coreform.abaqus

Table of contents

Source directory: `interop/abaqus/cf_app/coreform/abaqus`

Directory tree

- abaqus/
  - cf_export/
  - job/
  - mesh/
  - probe/

1 constants.py

Module path: `coreform.abaqus.constants`

Available API members

Name	Kind	Description
POINT_PROBE	Attribute	Abaqus symbolic constant for a point Probe.

Name	Kind	Description
`LINE_PROBE`	Attribute	Abaqus symbolic constant for a line Probe.
`MULTI_POINT_PROBE`	Attribute	Abaqus symbolic constant for a multi-point Probe.

1.1 `POINT_PROBE`

Object path: `coreform.abaqus.constants.POINT_PROBE`

Type: `object`

Abaqus symbolic constant for a point Probe.

1.2 `LINE_PROBE`

Object path: `coreform.abaqus.constants.LINE_PROBE`

Type: `object`

Abaqus symbolic constant for a line Probe.

1.3 `MULTI_POINT_PROBE`

Object path: `coreform.abaqus.constants.MULTI_POINT_PROBE`

Type: `object`

Abaqus symbolic constant for a multi-point Probe.

2 lifecycle.py

Module path: `coreform.abaqus.lifecycle`

Available API members

Name	Kind	Description
`COREFORM_STATUS_ABSENT`	Attribute	No Coreform IGA custom data is present.
`COREFORM_STATUS_CURRENT`	Attribute	Coreform IGA custom data matches the current schema.
`COREFORM_STATUS_INCOMPATIBLE`	Attribute	Coreform IGA custom data is present but incompatible.

Name	Kind	Description
`model_coreform_status`	Function	Return the Coreform IGA customData status for an Abaqus model.
`mdb_coreform_status`	Function	Return the MDB-level Coreform Job storage status without changing it.
`init_iga_mdb`	Function	Create CoreformIGA customData needed by the current API.
`init_iga_mesh_owner`	Function	Attach the Coreform IGA mesh-owner namespace to one Part/PartInstance.

2.1 `COREFORM_STATUS_ABSENT`

Object path: `coreform.abaqus.lifecycle.COREFORM_STATUS_ABSENT`

Type: `str`

No Coreform IGA custom data is present.

2.2 `COREFORM_STATUS_CURRENT`

Object path: `coreform.abaqus.lifecycle.COREFORM_STATUS_CURRENT`

Type: `str`

Coreform IGA custom data matches the current schema.

2.3 `COREFORM_STATUS_INCOMPATIBLE`

Object path: `coreform.abaqus.lifecycle.COREFORM_STATUS_INCOMPATIBLE`

Type: `str`

Coreform IGA custom data is present but incompatible.

2.4 `model_coreform_status`

Object path: `coreform.abaqus.lifecycle.model_coreform_status`

Description: Return the Coreform IGA customData status for an Abaqus model.

Call syntax

```
result = coreform.abaqus.lifecycle.model_coreform_status(model=model)
```

Returns

Type	Description
`str`	Return the Coreform IGA customData status for an Abaqus model.

Signatures

```
model_coreform_status(model: object) -> str
```

Parameters

Name	Type	Required	Default	Description
`model`	`object`	Yes	—	Abaqus model whose Steps and Mesh namespaces are inspected.

2.5 `mdb_coreform_status`

Object path: `coreform.abaqus.lifecycle.mdb_coreform_status`

Description: Return the MDB-level Coreform Job storage status without changing it.

Call syntax

```
result = coreform.abaqus.lifecycle.mdb_coreform_status(mdb_object=mdb_object)
```

Returns

Type	Description
`str`	Return the MDB-level Coreform Job storage status without changing it.

Signatures

```
mdb_coreform_status(mdb_object: object) -> str
```

Parameters

Name	Type	Required	Default	Description
`mdb_object`	`object`	Yes	—	Abaqus model database whose Job namespace is inspected.

2.6 `init_iga_mdb`

Object path: `coreform.abaqus.lifecycle.init_iga_mdb`

Description: Create CoreformIGA customData needed by the current API.

Call syntax

```
result = coreform.abaqus.lifecycle.init_iga_mdb()
```

Returns

Type	Description
`object`	Initialized Abaqus model database.

Signatures

```
init_iga_mdb(backup: object = False, backupPath: Optional[str] = None,
saveAsPath: Optional[str] = None, requireJournalBackup: object = True) -> object
```

Parameters

Name	Type	Required	Default	Description
`backup`	`object`	No	`False`	Whether to create a model-data-base backup before initial-ization.
`backupPath`	`Optional[str]`	No	`None`	Explicit backup path, or `None` to derive it from the active database.

Name	Type	Required	Default	Description
`saveAsPath`	`Optional[str]`	No	`None`	Optional path at which to save and continue with a copy before initialization.
`requireJournalBackup`	`object`	No	`True`	Whether a missing journal backup should make initialization fail.

2.7 `init_iga_mesh_owner`

Object path: `coreform.abaqus.lifecycle.init_iga_mesh_owner`

Description: Attach the Coreform IGA mesh-owner namespace to one Part/PartInstance.

Call syntax

```
result = coreform.abaqus.lifecycle.init_iga_mesh_owner(owner=owner)
```

Returns

Type	Description
`object`	Attached or existing Mesh-owner Coreform IGA namespace.

Signatures

```
init_iga_mesh_owner(owner: object, ownerName: Optional[str] = None, ownerKind:
Optional[str] = None) -> object
```

Parameters

Name	Type	Required	Default	Description
`owner`	`object`	Yes	—	Live Abaqus Part/PartInstance, or a model name

Name	Type	Required	Default	Description
				during journal replay.
`ownerName`	`Optional[str]`	No	`None`	Owner name required by the journal-replay form.
`ownerKind`	`Optional[str]`	No	`None`	`part` or `instance` for the journal-replay form.

3 verification.py

Module path: `coreform.abaqus.verification`

Available API members

Name	Kind	Description
`VerificationResult.passed`	Attribute	Whether all inspected persistent data is compatible and valid.
`VerificationResult.issues`	Attribute	Structured issues found during verification.
`verify_iga_mdb`	Function	Validate existing CoreformIGA data without mutating the MDB.
`VerificationResult`	Class	Contain the pass/fail result and issues from MDB verification.

3.1 `VerificationResult`

Object path: `coreform.abaqus.verification.VerificationResult`

Contain the pass/fail result and issues from MDB verification.

Call syntax

```
result = VerificationResult(passed=passed, issues=issues)
```

Signatures

```
VerificationResult(passed: bool, issues: Tuple[ValidationIssue, ...])
```

Parameters

Name	Type	Required	Default	Description
`passed`	`bool`	Yes	—	Whether all inspected persistent data is compatible and valid.
`issues`	`Tuple[ValidationIssue, ...]`	Yes	—	Structured issues found during verification.

3.1.a Attributes

Name	Type	Value	Description
`passed`	`bool`	—	Whether all inspected persistent data is compatible and valid.
`issues`	`Tuple[ValidationIssue, ...]`	—	Structured issues found during verification.

3.2 `verify_iga_mdb`

Object path: `coreform.abaqus.verification.verify_iga_mdb`

Description: Validate existing CoreformIGA data without mutating the MDB.

Call syntax

```
result = coreform.abaqus.verification.verify_iga_mdb()
```

Returns

Type	Description
`VerificationResult`	Pass/fail status and all persistent-data issues found.

Signatures

```
verify_iga_mdb() -> VerificationResult
```

Parameters

None.