advanced evaluation capability; see Nearly incompressible elasticity — Cook's membrane for its activation workflow and limitations.

3.3 Improved ODB output

The output mesh in `<job-name>_iga.odb` now guarantees continuous fields throughout the interior without holes, and field results are exact at output-node locations. This provides cleaner, more reliable contour plots in Abaqus/CAE without changing the underlying analysis. Output file sizes have also been significantly reduced and are typically one-fifth to one-tenth their previous size for the same problem.

3.4 Abaqus translation and results

This release includes several interoperability improvements:

- Abaqus names and labels containing spaces are quoted correctly in translated input files.
- Unsupported IGA output variables are reported and skipped without preventing supported output from being generated.
- IGA output databases honor the requested field-output time interval.

- Generated IGA boundary sets remain isolated from native finite-element sets in mixed models.

3.5 Parallel execution

Coreform IGA jobs continue to use the supported shared-memory, threaded Abaqus/Standard execution mode by default. Distributed-memory Abaqus/Standard execution with MPI is now available as an **Experimental** capability for immersed IGA workflows, including hybrid MPI/threaded execution. MPI runs produce a single `<job-name>_iga.odb` with supported displacement (`U`), stress (`S`), and equivalent plastic strain (`PEEQ`) fields.

The Experimental MPI workflow currently supports only these nodal IGA output fields; Coreform probes and other custom output remain unsupported in MPI runs.

4 Installation and compatibility

4.1 Simplified Abaqus Python setup

Coreform IGA for Abaqus no longer requires a separate `h5py` installation in the Abaqus Python environment. The product now includes the HDF5 support needed to write Coreform model files, eliminating that installation and configuration step.

4.2 Supported application

Coreform IGA for Abaqus is supported through an Abaqus CAE custom application. Launch it with `coreform_iga` on Linux or `coreform_iga.bat` on Windows. Plain `abaqus cae` does not load the Coreform IGA interface or data classes. Importing the installed `coreform` modules into a plain Abaqus/CAE session is possible but unsupported.

4.3 Compatibility changes

- The redesigned Python interface and GUI do not provide backward compatibility with earlier Coreform IGA journals or with Coreform-specific data saved in model databases by earlier releases. Recreate earlier models and scripts with the new interface; the User Guide and Example Problems show the current workflows. The former `coreform` name now belongs to the Coreform IGA for Abaqus Python package.
- The legacy `incompressibility` setting has been removed and has no replacement field. Remove it from carried-over scripts or saved settings.
- Maintained workflows are available in the Example Problems section of this documentation.