

coreform

Table of contents

1	<code>types.py</code>	1
1.1	<code>Float3</code>	2
1.2	<code>Point3D</code>	2
1.3	<code>Vector3D</code>	2
2	<code>validation.py</code>	2
2.1	<code>ValidationIssue</code>	3
2.1.a	Attributes	4
2.2	<code>ValidationError</code>	4
2.2.a	Attributes	5

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

Directory tree

- `coreform/`
 - `abaqus/`
 - `cf_export/`
 - `job/`
 - `mesh/`
 - `probe/`
 - `job/`
 - `mesh/`
 - `probe/`
 - `utils/`

1 `types.py`

Module path: `coreform.types`

Available API members

Name	Kind	Description
<code>Float3</code>	Attribute	Three floating-point components.
<code>Point3D</code>	Attribute	A point represented by three floating-point coordinates.

Name	Kind	Description
<code>Vector3D</code>	Attribute	A vector represented by three floating-point components.

1.1 Float3

Object path: `coreform.types.Float3`

Type: `Tuple[float, float, float]`

Three floating-point components.

1.2 Point3D

Object path: `coreform.types.Point3D`

Type: `Tuple[float, float, float]`

A point represented by three floating-point coordinates.

1.3 Vector3D

Object path: `coreform.types.Vector3D`

Type: `Tuple[float, float, float]`

A vector represented by three floating-point components.

2 validation.py

Module path: `coreform.validation`

Available API members

Name	Kind	Description
<code>ValidationIssue.path</code>	Attribute	Field path locating the problem within a nested specification.
<code>ValidationIssue.code</code>	Attribute	Stable machine-readable issue identifier.
<code>ValidationIssue.message</code>	Attribute	Human-readable explanation of the problem.
<code>ValidationError.issues</code>	Attribute	Structured validation issues carried by this exception.

Name	Kind	Description
<code>ValidationIssue</code>	Class	Describe one validation problem with a stable path and code.
<code>ValidationError</code>	Class	Collect one or more structured Coreform validation issues.

2.1 ValidationIssue

Object path: `coreform.validation.ValidationIssue`

Describe one validation problem with a stable path and code.

Call syntax

```
result = ValidationIssue(path=path, code=code, message=message)
```

Signatures

```
ValidationIssue(path: Tuple[str, ...], code: str, message: str)
```

Parameters

Name	Type	Required	Default	Description
<code>path</code>	<code>Tuple[str, ...]</code>	Yes	—	Field <code>path</code> locating the problem within a nested specification.
<code>code</code>	<code>str</code>	Yes	—	Stable machine-readable issue identifier.
<code>message</code>	<code>str</code>	Yes	—	Human-readable explanation of the problem.

2.1.a Attributes

Name	Type	Value	Description
<code>path</code>	<code>Tuple[str, ...]</code>	—	Field path locating the problem within a nested specification.
<code>code</code>	<code>str</code>	—	Stable machine-readable issue identifier.
<code>message</code>	<code>str</code>	—	Human-readable explanation of the problem.

2.2 ValidationError

Object path: `coreform.validation.ValidationError`

Collect one or more structured Coreform validation issues.

Call syntax

```
result = ValidationError(issues=issues)
```

Signatures

```
ValidationError(issues: Union['ValidationError', str, Iterable[ValidationIssue]])
```

Parameters

Name	Type	Required	Default	Description
<code>issues</code>	<code>Union['ValidationError', str, Iterable[ValidationIssue]]</code>	Yes	—	Existing error, fallback message, or non-empty structured issue collection.

2.2.a Attributes

Name	Type	Value	Description
<code>issues</code>	<code>Tuple[ValidationIssue]</code>	<code>tuple(issues)</code>	Structured validation issues carried by this exception.