



JPA ORM XML MetaData Reference (v5.0)

Table of Contents

| | |
|--|----|
| Metadata for description tag | 8 |
| Metadata for xml-mapping-metadata-complete tag | 9 |
| Metadata for package tag | 10 |
| Metadata for schema tag | 11 |
| Metadata for catalog tag | 12 |
| Metadata for access tag | 13 |
| Metadata for sequence-generator tag | 14 |
| Metadata for table-generator tag | 15 |
| Metadata for named-query tag | 16 |
| Metadata for named-native-query tag | 17 |
| Metadata for sql-result-set-mapping tag | 18 |
| Metadata for named-entity-graph tag | 19 |
| Metadata for named-attribute-node tag | 20 |
| Metadata for subgraph/subclass-subgraph tag | 21 |
| Metadata for entity-result tag | 22 |
| Metadata for field-result tag | 23 |
| Metadata for column-result tag | 24 |
| Metadata for mapped-superclass tag | 25 |
| Metadata for entity tag | 26 |
| Metadata for description tag | 27 |
| Metadata for table tag | 28 |
| Metadata for secondary-table tag | 29 |
| Metadata for join-table tag | 30 |
| Metadata for collection-table tag | 31 |
| Metadata for unique-constraint tag | 32 |
| Metadata for column tag | 33 |
| Metadata for primary-key-join-column tag | 34 |
| Metadata for join-column tag | 35 |
| Metadata for inverse-join-column tag | 36 |
| Metadata for shared-relation tag | 37 |
| Metadata for id-class tag | 38 |
| Metadata for inheritance tag | 39 |
| Metadata for discriminator-value tag | 40 |
| Metadata for discriminator-column tag | 41 |
| Metadata for id tag | 42 |
| Metadata for generated-value tag | 43 |
| Metadata for datastore-id tag | 44 |
| Metadata for surrogate-version tag | 45 |

| | |
|---|----|
| Metadata for embedded-id tag | 46 |
| Metadata for version tag | 47 |
| Metadata for basic tag | 48 |
| Metadata for temporal tag | 49 |
| Metadata for enumerated tag | 50 |
| Metadata for one-to-one tag | 51 |
| Metadata for many-to-one tag | 52 |
| Metadata for element-collection tag | 53 |
| Metadata for one-to-many tag | 54 |
| Metadata for many-to-many tag | 55 |
| Metadata for embedded tag | 56 |
| Metadata for order-by tag | 57 |
| Metadata for order-column tag | 58 |
| Metadata for map-key tag | 59 |
| Metadata for map-key-class tag | 60 |
| Metadata for map-key-temporal tag | 61 |
| Metadata for map-key-enumerated tag | 62 |
| Metadata for transient tag | 63 |
| Metadata for index tag | 64 |
| Metadata for foreign-key tag | 65 |
| Metadata for convert tag | 66 |
| Metadata for exclude-default-listeners tag | 67 |
| Metadata for exclude-superclass-listeners tag | 68 |
| Metadata for entity-listener tag | 69 |
| Metadata for pre-persist tag | 70 |
| Metadata for post-persist tag | 71 |
| Metadata for pre-remove tag | 72 |
| Metadata for post-remove tag | 73 |
| Metadata for pre-update tag | 74 |
| Metadata for post-update tag | 75 |
| Metadata for post-load tag | 76 |
| Metadata for attribute-override tag | 77 |
| Metadata for association-override tag | 78 |

JPA XML MetaData allows you to define mapping information but in a separate file (`orm.xml`) separating persistence mapping from your model. What follows provides a reference guide to `orm.xml` MetaData elements. Here is an example header for `orm.xml` files with XSD specification

```
<?xml version="1.0" encoding="UTF-8" ?>
<entity-mappings xmlns="http://xmlns.jcp.org/xml/ns/persistence/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/persistence/orm
    http://xmlns.jcp.org/xml/ns/persistence/orm_2_1.xsd" version="2.1">
  ...
</entity-mappings>
```

If using any of the DataNucleus extensions, then the XSD is [defined here](#), in which case you would define your header as :-

```
<?xml version="1.0" encoding="UTF-8" ?>
<entity-mappings xmlns="http://www.datanucleus.org/xsd/jpa/orm"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.datanucleus.org/xsd/jpa/orm
    http://www.datanucleus.org/xsd/jpa/orm_2_1.xsd" version="2.1">
  ...
</entity-mappings>
```

- `entity-mappings`
 - `description`
 - `persistence-unit-metadata`
 - `xml-mapping-metadata-complete`
 - `package`
 - `schema`
 - `catalog`
 - `access`
 - `sequence-generator`
 - `table-generator`
 - `named-query`
 - `query`
 - `named-native-query`

- query
- sql-result-set-mapping
 - entity-result
 - field-result
 - column-result
- mapped-superclass
 - description
 - id-class
 - datastore-id
 - column
 - generated-value
 - surrogate-version
 - column
 - exclude-default-listeners
 - exclude-superclass-listeners
 - entity-listeners
 - entity-listener
 - pre-persist
 - post-persist
 - pre-remove
 - post-remove
 - pre-update
 - post-update
 - post-load
- pre-persist
- post-persist
- pre-remove
- post-remove

- pre-update
- post-update
- post-load
- attributes
 - Same elements as under <entity> → <attributes>
- entity
 - description
 - table
 - unique-constraint
 - column-name
 - index
 - secondary-table
 - primary-key-join-column
 - primary-key-foreign-key
 - unique-constraint
 - column-name
 - index
 - primary-key-join-column
 - primary-key-foreign-key
 - id-class
 - datastore-id
 - column
 - generated-value
 - surrogate-version
 - column
 - inheritance
 - discriminator-value
 - discriminator-column

- [sequence-generator](#)
- [table-generator](#)
 - [index](#)
- [named-query](#)
 - [query](#)
- [named-native-query](#)
 - [query](#)
- [sql-result-set-mapping](#)
 - [entity-result](#)
 - [field-result](#)
 - [column-result](#)
- [named-entity-graph](#)
 - [named-attribute-node](#)
 - [subgraph](#)
 - [named-attribute-node](#)
 - [subclass-subgraph](#)
 - [named-attribute-node](#)
- [exclude-default-listeners](#)
- [exclude-superclass-listeners](#)
- [entity-listeners](#)
 - [entity-listener](#)
 - [pre-persist](#)
 - [post-persist](#)
 - [pre-remove](#)
 - [post-remove](#)
 - [pre-update](#)
 - [post-update](#)
 - [post-load](#)

- pre-persist
- post-persist
- pre-remove
- post-remove
- pre-update
- post-update
- post-load
- attribute-override
 - column
- association-override
 - join-column
- attributes
 - id
 - column
 - generated-value
 - sequence-generator
 - table-generator
 - embedded-id
 - basic
 - column
 - lob
 - temporal
 - enumerated
 - convert
 - version
 - column
 - many-to-one
 - join-column

- `join-table join-column inverse-join-column unique-constraint * column-name`
- `cascade cascade-all cascade-persist cascade-merge cascade-remove cascade-refresh`
- `element-collection`
 - `collection-table join-column index foreign-key`
 - `order-by`
 - `order-column`
 - `map-key`
 - `map-key-class`
 - `map-key-temporal`
 - `map-key-enumerated`
 - `join-table join-column foreign-key inverse-join-column inverse-foreign-key unique-constraint * column-name`
 - `join-column`
- `one-to-many`
 - `order-by`
 - `order-column`
 - `map-key`
 - `map-key-class`
 - `map-key-temporal`
 - `map-key-enumerated`
 - `join-table join-column inverse-join-column unique-constraint * column-name`
 - `join-column`
 - `cascade cascade-all cascade-persist cascade-merge cascade-remove cascade-refresh`
 - `shared-relation`
- `one-to-one`
 - `join-column`

- [foreign-key](#)
- [join-table](#) [join-column](#) [inverse-join-column](#) [unique-constraint](#) *
[column-name](#)
- [cascade](#) [cascade-all](#) [cascade-persist](#) [cascade-merge](#) [cascade-remove](#) [cascade-refresh](#)
- [many-to-many](#)
 - [order-by](#)
 - [order-column](#)
 - [map-key](#)
 - [map-key-class](#)
 - [map-key-temporal](#)
 - [map-key-enumerated](#)
 - [join-table](#) [join-column](#) [inverse-join-column](#) [unique-constraint](#) *
[column-name](#)
 - [cascade](#) [cascade-all](#) [cascade-persist](#) [cascade-merge](#) [cascade-remove](#) [cascade-refresh](#)
 - [shared-relation](#)
- [embedded](#)
 - [attribute-override](#)
- [transient](#)
- [embeddable](#)
 - [embeddable-attributes](#)
 - [basic](#)
 - [transient](#)

Metadata for description tag

The **<description>** element (**<entity-mappings>**) contains the text describing all classes (and hence entities) defined in this file. It serves no useful purpose other than descriptive.

Metadata for `xml-mapping-metadata-complete` tag

The `<xml-mapping-metadata-complete>` element (under `<persistence-unit-metadata>`) when specified defines that the classes in this file are fully specified with just their metadata and that any annotations should be ignored.

Metadata for package tag

The `<package>` element (under `<entity-mappings>`) contains the text defining the package into which all classes in this file belong.

Metadata for schema tag

The `<schema>` element (under `<entity-mappings>`) contains the default schema for all classes in this file.

Metadata for catalog tag

The `<catalog>` element (under `<entity-mappings>`) contains the default catalog for all classes in this file.

Metadata for access tag

The `<access>` element (under `<entity-mappings>`) contains the setting for how to access the persistent fields/properties. This can be set to either "FIELD" or "PROPERTY".

Metadata for sequence-generator tag

The **<sequence-generator>** element (under **<entity-mappings>**, or **<entity>** or **<id>**) defines a generator of sequence values, for use elsewhere in this persistence-unit.

| Attribute | Description | Values |
|-----------------|--|--------|
| name | Name of the generator (required) | |
| sequence-name | Name of the sequence | |
| initial-value | Initial value for the sequence | 1 |
| allocation-size | Number of values that the sequence allocates when needed | 50 |

Metadata for table-generator tag

The **<table-generator>** element (under **<entity-mappings>**, or **<entity>** or **<id>**) defines a generator of sequence values using a datastore table, for use elsewhere in this persistence-unit.

| Attribute | Description | Values |
|-------------------|---|---------------------|
| name | Name of the generator (required) | |
| table | name of the table to use for sequences | SEQUENCE_TABLE |
| catalog | Catalog to store the sequence table | |
| schema | Schema to store the sequence table | |
| pk-column-name | Name of the primary-key column in the table | SEQUENCE_NAME |
| value-column-name | Name of the value column in the table | NEXT_VAL |
| pk-column-value | Name of the value to use in the primary key column (for this row) | {name of the class} |
| initial-value | Initial value to use in the table | 0 |
| allocation-size | Number of values to allocate when needed | 50 |

Metadata for named-query tag

The **<named-query>** element (under **<entity-mappings>** or under **<entity>**) defines a JPQL query that will be accessible at runtime via the name. The element itself will contain the text of the query. It has the following attributes

| Attribute | Description | Values |
|-----------|-------------------|--------|
| name | Name of the query | |

Metadata for named-native-query tag

The **<named-native-query>** element (under **<entity-mappings>** or under **<entity>**) defines an SQL query that will be accessible at runtime via the name. The element itself will contain the text of the query. It has the following attributes

| Attribute | Description | Values |
|-----------|-------------------|--------|
| name | Name of the query | |

Metadata for sql-result-set-mapping tag

The **<sql-result-set-mapping>** element (under **<entity-mappings>** or under **<entity>**) defines how the results of the SQL query are output to the user per row of the result set. It will contain sub-elements. It has the following attributes

| Attribute | Description | Values |
|-----------|---|--------|
| name | Name of the SQL result-set mapping (referenced by native queries) | |

Metadata for named-entity-graph tag

The `<named-entity-graph>` element (under `<entity>`) defines an entity graph with root as that entity, accessible at runtime via the name. It has the following attributes

| Attribute | Description | Values |
|-----------|--------------------------|--------|
| name | Name of the entity graph | |

Metadata for named-attribute-node tag

The **<named-attribute-node>** element (under **<named-entity-graph>**) defines a node in the entity graph. It has the following attributes

| Attribute | Description | Values |
|-----------|--|--------|
| name | Name of the node (field/property) | |
| subgraph | Name of a subgraph that maps this attribute fully (optional) | |

Metadata for subgraph/subclass-subgraph tag

The <subgraph>/subclass-subgraph element (under <named-entity-graph>) defines a subgraph in the entity graph. It has the following attributes

| Attribute | Description | Values |
|-----------|---|--------|
| name | Name of the subgraph (referenced in the named-attribute-node) | |
| class | Type of the subgraph attribute | |

Metadata for entity-result tag

The `<entity-result>` element (under `<sql-result-set-mapping>`) defines an entity that is output from an SQL query per row of the result set. It can contain sub-elements of type `<field-result>`. It has the following attributes

| Attribute | Description | Values |
|----------------------|---|--------|
| entity-class | Class of the entity | |
| discriminator-column | Column containing any discriminator (so subclasses of the entity type can be distinguished) | |

Metadata for field-result tag

The **<field-result>** element (under **<entity-result>**) defines a field of an entity and the column representing it in an SQL query. It has the following attributes

| Attribute | Description | Values |
|-----------|--------------------------|--------|
| name | Name of the entity field | |
| column | Name of the SQL column | |

Metadata for column-result tag

The `<column-result>` element (under `<sql-result-set-mapping>`) defines a column that is output directly from an SQL query per row of the result set. It has the following attributes

| Attribute | Description | Values |
|-----------|------------------------|--------|
| name | Name of the SQL column | |

Metadata for mapped-superclass tag

These are attributes within the `<mapped-superclass>` tag (under `<entity-mappings>`). This is used to define the persistence definition for a class that has no table but is mapped.

| Attribute | Description | Values |
|-------------------|--|--------------------|
| class | Name of the class (required) | |
| metadata-complete | Whether the definition of persistence of this class is complete with this MetaData definition. That is, should any annotations be ignored. | true, false |

Metadata for entity tag

These are attributes within the `<entity>` tag (under `<entity-mappings>`). This is used to define the persistence definition for this class.

| Attribute | Description | Values |
|-------------------|--|---------------------|
| class | Name of the class (required) | |
| name | Name of the entity. Used by JPQL queries | |
| metadata-complete | Whether the definition of persistence of this class is complete with this MetaData definition. That is, should any annotations be ignored. | true, false |
| cacheable | Whether instances of this class should be cached in the L2 cache. New in JPA2 | true , false |

Metadata for description tag

The **<description>** element (under **<entity>**) contains the text describing the class being persisted. It serves no useful purpose other than descriptive.

Metadata for table tag

These are attributes within the `<table>` tag (under `<entity>`). This is used to define the table where this class will be persisted.

| Attribute | Description | Values |
|-----------|-----------------------------------|--------|
| name | Name of the table | |
| catalog | Catalog where the table is stored | |
| schema | Schema where the table is stored | |

Metadata for secondary-table tag

These are attributes within the `<secondary-table>` tag (under `<entity>`). This is used to define the join of a secondary table back to the primary table where this class will be persisted.

| Attribute | Description | Values |
|-----------|-----------------------------------|--------|
| name | Name of the table | |
| catalog | Catalog where the table is stored | |
| schema | Schema where the table is stored | |

Metadata for join-table tag

These are attributes within the `<join-table>` tag (under `<one-to-one>`, `<one-to-many>`, `<many-to-many>`). This is used to define the join table where a collection/maps relationship will be persisted.

| Attribute | Description | Values |
|----------------|--|--------|
| name | Name of the join table | |
| catalog | Catalog where the join table is stored | |
| schema | Schema where the join table is stored | |
| orphan-removal | Whether to remove orphans when either removing the owner or nulling the relation | false |

Metadata for collection-table tag

These are attributes within the <collection-table> tag (under <element-collection>). This is used to define the join table where a collections relationship will be persisted.

| Attribute | Description | Values |
|-----------|--|--------|
| name | Name of the join table | |
| catalog | Catalog where the join table is stored | |
| schema | Schema where the join table is stored | |

Metadata for unique-constraint tag

This element is specified under the <table>, <secondary-table> or <join-table> tags. This is used to define a unique constraint on the table. No attributes are provided, just sub-element(s) "column-name"

Metadata for column tag

These are attributes within the <column> tag (under <basic>). This is used to define the column where the data will be stored.

| Attribute | Description | Values |
|-------------------|---|-------------|
| name | Name of the column | |
| unique | Whether the column is unique | true, false |
| nullable | Whether the column is nullable | true, false |
| insertable | Whether the column is insertable | true, false |
| updatable | Whether the column is updatable | true, false |
| column-definition | DDL to use for the column (everything except the column name). This must include the SQL type of the column | |
| table | Table for the column ? | |
| length | Length for the column (when string type) | 255 |
| precision | Precision for the column (when numeric type) | 0 |
| scale | Scale for the column (when numeric type) | 0 |
| jdbc-type | The JDBC Type to use for this column (DataNucleus extension) | |
| position | The position to use for this column (first=0) (DataNucleus extension) | |

Metadata for primary-key-join-column tag

These are attributes within the `<primary-join-key-column>` tag (under `<secondary-table>` or `<entity>`). This is used to define the join of PK columns between secondary and primary tables, or between table of subclass and table of base class.

| Attribute | Description | Values |
|------------------------|---------------------------------|--------|
| name | Name of the column | |
| referenced-column-name | Name of column in primary table | |

Metadata for join-column tag

These are attributes within the `<join-column>` tag (under `<join-table>`). This is used to define the join column.

| Attribute | Description | Values |
|------------------------|---|-------------|
| name | Name of the column | |
| referenced-column-name | Name of the column at the other side of the relation that this is a FK to | |
| unique | Whether the column is unique | true, false |
| nullable | Whether the column is nullable | true, false |
| insertable | Whether the column is insertable | true, false |
| updatable | Whether the column is updatable | true, false |
| column-definition | DDL to use for the column (everything except the column name). This must include the SQL type of the column | |
| table | Table for the column ? | |

Metadata for inverse-join-column tag

These are attributes within the `<inverse-join-column>` tag (under `<join-table>`). This is used to define the join column to the element.

| Attribute | Description | Values |
|------------------------|---|-------------|
| name | Name of the column | |
| referenced-column-name | Name of the column at the other side of the relation that this is a FK to | |
| unique | Whether the column is unique | true, false |
| nullable | Whether the column is nullable | true, false |
| insertable | Whether the column is insertable | true, false |
| updatable | Whether the column is updatable | true, false |
| column-definition | DDL to use for the column (everything except the column name). This must include the SQL type of the column | |
| table | Table for the column ? | |

Metadata for shared-relation tag

These are attributes within the <**shared-relation**> tag (under <one-to-many> or <many-to-many>). **This is a DataNucleus Extension.** This is used to define a relation as being shared, with a distinguisher column.

| Attribute | Description | Values |
|-------------|--|--------|
| column | Name of the distinguisher column | |
| value | Value to store in the distinguisher column for this field | |
| primary-key | Whether the distinguisher column is part of the primary key (when join table). | |

Metadata for id-class tag

These are attributes within the <id-class> tag (under <entity>). This defines a identity class to be used for this entity.

| Attribute | Description | Values |
|-----------|---------------------------------------|--------|
| class | Name of the identity class (required) | |

Metadata for inheritance tag

These are attributes within the <**inheritance**> tag (under <entity>). This defines the inheritance of the class.

| Attribute | Description | Values |
|-----------|---|---|
| strategy | Strategy for inheritance in terms of storing this class | SINGLE_TABLE, JOINED, TABLE_PER_CLASS |

Metadata for discriminator-value tag

These are attributes within the <**discriminator-value**> tag (under <entity>). This defines the value used in a discriminator. The value is contained in the element. Specification of the value will result in a "value-map" discriminator strategy being adopted. If no discriminator-value is present, but discriminator-column is then "class-name" discriminator strategy is used.

Metadata for discriminator-column tag

These are attributes within the <**discriminator-column**> tag (under <entity>). This defines the column used for a discriminator.

| Attribute | Description | Values |
|--------------------|---|-----------------------|
| name | Name of the discriminator column | DTYPE |
| discriminator-type | Type of data stored in the discriminator column | STRING, CHAR, INTEGER |
| length | Length of the discriminator column | |

Metadata for id tag

These are attributes within the <id> tag (under <attributes>). This is used to define the field used to be the identity of the class.

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for generated-value tag

These are attributes within the `<generated-value>` tag (under `<id>`). This is used to define how to generate the value for the identity field.

| Attribute | Description | Values |
|-----------|--|---|
| strategy | Generation strategy. Please refer to the Identity Generation Guide | auto , identity, sequence, table |
| generator | Name of the generator to use if wanting to override the default DataNucleus generator for the specified strategy. Please refer to the <code><sequence-generator></code> and <code><table-generator></code> | |

Metadata for datastore-id tag

These are attributes within the `<datastore-id>` tag (under `<entity>`). This is used to define the entity is using datastore identity (DataNucleus extension).

| Attribute | Description | Values |
|-----------------|--|--------|
| column | Name of the surrogate column to add for the datastore identity. | |
| generated-value | Details of the generated value strategy and generator. Please refer to the <generated-value> | |

Metadata for surrogate-version tag

These are attributes within the <**surrogate-version**> tag (under <entity>). This is used to define the entity has a surrogate version column (DataNucleus extension).

| Attribute | Description | Values |
|-----------|---|--------------------|
| column | Name of the surrogate column to add for the version. | |
| indexed | Whether the surrogate version column should be indexed. | true, false |

Metadata for embedded-id tag

These are attributes within the `<embedded-id>` tag (under `<attributes>`). This is used to define the field used to be the (embedded) identity of the class. **Note that this is not yet fully supported - specify the fields in the class**

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for version tag

These are attributes within the `<version>` tag (under `<attributes>`). This is used to define the field used to hold the version of the class.

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for basic tag

These are attributes within the `<basic>` tag (under `<attributes>`). This is used to define the persistence information for the field.

| Attribute | Description | Values |
|-----------|---|-------------|
| name | Name of the field (required) | |
| fetch | Fetch type for this field | LAZY, EAGER |
| optional | Whether this field may be null and may be used in schema generation | true, false |

Metadata for temporal tag

These are attributes within the `<temporal>` tag (under `<basic>`). This is used to define the details of persistence as a temporal type. The contents of the element can be one of DATE, TIME, TIMESTAMP.

Metadata for enumerated tag

These are attributes within the `<enumerated>` tag (under `<basic>`). This is used to define the details of persistence as an enum type. The contents of the element can be one of **ORDINAL** or **STRING** to represent whether the enum is persisted as an integer-based or the actual string.

Metadata for one-to-one tag

These are attributes within the `<one-to-one>` tag (under `<attributes>`). This is used to define that the field is part of a 1-1 relation.

| Attribute | Description | Values |
|---------------|--|--------------------|
| name | Name of the field (required) | |
| target-entity | Class name of the related entity | |
| fetch | Whether the field should be fetched immediately | EAGER, LAZY |
| optional | Whether the field can store nulls. | true, false |
| mapped-by | Name of the field that owns the relation (specified on the inverse side) | |

Metadata for many-to-one tag

These are attributes within the <**many-to-one**> tag (under <attributes>). This is used to define that the field is part of a N-1 relation.

| Attribute | Description | Values |
|---------------|---|--------------------|
| name | Name of the field (required) | |
| target-entity | Class name of the related entity | |
| fetch | Whether the field should be fetched immediately | EAGER, LAZY |
| optional | Whether the field can store nulls. | true, false |

Metadata for element-collection tag

These are attributes within the <element-collection> tag (under <attributes>). This is used to define that the field is part of a 1-N non-PC relation.

| Attribute | Description | Values |
|--------------|---|-------------|
| name | Name of the field (required) | |
| target-class | Class name of the related object | |
| fetch | Whether the field should be fetched immediately | EAGER, LAZY |

Metadata for one-to-many tag

These are attributes within the <**one-to-many**> tag (under <attributes>). This is used to define that the field is part of a 1-N relation.

| Attribute | Description | Values |
|----------------|--|-------------|
| name | Name of the field (required) | |
| target-entity | Class name of the related entity | |
| fetch | Whether the field should be fetched immediately | EAGER, LAZY |
| mapped-by | Name of the field that owns the relation (specified on the inverse side) | |
| orphan-removal | Whether to remove orphans when either removing the owner or removing the element | false |

Metadata for many-to-many tag

These are attributes within the <**many-to-many**> tag (under <attributes>). This is used to define that the field is part of a M-N relation.

| Attribute | Description | Values |
|---------------|---|-------------|
| name | Name of the field (required) | |
| target-entity | Class name of the related entity | |
| fetch | Whether the field should be fetched immediately | EAGER, LAZY |
| mapped-by | Name of the field on the non-owning side that completes the relation. Specified on the owner side | |

Metadata for embedded tag

These are attributes within the `<embedded>` tag (under `<attributes>`). This is used to define that the field is part of an embedded relation.

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for order-by tag

This element is specified under <one-to-many> or <many-to-many>. It is used to define the field(s) of the element class that is used for ordering the elements when they are retrieved from the datastore. It has no attributes and the ordering is specified within the element itself. It should be a comma-separated list of field names (of the element) with optional "ASC" or "DESC" to signify ascending or descending

Metadata for order-column tag

This element is specified under <one-to-many> or <many-to-many>. It is used to define that the List will be ordered with the ordering stored in a surrogate column in the other table.

| Attribute | Description | Values |
|-------------------|---|-------------------|
| name | Name of the column | {fieldName}_ORDER |
| nullable | Whether the column is nullable | true, false |
| insertable | Whether the column is insertable | true, false |
| updatable | Whether the column is updatable | true, false |
| column-definition | DDL to use for the column (everything except the column name). This must include the SQL type of the column | |
| base | Origin of the ordering (value for the first element) | 0 |

Metadata for map-key tag

These are attributes within the `<map-key>` tag (under `<one-to-many>` or `<many-to-many>`). This is used to define the field of the value class that is the key of a Map.

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for map-key-class tag

These are attributes within the `<map-key-class>` tag (under `<one-to-many>` or `<many-to-many>`). This is used to define the key type for a Map.

| Attribute | Description | Values |
|-----------|----------------------------------|--------|
| class | Type used for the key (required) | |

Metadata for map-key-temporal tag

Within the `<map-key-temporal>` tag (under `<element-collection>`, `<one-to-many>` or `<many-to-many>`) you put the `TemporalType` value.

Metadata for map-key-enumerated tag

Within the **<map-key-enumerated>** tag (under **<element-collection>**, **<one-to-many>** or **<many-to-many>**) you put the EnumType value.

Metadata for transient tag

These are attributes within the <**transient**> tag (under <attributes>). This is used to define that the field is not to be persisted.

| Attribute | Description | Values |
|-----------|------------------------------|--------|
| name | Name of the field (required) | |

Metadata for index tag

These are attributes within the `<index>` element. This is used to define the details of an index when overriding the provider default.

| Attribute | Description | Values |
|-------------|--|--------|
| name | Name of the index | |
| unique | Whether the index is unique | |
| column-list | List of columns (including any ASC, DESC specifications for each column) | |

Metadata for foreign-key tag

These are attributes within the <foreign-key> element. This is used to define the details of a foreign-key when overriding the provider default.

| Attribute | Description | Values |
|---------------------|-----------------------------|--------|
| name | Name of the foreign-key | |
| value | Constraint mode | |
| foreignKeyDefiniton | The DDL for the foreign key | |

Metadata for convert tag

These are attributes within the <**convert**> element, under <**basic**>. This is used to define the use of type conversion on this field.

| Attribute | Description | Values |
|--------------------|--|--------------------|
| converter | Class name of the converter | |
| attribute-name | Name of the embedded field to convert (optional). Not yet supported | |
| disable-conversion | Whether to disable any auto-apply converters for this field | true, false |

Metadata for exclude-default-listeners tag

This element is specified under `<mapped-superclass>` or `<entity>` and is used to denote that any default listeners defined in this file will be ignored.

Metadata for `exclude-superclass-listeners` tag

This element is specified under `<mapped-superclass>` or `<entity>` and is used to denote that any listeners of superclasses will be ignored.

Metadata for entity-listener tag

These are attributes within the `<entity-listener>` tag (under `<entity-listeners>`). This is used to an EntityListener class and the methods it uses

| Attribute | Description | Values |
|-----------|--|--------|
| class | Name of the EntityListener class that receives the callbacks for this Entity | |

Metadata for pre-persist tag

These are attributes within the `<pre-persist>` tag (under `<entity>`). This is used to define any "PrePersist" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for post-persist tag

These are attributes within the `<post-persist>` tag (under `<entity>`). This is used to define any "PostPersist" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for pre-remove tag

These are attributes within the `<pre-remove>` tag (under `<entity>`). This is used to define any "PreRemove" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for post-remove tag

These are attributes within the <post-remove> tag (under <entity>). This is used to define any "PostRemove" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for pre-update tag

These are attributes within the `<pre-remove>` tag (under `<entity>`). This is used to define any "PreUpdate" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for post-update tag

These are attributes within the <post-update> tag (under <entity>). This is used to define any "PostUpdate" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for post-load tag

These are attributes within the `<post-load>` tag (under `<entity>`). This is used to define any "PostLoad" method callback.

| Attribute | Description | Values |
|-------------|-------------------------------|--------|
| method-name | Name of the method (required) | |

Metadata for attribute-override tag

These are attributes within the `<attribute-override>` tag (under `<entity>`). This is used to override the columns for any fields in superclasses

| Attribute | Description | Values |
|-----------|---------------------------------------|--------|
| name | Name of the field/property (required) | |

Metadata for association-override tag

These are attributes within the `<association-override>` tag (under `<entity>`). This is used to override the columns for any N-1/1-1 fields in superclasses

| Attribute | Description | Values |
|-----------|---------------------------------------|--------|
| name | Name of the field/property (required) | |