

coreform.abaqus.mesh

Table of contents

1 records.py	1
1.1 CoreformIGAMeshInfo	4
1.1.a Attributes	5
1.1.b to_dict	6
1.2 CoreformIGA	6
1.2.a Attributes	7
1.2.b owner	7
1.2.c ImmersedRectilinearMesh	7
1.2.d BodyFitMesh	10
1.2.e deleteMesh	12
1.3 mesh	13
1.3.a Attributes	15
1.3.b owner	17
1.3.c toSpec	17
1.3.d meshInfo	18
1.3.e validateForOwner	18
1.3.f setValues	19
1.3.g resolveForOwner	23

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

Directory tree

- `mesh/`

1 records.py

Module path: `coreform.abaqus.mesh.records`

Available API members

Name	Kind	Description
<code>CoreformIGAMeshInfo.frame_orientation_mode</code>	Attribute	GUI mode used to derive the Mesh frame orientation.

Name	Kind	Description
<code>CoreformIGAMeshInfo.padding</code>	Attribute	GUI mode used to derive frame padding.
<code>CoreformIGAMeshInfo.continuity</code>	Attribute	GUI mode used to derive basis continuity.
<code>CoreformIGAMeshInfo.bc_penalty</code>	Attribute	GUI setting used to derive the boundary-condition penalty.
<code>CoreformIGA.mesh</code>	Attribute	The persistent Mesh record, or <code>None</code> when no Mesh is defined.
<code>mesh.meshType</code>	Attribute	Type of the persistent Mesh record.
<code>mesh.visible</code>	Attribute	Whether the Mesh is visible in Abaqus/CAE.
<code>mesh.degree</code>	Attribute	Polynomial degree of the Mesh basis.
<code>mesh.continuity</code>	Attribute	Continuity of the Mesh basis.
<code>mesh.bcPenaltyValue</code>	Attribute	Penalty value used to impose essential boundary conditions.
<code>mesh.tessellationSize</code>	Attribute	Target tessellation size, or <code>None</code> when no target is specified.
<code>mesh.elementSize</code>	Attribute	Target element size when size-based Mesh sizing is active.
<code>mesh.elementIntervalCount</code>	Attribute	Element interval counts when interval-count sizing is active.
<code>mesh.elementIntervalCountByAxis</code>	Attribute	Per-axis interval counts when per-axis sizing is active.

Name	Kind	Description
<code>mesh.elementLayout</code>	Attribute	Element-layout choice for each parametric axis.
<code>mesh.smallCellVolumeRatio</code>	Attribute	Volume-ratio threshold used to identify small immersed cells.
<code>mesh.padding</code>	Attribute	Additional cell padding along each frame axis.
<code>mesh.frameOrigin</code>	Attribute	Origin of an explicit Mesh frame, or <code>None</code> for owner resolution.
<code>mesh.frameBasis</code>	Attribute	Basis of an explicit Mesh frame, or <code>None</code> for owner resolution.
<code>mesh.frameExtents</code>	Attribute	Minimum and maximum extents in the Mesh frame.
<code>CoreformIGAMeshInfo.to_dict</code>	Instance method	Return the GUI sidecar values as a dictionary.
<code>CoreformIGA.owner</code>	Instance method	Return the Abaqus Part or PartInstance that owns this namespace.
<code>CoreformIGA.ImmersedRectilinearMesh</code>	Instance method	Create and persist an immersed rectilinear Mesh for this owner.
<code>CoreformIGA.BodyFitMesh</code>	Instance method	Create and persist a body-fit Mesh for this owner.
<code>CoreformIGA.deleteMesh</code>	Instance method	Delete the Mesh currently attached to this owner.
<code>mesh.owner</code>	Instance method	Return the Abaqus Part or PartInstance that owns this Mesh.
<code>mesh.toSpec</code>	Instance method	Return a portable specification representing this Mesh record.

Name	Kind	Description
<code>mesh.meshInfo</code>	Instance method	Return the Abaqus GUI sidecar associated with this Mesh.
<code>mesh.validateForOwner</code>	Instance method	Validate this Mesh for a specified Abaqus owner.
<code>mesh.setValues</code>	Instance method	Update this persistent Mesh through Abaqus-facing camelCase fields.
<code>mesh.resolveForOwner</code>	Instance method	Return the portable Mesh specification with its owner-dependent frame resolved.
<code>CoreformIGAMeshInfo</code>	Class	Store Abaqus/CAE GUI choices that derive portable Mesh values.
<code>CoreformIGA</code>	Class	Persistent Mesh namespace attached to an Abaqus Part or PartInstance.
<code>mesh</code>	Class	Persistent Abaqus-facing Mesh record.

1.1 CoreformIGAMeshInfo

Object path: `coreform.abaqus.mesh.records.CoreformIGAMeshInfo`

Store Abaqus/CAE GUI choices that derive portable Mesh values.

Call syntax

```
result = CoreformIGAMeshInfo()
```

Signatures

```
CoreformIGAMeshInfo(frame_orientation_mode: Optional[str] = None, padding_mode:
Optional[str] = None, continuity_mode: Optional[str] = None, bc_penalty_setting:
Optional[str] = None)
```

Parameters

Name	Type	Required	Default	Description
<code>frame_orientation_mode</code>	<code>Optional[str]</code>	No	None	GUI mode used to derive the Mesh frame orientation.
<code>padding_mode</code>	<code>Optional[str]</code>	No	None	GUI mode used to derive frame padding.
<code>continuity_mode</code>	<code>Optional[str]</code>	No	None	GUI mode used to derive basis continuity.
<code>bc_penalty_setting</code>	<code>Optional[str]</code>	No	None	GUI setting used to derive the boundary-condition penalty.

1.1.a Attributes

Name	Type	Value	Description
<code>frame_orientation_mode</code>	<code>str or None</code>	<code>frame_orientation_mode</code>	GUI mode used to derive the Mesh frame orientation.
<code>padding_mode</code>	<code>str or None</code>	<code>padding_mode</code>	GUI mode used to derive frame padding.
<code>continuity_mode</code>	<code>str or None</code>	<code>continuity_mode</code>	GUI mode used to derive basis continuity.
<code>bc_penalty_setting</code>	<code>str or None</code>	<code>bc_penalty_setting</code>	GUI setting used to derive the boundary-condition penalty.

1.1.b to_dict

Defined on: `coreform.abaqus.mesh.records.CoreformIGAMeshInfo.to_dict`

Description: Return the GUI sidecar values as a dictionary.

Call syntax

```
result = coreformIGAMeshInfo.to_dict()
```

Returns

Type	Description
<code>dict[str, object]</code>	Read-only GUI-choice fields keyed by their portable names.

Signatures

```
to_dict() -> dict[str, object]
```

Parameters

None.

1.2 CoreformIGA

Public path: `meshOwner.customData.CoreformIGA`

Available API members

Name	Kind	Description
<code>mesh</code>	Attribute	The persistent Mesh record, or <code>None</code> when no Mesh is defined.
<code>owner</code>	Instance method	Return the Abaqus Part or PartInstance that owns this namespace.
<code>ImmersedRectilinearMesh</code>	Instance method	Create and persist an immersed rectilinear Mesh for this owner.
<code>BodyFitMesh</code>	Instance method	Create and persist a body-fit Mesh for this owner.
<code>deleteMesh</code>	Instance method	Delete the Mesh currently attached to this owner.

1.2.a Attributes

Name	Type	Value	Description
<code>mesh</code>	<code>object</code>	—	The persistent Mesh record, or <code>None</code> when no Mesh is defined.

1.2.b `owner`

Defined on: `meshOwner.customData.CoreformIGA.owner`

Description: Return the Abaqus Part or PartInstance that owns this namespace.

Call syntax

```
result = meshes.owner()
```

Returns

Type	Description
<code>object</code>	Return the Abaqus Part or PartInstance that owns this namespace.

Signatures

```
owner() -> object
```

Parameters

None.

1.2.c `ImmersedRectilinearMesh`

Defined on: `meshOwner.customData.CoreformIGA.ImmersedRectilinearMesh`

Description: Create and persist an immersed rectilinear Mesh for this owner.

Call syntax

```
result = meshes.ImmersedRectilinearMesh(spec=spec)
result = meshes.ImmersedRectilinearMesh(elementSize=elementSize)
```

Returns

Type	Description
ImmersedRectilinearMesh record	Newly created persistent Mesh record.

Signatures

```
ImmersedRectilinearMesh(*, spec, visible=True, meshInfo=None, overwrite=False)
-> object
ImmersedRectilinearMesh(*, degree=1, continuity=0, bcPenaltyValue=None,
tessellationSize=None, elementSize=None, elementIntervalCount=None,
elementIntervalCountByAxis=None, elementLayout=None, smallCellVolumeRatio=0.2,
padding=None, frameOrigin=None, frameBasis=None, frameExtents=None,
visible=True, meshInfo=None, overwrite=False) -> object
```

Parameters

For ImmersedRectilinearMesh(*, spec, visible=True, meshInfo=None, overwrite=False)
-> object:

Name	Type	Required	Default	Description
spec	ImmersedRectilinearMeshSpec or Mapping[str, object]	Yes	—	Complete portable specification or serialized specification mapping.
visible	bool	No	True	Whether the Mesh is visible in Abaqus/CAE.
meshInfo	CoreformIGAMeshInfo or None	No	None	Abaqus GUI sidecar values.
overwrite	bool	No	False	Replace an existing Mesh atomically when true.

For ImmersedRectilinearMesh(*, degree=1, continuity=0, bcPenaltyValue=None, tessellationSize=None, elementSize=None, elementIntervalCount=None, elementIntervalCountByAxis=None, elementLayout=None, smallCellVolumeRatio=0.2, padding=None, frameOrigin=None, frameBasis=None, frameExtents=None, visible=True, meshInfo=None, overwrite=False) -> object:

Name	Type	Required	Default	Description
degree	int	No	1	Polynomial degree of the Mesh basis.
continuity	int	No	0	Continuity of the Mesh basis.
bcPenaltyValue	float or None	No	None	Penalty value used to impose essential boundary conditions.
tessellationSize	float or None	No	None	Target tessellation size.
elementSize	Float3 or None	Exactly one of	None	Target element size.
		elementSize, elementIntervalCount, or elementIntervalCountByAxis		
elementIntervalCount	Tuple[int, int] or None	Exactly one of	None	Element interval counts.
		elementSize, elementIntervalCount, or elementIntervalCountByAxis		
elementIntervalCountByAxis	Tuple[int, int] or None	Exactly one of	None	Per-axis element interval counts.
		elementSize, elementIntervalCount, or elementIntervalCountByAxis		
elementLayout	Tuple[str, str, str] or None	No	None	Element-layout choice for each axis.
smallCellVolumeRatio	float	No	0.2	Volume-ratio threshold used to identify small cells.
padding	Tuple[int, int] or None	No	None	Additional cells along each frame axis.
frameOrigin	Vector3D or None	No	None	Origin of an explicit Mesh frame.

Name	Type	Required	Default	Description
<code>frameBasis</code>	<code>Tuple[Vector3D, Vector3D, Vector3D]</code> or <code>None</code>	No	<code>None</code>	Basis of an explicit Mesh frame.
<code>frameExtents</code>	<code>Tuple[Float3, Float3]</code> or <code>None</code>	No	<code>None</code>	Minimum and maximum frame extents.
<code>visible</code>	<code>bool</code>	No	<code>True</code>	Whether the Mesh is visible in Abaqus/CAE.
<code>meshInfo</code>	<code>CoreformIGAMeshInfo</code> or <code>None</code>	No	<code>None</code>	Abaqus GUI sidecar values.
<code>overwrite</code>	<code>bool</code>	No	<code>False</code>	Replace an existing Mesh atomically when true.

Parameter rules

Rule	Requirement
Call forms	Use exactly one of these call forms: <code>spec</code> or <code>abaqus_definition</code> .
Required group	Supply exactly one of <code>elementSize</code> , <code>elementIntervalCount</code> , or <code>elementIntervalCountByAxis</code> .
Parameter group	Supply <code>frameOrigin</code> , <code>frameBasis</code> , and <code>frameExtents</code> together, or omit all of them.

Raises

Type	Condition
<code>ValidationError</code>	If the specification or owner-dependent Mesh settings are invalid.

1.2.d BodyFitMesh

Defined on: `meshOwner.customData.CoreformIGA.BodyFitMesh`

Description: Create and persist a body-fit Mesh for this owner.

Call syntax

```
result = meshes.BodyFitMesh(spec=spec)
result = meshes.BodyFitMesh()
```

Returns

Type	Description
BodyFitMesh record	Newly created persistent Mesh record.

Signatures

```
BodyFitMesh(*, spec, visible=True, meshInfo=None, overwrite=False) -> object
BodyFitMesh(*, degree=1, continuity=0, bcPenaltyValue=None,
tessellationSize=None, visible=True, meshInfo=None, overwrite=False) -> object
```

Parameters

For `BodyFitMesh(*, spec, visible=True, meshInfo=None, overwrite=False) -> object`:

Name	Type	Required	Default	Description
spec	BodyFitMeshSpec or Mapping[str, object]	Yes	—	Complete portable specification or serialized specification mapping.
visible	bool	No	True	Whether the Mesh is visible in Abaqus/CAE.
meshInfo	CoreformIGAMeshInfo or None	No	None	Abaqus GUI sidecar values.
overwrite	bool	No	False	Replace an existing Mesh atomically when true.

For `BodyFitMesh(*, degree=1, continuity=0, bcPenaltyValue=None, tessellationSize=None, visible=True, meshInfo=None, overwrite=False) -> object`:

Name	Type	Required	Default	Description
<code>degree</code>	<code>int</code>	No	1	Polynomial degree of the Mesh basis.
<code>continuity</code>	<code>int</code>	No	0	Continuity of the Mesh basis.
<code>bcPenaltyValue</code>	<code>float</code> or <code>None</code>	No	<code>None</code>	Penalty value used to impose essential boundary conditions.
<code>tessellationSize</code>	<code>float</code> or <code>None</code>	No	<code>None</code>	Target tessellation size.
<code>visible</code>	<code>bool</code>	No	<code>True</code>	Whether the Mesh is visible in Abaqus/CAE.
<code>meshInfo</code>	<code>CoreformIGAMeshInfo</code> or <code>None</code>	No	<code>None</code>	Abaqus GUI sidecar values.
<code>overwrite</code>	<code>bool</code>	No	<code>False</code>	Replace an existing Mesh atomically when true.

Parameter rules

Rule	Requirement
Call forms	Use exactly one of these call forms: <code>spec</code> or <code>abaqus_definition</code> .

Raises

Type	Condition
<code>ValidationError</code>	If the specification or owner-dependent Mesh settings are invalid.

1.2.e `deleteMesh`

Defined on: `meshOwner.customData.CoreformIGA.deleteMesh`

Description: Delete the Mesh currently attached to this owner.

Call syntax

```
meshes.deleteMesh()
```

Returns

Type	Description
None	—

Signatures

```
deleteMesh() -> None
```

Parameters

None.

1.3 mesh

Public path: `meshOwner.customData.CoreformIGA.mesh`

Available API members

Name	Kind	Description
<code>meshType</code>	Attribute	Type of the persistent Mesh record.
<code>visible</code>	Attribute	Whether the Mesh is visible in Abaqus/CAE.
<code>degree</code>	Attribute	Polynomial degree of the Mesh basis.
<code>continuity</code>	Attribute	Continuity of the Mesh basis.
<code>bcPenaltyValue</code>	Attribute	Penalty value used to impose essential boundary conditions.
<code>tessellationSize</code>	Attribute	Target tessellation size, or <code>None</code> when no target is specified.

Name	Kind	Description
<code>elementSize</code>	Attribute	Target element size when size-based Mesh sizing is active.
<code>elementIntervalCount</code>	Attribute	Element interval counts when interval-count sizing is active.
<code>elementIntervalCountByAxis</code>	Attribute	Per-axis interval counts when per-axis sizing is active.
<code>elementLayout</code>	Attribute	Element-layout choice for each parametric axis.
<code>smallCellVolumeRatio</code>	Attribute	Volume-ratio threshold used to identify small immersed cells.
<code>padding</code>	Attribute	Additional cell padding along each frame axis.
<code>frameOrigin</code>	Attribute	Origin of an explicit Mesh frame, or <code>None</code> for owner resolution.
<code>frameBasis</code>	Attribute	Basis of an explicit Mesh frame, or <code>None</code> for owner resolution.
<code>frameExtents</code>	Attribute	Minimum and maximum extents in the Mesh frame.
<code>owner</code>	Instance method	Return the Abaqus Part or PartInstance that owns this Mesh.
<code>toSpec</code>	Instance method	Return a portable specification representing this Mesh record.
<code>meshInfo</code>	Instance method	Return the Abaqus GUI sidecar associated with this Mesh.
<code>validateForOwner</code>	Instance method	Validate this Mesh for a specified Abaqus owner.

Name	Kind	Description
<code>setValues</code>	Instance method	Update this persistent Mesh through Abaqus-facing camelCase fields.
<code>resolveForOwner</code>	Instance method	Return the portable Mesh specification with its owner-dependent frame resolved.

1.3.a Attributes

Name	Type	Applies to	Value	Description
<code>meshType</code>	<code>MeshType</code>	All	—	Type of the persistent Mesh record.
<code>visible</code>	<code>bool</code>	All	—	Whether the Mesh is visible in Abaqus/CAE.
<code>degree</code>	<code>int</code>	All	—	Polynomial degree of the Mesh basis.
<code>continuity</code>	<code>int</code>	All	—	Continuity of the Mesh basis.
<code>bcPenaltyValue</code>	<code>Optional[float]</code>	All	—	Penalty value used to impose essential boundary conditions.
<code>tessellationSize</code>	<code>Optional[float]</code>	All	—	Target tessellation size, or <code>None</code> when no target is specified.
<code>elementSize</code>	<code>Optional[Float3]</code>	<code>ImmersedRectilinearMesh</code>	—	Target element size when size-

Name	Type	Applies to	Value	Description
				based Mesh sizing is active.
elementIntervalCount	Optional[Tuple[int, int]]	ImmersedRectilinearMesh		Element interval counts when interval-count sizing is active.
elementIntervalCountByAxis	Optional[Tuple[int, int]]	ImmersedRectilinearMesh		Per-axis interval counts when per-axis sizing is active.
elementLayout	Optional[Tuple[str, str]]	ImmersedRectilinearMesh		Element-layout choice for each parametric axis.
smallCellVolumeRatio	float	ImmersedRectilinearMesh		Volume-ratio threshold used to identify small immersed cells.
padding	Optional[Tuple[int, int]]	ImmersedRectilinearMesh		Additional cell padding along each frame axis.
frameOrigin	Optional[Vector3D]	ImmersedRectilinearMesh		Origin of an explicit Mesh frame, or <code>None</code> for owner resolution.
frameBasis	Optional[Tuple[Vector3D, Vector3D, Vector3D]]	ImmersedRectilinearMesh		Basis of an explicit Mesh frame, or <code>None</code> for owner resolution.
frameExtents	Optional[Tuple[Float3, Float3]]	ImmersedRectilinearMesh		Minimum and maximum ex-

Name	Type	Applies to	Value	Description
				tents in the Mesh frame.

1.3.b owner

Defined on: `meshOwner.customData.CoreformIGA.mesh.owner`

Description: Return the Abaqus Part or PartInstance that owns this Mesh.

Call syntax

```
result = mesh.owner()
```

Returns

Type	Description
<code>object</code>	Return the Abaqus Part or PartInstance that owns this Mesh.

Signatures

```
owner() -> object
```

Parameters

None.

1.3.c toSpec

Defined on: `meshOwner.customData.CoreformIGA.mesh.toSpec`

Description: Return a portable specification representing this Mesh record.

Call syntax

```
result = mesh.toSpec()
```

Returns

Type	Description
<code>MeshSpec</code>	Return a portable specification representing this Mesh record.

Signatures

```
toSpec() -> MeshSpec
```

Parameters

None.

1.3.d meshInfo

Defined on: `meshOwner.customData.CoreformIGA.mesh.meshInfo`

Description: Return the Abaqus GUI sidecar associated with this Mesh.

Call syntax

```
result = mesh.meshInfo()
```

Returns

Type	Description
<code>CoreformIGAMeshInfo</code>	Return the Abaqus GUI sidecar associated with this Mesh.

Signatures

```
meshInfo() -> CoreformIGAMeshInfo
```

Parameters

None.

1.3.e validateForOwner

Defined on: `meshOwner.customData.CoreformIGA.mesh.validateForOwner`

Description: Validate this Mesh for a specified Abaqus owner.

Call syntax

```
mesh.validateForOwner(owner=owner)
```

Returns

Type	Description
<code>None</code>	—

Signatures

```
validateForOwner(owner: object, allow_existing: bool = False) -> None
```

Parameters

Name	Type	Required	Default	Description
<code>owner</code>	<code>object</code>	Yes	—	Abaqus Part or PartInstance against which to validate the Mesh.
<code>allow_existing</code>	<code>bool</code>	No	<code>False</code>	Whether validation permits an existing Mesh on the owner.

1.3.f setValues

Defined on: `meshOwner.customData.CoreformIGA.mesh.setValues`

Description: Update this persistent Mesh through Abaqus-facing camelCase fields.

Call syntax

```
mesh.setValues(...)
```

Returns

Type	Description
<code>None</code>	The record is updated in place.

Signatures

```
ImmersedRectilinearMesh.setValues(*, degree=...,
continuity=..., bcPenaltyValue=..., tessellationSize=..., elementSize=...,
elementIntervalCount=..., elementIntervalCountByAxis=..., elementLayout=...,
smallCellVolumeRatio=..., padding=..., frameOrigin=..., frameBasis=...,
frameExtents=..., visible=..., meshInfo=...) -> None
BodyFitMesh.setValues(*, degree=..., continuity=..., bcPenaltyValue=...,
tessellationSize=..., visible=..., meshInfo=...) -> None
```

Parameters

```
For ImmersedRectilinearMesh.setValues(*, degree=...,
continuity=..., bcPenaltyValue=..., tessellationSize=..., elementSize=...,
elementIntervalCount=..., elementIntervalCountByAxis=..., elementLayout=...,
smallCellVolumeRatio=..., padding=..., frameOrigin=..., frameBasis=...,
frameExtents=..., visible=..., meshInfo=...) -> None:
```

Name	Applies to	Type	Required	Default	Description
degree	All	int	No	Current value	Replace the polynomial degree.
continuity	All	int	No	Current value	Replace the basis continuity.
bcPenaltyValue	All	float None	or No	Current value	Replace the boundary-condition penalty value.
tessellationSize	All	float None	or No	Current value	Replace the target tessellation size.
elementSize	ImmersedRectilinearMesh	float	No	Current value	Switch to size-based rectilinear sizing.
elementIntervalCount	ImmersedRectilinearMesh	tuple[int, int]	No	Current value	Switch to interval-count rectilinear sizing.
elementIntervalCountByAxis	ImmersedRectilinearMesh	tuple[int, int]	No	Current value	Switch to per-axis interval-count sizing.

Name	Applies to	Type	Required	Default	Description
elementLayout	ImmersedRectilinear Mesh	Tuple[Mesh, str, str] or None	No	Current value	Replace the element-layout choices.
smallCellVolumeRatio	ImmersedRectilinear Mesh	float	No	Current value	Replace the small-cell volume-ratio threshold.
padding	ImmersedRectilinear Mesh	Tuple[Mesh, int, int] or None	No	Current value	Replace the frame padding.
frameOrigin	ImmersedRectilinear Mesh	Vector3D or None	No	Current value	Replace the explicit frame origin.
frameBasis	ImmersedRectilinear Mesh	Tuple[Mesh, Vector3D, Vector3D] or None	No	Current value	Replace the explicit frame basis.
frameExtents	ImmersedRectilinear Mesh	Tuple[Mesh, Float3] or None	No	Current value	Replace the frame extents.
visible	All	bool	No	Current value	Replace the Abaqus/CAE visibility setting.
meshInfo	All	CoreformIGAMeshInfo or None	No	Current value	Replace the Abaqus GUI sidecar.

For `BodyFitMesh.setValues(*, degree=..., continuity=..., bcPenaltyValue=..., tessellationSize=..., visible=..., meshInfo=...) -> None:`

Name	Applies to	Type	Required	Default	Description
<code>degree</code>	All	<code>int</code>	No	Current value	Replace the polynomial degree.
<code>continuity</code>	All	<code>int</code>	No	Current value	Replace the basis continuity.
<code>bcPenaltyValue</code>	All	<code>float</code> or <code>None</code>	No	Current value	Replace the boundary-condition penalty value.
<code>tessellationSize</code>	All	<code>float</code> or <code>None</code>	No	Current value	Replace the target tessellation size.
<code>visible</code>	All	<code>bool</code>	No	Current value	Replace the Abaqus/CAE visibility setting.
<code>meshInfo</code>	All	<code>CoreformIGA</code> , <code>MeshInfo</code> or <code>None</code>	No	Current value	Replace the Abaqus GUI sidecar.

Parameter rules

Rule	Requirement
Optional exclusive group	If updating this group, supply exactly one of <code>elementSize</code> , <code>elementIntervalCount</code> , or <code>elementIntervalCountByAxis</code> .
Parameter group	Supply <code>frameOrigin</code> , <code>frameBasis</code> , and <code>frameExtents</code> together, or omit all of them.
Applicability	<code>elementSize</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>elementIntervalCount</code> applies only to <code>ImmersedRectilinearMesh</code> .

Rule	Requirement
Applicability	<code>elementIntervalCountByAxis</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>elementLayout</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>smallCellVolumeRatio</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>padding</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>frameOrigin</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>frameBasis</code> applies only to <code>ImmersedRectilinearMesh</code> .
Applicability	<code>frameExtents</code> applies only to <code>ImmersedRectilinearMesh</code> .

Raises

Type	Condition
<code>ValidationError</code>	If an update is unsupported or produces an invalid Mesh.

1.3.g `resolveForOwner`

Defined on: `meshOwner.customData.CoreformIGA.mesh.resolveForOwner`

Description: Return the portable Mesh specification with its owner-dependent frame resolved.

Call syntax

```
result = mesh.resolveForOwner()
```

Returns

Type	Description
<code>ImmersedRectilinearMeshSpec</code>	Return the portable Mesh specification with its owner-dependent frame resolved.

Signatures

```
resolveForOwner(owner: object = None) -> ImmersedRectilinearMeshSpec
```

Parameters

Name	Type	Required	Default	Description
<code>owner</code>	<code>object</code>	No	<code>None</code>	Abaqus owner to resolve, or <code>None</code> to use this record's owner.