

Running Jobs

Table of contents

1 Configure a local Job	1
2 Prepare and run the pipeline	2
3 Configure a queue	2
4 Start from a portable Job	3

A Coreform IGA Job coordinates three ordered stages: Coreform IGA Mesh generation, Coreform-to-Abaqus interoperability, and the Abaqus solve. Its persistent definition is separate from the native Abaqus Job created when execution requires one.

1 Configure a local Job

```
from abaqus import mdb
import coreform.abaqus

coreform.abaqus.init_iga_mdb()

jobs = mdb.customData.CoreformIGA

job = jobs.Job(
    name="Beam-IGA",
    model="Beam",
    numIgaCpus=8,
    numCpus=16,
    memory=90,
    memoryUnits="PERCENTAGE",
)

job.setValues(numIgaCpus=12, numCpus=24)
```

The Mesh and solve stages have independent CPU settings. Executable fields can override the Coreform Mesh, MPI, and interoperability executables; leaving them empty uses the installed defaults.

2 Prepare and run the pipeline

Use `buildMesh()` when you want to prepare the Coreform Mesh and interoperability artifacts before submission. `submit()` prepares missing inputs and starts the supported local or Abaqus-queue execution path.

```
build = job.buildMesh()
if build.ok:
    print("Mesh and interoperability stages completed")

job.submit(overwriteExistingFiles=True)
job.waitForCompletion()

status = job.getRuntimeStatus()
print(status["status"])

odb = job.openOdb()
```

Job artifacts use the Job name as their default prefix. Existing output protection is intentional; opt into replacement only after confirming that the prior analysis files are no longer needed.

3 Configure a queue

Use `RemoteQueue` for a named Abaqus environment-file queue or an external scheduler. The same queue can apply to all stages, or each stage can have its own queue.

```
from coreform.job import RemoteQueue

slurm = RemoteQueue(
    scheduler="SLURM",
    queue_name="compute",
    account="project-123",
    walltime="02:00:00",
    nodes=2,
)

job = jobs.Job(
    name="Beam-IGA",
    model="Beam",
    queue=slurm,
    numIgaCpus=16,
    numCpus=32,
)
```

```
job.writeRunScript(  
    fileName="Beam-IGA.slurm",  
    dialect="SLURM",  
    prepareInputs=True,  
)
```

`writeRunScript()` is useful when the pipeline will run outside the current Abaqus/CAE process or when a scheduler configuration cannot be represented by in-CAE submission.

4 Start from a portable Job

Build or deserialize a `CoreformIGAJobSpec` outside Abaqus, then pass it through the mutually exclusive `spec=` form:

```
job = jobs.Job(spec=portableJobSpec)
```

Call `record.toSpec()` to return to the immutable portable representation. Power users can also inspect or compose the portable artifact, execution-plan, execution-context, and script-rendering types without using the persistent Abaqus record.

See the [Abaqus Job reference](#) and [portable Job reference](#) for exact construction fields, runtime methods, queue types, status fields, and repository operations.