Class DefaultObjective

Object
    AbstractMetadata
        ModifiableMetadata
            ISOMetadata
                DefaultObjective

All Implemented Interfaces:

Serializable, Emptiable, LenientComparable, IdentifiedObject, Objective

---

```
public class DefaultObjective
extends ISOMetadata
implements Objective
```

Describes the characteristics, spatial and temporal extent of the intended object to be observed. The following properties are mandatory in a well-formed metadata according ISO 19115:

```
MI_Objective
  ┬─identifier…… Code used to identify the objective.
  │   └─code…… Alphanumeric value identifying an instance in the
namespace.
  └─objectiveOccurrence… Event or events associated with objective completion.
      ├─identifier…… Event name or number.
      │   └─code…… Alphanumeric value identifying an instance in the
namespace.
      ├─trigger…… Initiator of the event.
      ├─context…… Meaning of the event.
      ├─sequence…… Relative time ordering of the event.
      └─time…… Time the event occurred.
```

Limitations

- Instances of this class are not synchronized for multi-threading. Synchronization, if needed, is caller's responsibility.
- Serialized objects of this class are not guaranteed to be compatible with future Apache SIS releases. Serialization support is appropriate for short term storage or RMI between applications running the same version of Apache SIS. For long term storage, use XML instead.

Since:

0.3

See Also:

Serialized Form

Nested Class Summary

Nested classes/interfaces inherited from class ModifiableMetadata

ModifiableMetadata.State

Field Summary

Fields inherited from class ISOMetadata

identifiers

Constructor Summary

Constructors

Constructor	Description
DefaultObjective()	Constructs an initially empty objective.
DefaultObjective(Objective object)	Constructs a new instance initialized with the values from the specified metadata object.

Method Summary

All Methods | **Static Methods** | **Instance Methods** | **Concrete Methods**

Modifier and Type	Method	Description
static DefaultObjective	**castOrCopy**(Objective object)	Returns a SIS metadata implementation with the values of the given arbitrary implementation.
Collection<Extent>	**getExtents**()	Returns the extent information including the bounding box, bounding polygon, vertical and temporal extent of the objective.
Collection<InternationalString>	**getFunctions**()	Returns the role or purpose performed by or activity performed at the objective.
Collection<Identifier>	**getIdentifiers**()	Returns the code used to identify the objective.

Collection<Event>	**getObjectiveOccurences**()	Returns the event or events associated with objective completion.
Collection<Platform-Pass>	**getPass**()	Returns the pass of the platform over the objective.
InternationalString	**getPriority**()	Returns the priority applied to the target.
Collection<Instrument>	**getSensingInstruments**()	Returns the instrument which senses the objective data.
Collection<Objective-Type>	**getTypes**()	Returns the collection technique for the objective.
void	**setExtents**(Collection<? extends Extent> newValues)	Sets the extent information including the bounding box, bounding polygon, vertical and temporal extent of the objective.
void	**setFunctions**(Collection<? extends International-String> newValues)	Sets the role or purpose performed by or activity performed at the objective.
void	**setIdentifiers**(Collection<? extends Identifier> new-Values)	Sets the code used to identify the objective.
void	**setObjectiveOccurences**(Collection<? extends Event> newValues)	Sets the event or events associated with objective completion.
void	**setPass**(Collection<? extends PlatformPass> new-Values)	Sets the pass of the platform over the objective.
void	**setPriority**(International-String newValue)	Sets the priority applied to the target.
void	**setSensingInstruments**(Collection<? extends Instrument> newValues)	Sets the instrument which senses the objective data.
void	**setTypes**(Collection<? extends ObjectiveType> new-Values)	Sets the collection technique for the objective.

Methods inherited from class ISOMetadata

getIdentifier, getIdentifierMap, getStandard, setIdentifier

Methods inherited from class ModifiableMetadata

checkWritePermission, copyCollection, copyList, copyMap, copySet, deepCopy, nonNullCollection, nonNullList, nonNullMap, nonNullSet, singleton, state, writeCollection, writeList, writeMap, writeSet

Methods inherited from class AbstractMetadata

asMap, asTreeTable, clone, equals, equals, getInterface, hashCode, isEmpty, nilReasons, prune, toString

Methods inherited from class Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Constructor Details

DefaultObjective

```
public DefaultObjective()
```

Constructs an initially empty objective.

DefaultObjective

```
public DefaultObjective(Objective object)
```

Constructs a new instance initialized with the values from the specified metadata object. This is a *shallow* copy constructor, because the other metadata contained in the given object are not recursively copied.

Parameters:

`object` - the metadata to copy values from, or `null` if none.

See Also:

castOrCopy(Objective)

Method Details

castOrCopy

`public static DefaultObjective castOrCopy(Objective object)`

Returns a SIS metadata implementation with the values of the given arbitrary implementation. This method performs the first applicable action in the following choices:

- If the given object is `null`, then this method returns `null`.
- Otherwise if the given object is already an instance of `DefaultObjective`, then it is returned unchanged.
- Otherwise a new `DefaultObjective` instance is created using the copy constructor and returned. Note that this is a *shallow* copy operation, because the other metadata contained in the given object are not recursively copied.

Parameters:

`object` - the object to get as a SIS implementation, or `null` if none.

Returns:

a SIS implementation containing the values of the given object (may be the given object itself), or `null` if the argument was null.

getIdentifiers

`public Collection<Identifier> getIdentifiers()`

Returns the code used to identify the objective.

Unified identifiers view

In this SIS implementation, the collection returned by this method includes the XML identifiers (ID, UUID, *etc.*), thus providing a unified view of every kind of identifiers associated to this objective.

XML marshalling note

The `<mac:identifier>` element marshalled to XML will exclude all the above cited identifiers, for compliance with ISO 19115 model. Those identifiers will appear in other XML elements or attributes.

Specified by:

`getIdentifiers` in interface `IdentifiedObject`

Specified by:

`getIdentifiers` in interface `Objective`

Overrides:

`getIdentifiers` in class `ISOMetadata`

Returns:

identification of the objective.

See Also:

DefaultCitation.getIdentifiers(),
getIdentifiers(),
AbstractIdentifiedObject.getIdentifiers()

setIdentifiers

```
public void setIdentifiers(Collection<? extends Identifier> newValues)
```

Sets the code used to identify the objective.

XML identifiers (ID, UUID, *etc.*), are not affected by this method, unless they are explicitly provided in the given collection.

Parameters:

`newValues` - the new identifiers values.

getPriority

```
public InternationalString getPriority()
```

Returns the priority applied to the target. `null` if unspecified.

Specified by:

`getPriority` in interface `Objective`

Returns:

priority applied, or `null`.

setPriority

```
public void setPriority(InternationalString newValue)
```

Sets the priority applied to the target.

Parameters:

`newValue` - the new priority value.

getTypes

```
public Collection<ObjectiveType> getTypes()
```

Returns the collection technique for the objective.

Specified by:

`getTypes` in interface `Objective`

Returns:

collection technique for the objective.

setTypes

`public void setTypes(Collection<? extends ObjectiveType> newValues)`

Sets the collection technique for the objective.

Parameters:

`newValues` - the new types values.

getFunctions

`public Collection<InternationalString> getFunctions()`

Returns the role or purpose performed by or activity performed at the objective.

Specified by:

`getFunctions` in interface `Objective`

Returns:

role or purpose performed by or activity performed at the objective.

setFunctions

`public void setFunctions(Collection<? extends InternationalString> newValues)`

Sets the role or purpose performed by or activity performed at the objective.

Parameters:

`newValues` - the new functions values.

getExtents

`public Collection<Extent> getExtents()`

Returns the extent information including the bounding box, bounding polygon, vertical and temporal extent of the objective.

Specified by:

`getExtents` in interface `Objective`

Returns:

extent information.

setExtents

`public void setExtents(Collection<? extends Extent> newValues)`

Sets the extent information including the bounding box, bounding polygon, vertical and temporal extent of the objective.

Parameters:

`newValues` - the new extents values.

getObjectiveOccurences

`public Collection<Event> getObjectiveOccurences()`

Returns the event or events associated with objective completion.

> **Upcoming API change**
> This method is misspelled (missing "r"). Its name may be fixed in GeoAPI 4.0.

Specified by:

`getObjectiveOccurences` in interface `Objective`

Returns:

events associated with objective completion.

setObjectiveOccurences

`public void setObjectiveOccurences(Collection<? extends Event> newValues)`

Sets the event or events associated with objective completion.

> **Upcoming API change**
> This method is misspelled (missing "r"). Its name may be fixed in GeoAPI 4.0.

Parameters:

`newValues` - the new objective occurrences values.

getPass

`public Collection<PlatformPass> getPass()`

Returns the pass of the platform over the objective.

Specified by:

getPass in interface Objective

Returns:

pass of the platform.

setPass

```
public void setPass(Collection<? extends PlatformPass> newValues)
```

Sets the pass of the platform over the objective.

Parameters:

`newValues` - the new pass values.

getSensingInstruments

```
public Collection<Instrument> getSensingInstruments()
```

Returns the instrument which senses the objective data.

Specified by:

getSensingInstruments in interface Objective

Returns:

instrument which senses the objective data.

setSensingInstruments

```
public void setSensingInstruments(Collection<? extends Instrument> newValues)
```

Sets the instrument which senses the objective data.

Parameters:

`newValues` - the new sensing instruments values.